

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

## ДИПЛОМНИЙ ПРОЕКТ

Інтернет-платформа із CRM-системою для станцій технічного обслуговування  
Назва теми  
автомобілів

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр ДППЗ. 180146.01.08.ПЗ

Виконав студент IV курсу група ПЗ-18-1

  
Підпис

В. В. Охота  
Ініціали, прізвище

Керівник д-р фіз.-мат. наук професор  
Науковий ступінь, звання

  
Підпис

Л. П. Бедратюк  
Ініціали, прізвище

Нормоконтролер канд. пед. наук, доцент

  
Підпис

Н. І. Праворська  
Ініціали, прізвище

До захисту допускаю:  
Завідувач кафедри інженерії  
програмного забезпечення

  
Підпис

Л. П. Бедратюк  
Ініціали, прізвище

2 червня 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій  
Кафедра Інженерії програмного забезпечення  
Рівень вищої освіти Перший (бакалаврський)  
Галузь знань 12 «Інформаційні технології»  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ 173  
Завідувач кафедри [підпис]  
Л. П. Бедратюк  
01 03 2022 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)**

Охоті Вадимові Володимировичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інтернет-платформа із CRM-системою для станцій технічного обслуговування автомобілів

Керівник проекту (роботи) Бедратюк Леонід Петрович, д-р фіз.-мат. наук, професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2022 р. № 18





2. Строк подання студентом проекту (роботи) на кафедру 01.06.2022 р.

3. Вихідні дані до проекту (роботи) Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Презентаційні матеріали (слайди, 15 шт.)

6. Консультанти розділів дипломного проєкту (роботи)


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Праворська Н.І., доцент кафедри ПЗ		
Антиплагіат	Гурман І.В., доцент кафедри ПЗ		

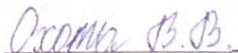
7. Дата видачі завдання «01» березня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1 Ознайомлення з тематикою дипломного проєктування (ДП), визначення та узгодження індивідуальних тем ДП	01.12– 30.12.2021	
2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2022	
3 Проєктування програмного забезпечення	01.02 – 28.02 2022	
4 Програмна реалізація	01.03 – 10.04.2022	
5 Тестування програмного забезпечення	11.04 – 30.04.2022	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2022	
7 Попередній захист ДП	травень 2022 (згідно графіка)	
8 Перевірка ДП на плагіат, нормоконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки	26.05 – 30.05.2022	
9 Підготовка до захисту та захист ДП	з 01.06.2022	

Студент

  
Підпис

  
Ініціали, прізвище

Керівник проєкту (роботи)

  
Підпис

  
Ініціали, прізвище

## АНОТАЦІЯ

Тема дипломного проекту: «Інтернет-платформа із CRM-системою для станцій технічного обслуговування автомобілів».

Автор проекту: Охота Вадим Володимирович.

Керівник проекту: Бедратюк Леонід Петрович.

Пояснювальна записка: 204 с., 48 рис., 6 табл., 4 дод., 30 джерел.

Графічна частина: 15 презентаційних слайдів.

СТАНЦІЯ СЕРВІСНОГО ОБСЛУГОВУВАННЯ АВТОМОБІЛІВ, ІНТЕРНЕТ-ПЛАТФОРМА, CRM, SPRING, APACHE, MySQL, MongoDB, DOCKER.

Об'єктом дослідження є CRM системи, що займається реалізацією систем у мережі Інтернет із створенням веб-сайту для підприємства.

Метою проекту є створення інтернет-платформи для сервісного центру обслуговування автомобілів із власною CRM системою, сторінкою для керування працівниками та сторінкою для фінансового обліку.

У дипломному проекті проведено аналіз предметної області, проведено аналіз схожого програмного забезпечення. Проведено аналіз архітектур для побудови WEB застосунків. Визначено основні модулі системи та здійсненна програмна реалізація.

Для розробки програмної системи використано мову програмування Java, фреймворк Spring, движок шаблонів Thymeleaf, сервер бази даних MySQL і MongoDB, систему авторизації та аутентифікації Keycloak, засіб контейнеризації Docker.

У результаті проектування здійснена програмна реалізація інтернет-платформи для станції сервісного обслуговування автомобілів із власною CRM системою.

31.05.22  
Дата

  
Підпис





## ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	ДППЗ. 180146.01.08.ПЗ	Пояснювальна записка	204		
2	A4		Завдання на дипломний проект	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4		Презентаційні матеріали	15		

ДППЗ. 180146.01.08.ВД										
Змн.	Арк.	№ докум.	Підпис	Дата	Інтернет-платформа із CRM-системою для станцій технічного обслуговування автомобілів	Лім.	Арк.	Аркуше		
		Охота В.В.		31.05	Відомість документів					
		Бедратюк Л.П.		31.05			4	204		
		Праворська Н.І.		31.05		ХНУ, ПЗ-18-1				
		Бедратюк Л.П.		31.05						

## ЗМІСТ

Перелік скорочень .....	4
Вступ .....	5
1 Дослідження предметної області та постановка задачі .....	8
1.1 Змістовий аналіз предметної області, її структурних та функціональних особливостей .....	8
1.2 Аналіз наявного програмно-технічного забезпечення предметної області .....	12
1.3 Визначення вимог до інтернет-платформи із системою CRM .....	23
2 Проектування інтернет-платформи .....	28
2.1 Архітектура та функціональна структура додатку .....	28
2.2 Аналіз продукту для авторизації та аутентифікації Keycloak .....	32
2.3 Визначення основних модулів застосунку .....	36
2.4 Проектування моделі бази даних .....	38
2.5 Проектування графічного інтерфейсу користувача .....	42
2.6 Аналіз та вибір технології та методів для реалізації застосунку .....	54
2.7 Розробка алгоритмів роботи застосунку .....	54
3 Програмна реалізація .....	61
3.1 Реалізація серверної частини програмного забезпечення .....	61
3.2 Реалізація сторінок інтернет-платформи .....	70
3.3 Керівництво користувача .....	71
3.4 Розгортання та встановлення системи .....	79
4 Тестування програмного забезпечення .....	83

ДІПІЗ. 180146.01.08.ВД									
Змн.	Арк.	№ докум.	Підпис	Дата	Інтернет-платформа із CRM-системою для станцій технічного обслуговування автомобілів  Відомість документів	Літ.	Арк.	Акрушіє	
		Виконав	Охота В.В.			21.05			
		Керівник	Бедратюк Л.П.			31.05		4	204
		Н. Контр.	Праворська Н.С.			31.05	ХНУ, ІПЗ-18-1		
		Затверд.	Бедратюк Л.П.		31.05				

4.1 Вибір та обґрунтування методів тестування застосунку .....	83
4.2 Розробка тестових наборів даних .....	84
4.3 Аналіз результатів тестування.....	87
Висновки.....	89
Перелік джерел посилання .....	91
Додаток А Діаграма варіантів використання .....	89
Додаток Б Технічне завдання.....	90
Додаток В Код (лістинг) програми .....	94
Додаток Г Презентаційні матеріали .....	196

					ЗППІПЗ.180146.01.08.00	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ	–	Програмне забезпечення
БД	–	база даних
ККД	–	Коефіцієнт корисної дії
CRM	–	Система управління відносинами з клієнтами
RPC	–	Виклик віддалених процедур

					ДППЗ. 180146.01.08.ВД	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Із збільшенням кількості автомобілів аналогічно збільшується і кількість сервісних центрів обслуговування, тож щоб бізнес процвітав потрібно привертати увагу нових клієнтів і що ж у теперішній час зробить це краще ніж веб-сторінка. Яскравий дизайн, перелік послуг та відгуки надають клієнтам одразу уяву за підприємство. Але простого сайту не достатньо, що якщо дати клієнтам можливість онлайн створити замовлення, дивитися статус виконання та отримувати інформацію за деталі ремонту автомобіля. Такий підхід для клієнтів дає можливість у будь-який час дізнатися всю потрібну інформацію без турбування працівників сервісу. У час коли смартфони становлять найбільшу ланку використання із всіх типів комп'ютерів, важливо щоб інтернет-застосунок інтегрувався під різні формати екранів, розроблене програмне забезпечення має інтегруватися під різні розміри дисплеїв.

Різні типи працівників зазвичай змушені використовувати різні програмні продукти для виконання своєї роботи, наприклад бухгалтер використовувати програми для обліку фінансів, менеджера табличні програми для контролю виконання та створення звітності. Тому з'являється чудова перспектива не просто сайту для клієнтів, а цілої системи яка зможе замінити стороні програмні продукти і дозволить використовувати лиш одне програмне забезпечення усім типам працівників.

CRM системи дають можливість відслідковувати та керувати процесом виконання замовлення, все більше підприємств використовує дані системи через зручність та зрозумілу наочність. Прозорість роботи, та наочність дає великі можливості для покращення взаємодії з клієнтами, також дозволяє отримати інформацію для подальшого розвитку підприємства, розподілу ресурсів сервісу, що призводить до збільшення прибутковості.

Актуальність теми полягає в тому, що на сьогодні вищезгадане програмне забезпечення є надзвичайно потрібним, більшість рішень які присутні на ринку не дають можливостей реалізації єдиного програмного продукту для

					ДППЗ. 180146.01.08.ВД	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

підприємства. Окрім того, більшість програмних продуктів потребують ліцензії для можливості їх використання, щоб забезпечити всі процеси на підприємстві необхідно хоча б декілька програмних продуктів, що у свою чергу приносить збільшення витрат.

Мета проєкту – розробка інтернет-платформи для станції сервісного обслуговування автомобілів, яка буде слугувати як сайт візитівка із особистим кабінетом для користувачів, із можливістю відстеження виконання замовлення та інформації про виконання. Також програмне забезпечення має включати у себе робочі сторінки для працівників СТО:

- CRM;
- керування працівниками;
- бухгалтерського обліку;
- відображення взаємодії із програмним забезпеченням.

Для побудови, реалізації та впровадження програмного продукту потрібно виконати наступні завдання:

- провести аналіз та визначити актуальність програмного продукту, знайти та проаналізувати корисність ПЗ та ключові особливості;
- дослідити та проаналізувати програмне забезпечення яке доступне на ринкові, знайти та проаналізувати найбільш корисні функції та наявні недоліки;
- визначити вимоги для розроблюваного програмного забезпечення та створити технічне завдання;
- визначити підходи та методи реалізації, спроектувати програмний продукт;
- виконати програмну реалізацію проєкту використовуючи мову програмування Java та фреймворк Spring, забезпечити кондефеційність та безпеку системи продуктом із відкритим кодом Keusloak;
- провести тестування ПЗ;
- створити Docker image ПЗ та БД які використовуються програмним забезпеченням, та контейнеризувати усі потрібні сторони елементи.

						ДППЗ. 180146.01.08.ВД	Арк.
							6
Змн.	Арк.	№ докум.	Підпис	Дата			

– Налаштувати та впровадити програмне забезпечення у контейнерах Docker у спільній локальній мережі.

					ДППЗ. 180146.01.08.ВД	Арк.
						7
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Змістовий аналіз предметної області, її структурних та функціональних особливостей

Інтернет-платформи замінили у нас час довідники та друковані газети, брошури. Інтернет сторінки в своїй більшості не являються статичними, та дозволяють виконувати різні дії та оперувати інформацією. За допомогою інтернет-платформ можливо створити нове замовлення, дізнаватися персоналізовану інформацією, керувати різними процесами. Тобто дані ресурси стали своєрідними панелями керування відповідним програмним забезпеченням у мережах як і локальних так і глобальних.

Популярність мобільних пристроїв зокрема смартфонів надав різкий розвиток технологій візуалізації програмного забезпечення, розробники створюють адаптивні інтерфейси які максимально ергономічні для користувачів на різній роздільній якості дисплею, що дозволяє використовувати ПЗ на більшості комп'ютерних пристроях.

Інтернет-платформи включають в один програмний продукт програмні рішення для безлічі проблем. Наприклад окремі можливості для користувачів, для створення відповідної інформації та оперування з нею, так і для працівників, функції для виконання власної роботи. Дані системи вже давно не однотипні та дають можливості розробникам реалізувати себе, відкривають цілий світ можливостей та ідей для розробки, що і дозволяє зробити програмне забезпечення індивідуальним, ергономічним що і дозволяє привабити нових користувачів.

Всесвітня практика показує що мережі для введення бізнесу є надзвичайно корисною річчю, адже дозволяє в одному програмному продукті об'єднати рішення на проблеми для яких раніше було потрібно використовувати окреме програмне забезпечення, що у свою чергу приносить зменшення витрат фінансової частини підприємства, а також полегшує роботу працівників за ПЗ та

					ДППЗ. 180146.01.08.ВД	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє у менші терміни виконувати більше завдань, що призводить до суттєвого збільшення ККД працівника.

Тенденцію популярності задають корпорації які активно впроваджують подібні програмні рішення у власні виробництва. Згідно з аналізами за 2022 рік майже 90% корпорацій та підприємств із штатом працівників більшим за 100 чоловік використовують інтернет-платформи для ведення бізнесу та керування всіма процесами підприємства.

Інтернет-платформи поєднали в собі можливості для повного ведення підприємства одним програмним продуктом. Тепер щоб керувати всіма ланками виробництва і відстеження продуктивних можливостей достатньо одного програмного продукту. Що у свою чергу дозволяє здійснити швидкий аналіз потребностей клієнта, популярні напрямки типів надання послуг. Також заданні системи дозволяють визначити які підрозділи підприємства найбільш затребуванні. Керуючись інформацією яку ми можемо отримати, відкриваються можливості для вдосконалення підприємства, грамотне розпорядження ресурсів, виконання затребуваних послуг дають напрямок для вдалого розвитку бізнесу та збільшення ККД усіх ланок підприємства, що у свою чергу веде знову до зменшення витрат та збільшення прибутку.

CRM системи набувають величезної ролі у бізнесах які спеціалізуються в сферах надання послуг. Згідно досліджень даної тематики із 2016 до 2022 року 90% успішних підприємств розпочало використовувати дані програмні продукти у веденні бізнесу. Що ж таке CRM система – це застосунок який дозволяє візуалізувати процес взаємодії з клієнтом та налагодити його. Відповідні працівники які займаються обслуговуванням клієнтів та процесом виконання замовлення мають змогу наглядно спостерігати за тенденціями замовлень, що дозволяє створювати план розподілу ресурсів підприємства для швидшого виконання замовлення, та оптимізування ресурсів. CRM система надає дані можливості керування замовленнями:

- зміна статусу виконання;
- розподіл працівників для виконання замовлення;

					ДППЗ. 180146.01.08.ВД	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- розподіл продуктивних ресурсів;
- аналіз та оптимізація часу виконання;
- підведення підсумків результатів по закінченню виконання роботи.

Все більше людей обживається власними автомобільними засобами, тенденція відображає що на середньостатичну сім'ю в Україні приходиться два автомобілі, тенденція також показує що іде активний розвиток і до трьох автомобілів. Також слід відзначити збільшення кількості нових сімей, що порозводить до збільшення сумарної кількості. Тобто кількість автомобілів збільшується у геометричній прогресії, відповідно й до цього зростає потрібність у ремонті автомобілів, що збільшує кількість сервісних станцій обслуговування автомобілів. Кількість підприємств які спеціалізуються в наданнях послуг в даній тематиці збільшується та наявні невпинно розвиваються та збільшують свої продуктивні можливості. Роль конкуренції теж відіграє ключову позицію в даному секторі надання послуг. Тому більшість підприємств намагаються методами реклами збільшити потік клієнтів у підприємство. У останні роки домінували у цьому односторінкові інтернет-сторінки які містили зазвичай інформацію про підприємство та форму зворотного зв'язку. Конкуренція відіграла велику роль у розвитку даної тематики та створила умови для розвитку рекламно-інформаційних сторінок на функціональні сторінки. Клієнтам надана можливість в режимі реального часу дізнатися статус виконання роботи, послуги які надає підприємство для виконання замовлення. Також до інформації до статусу виконання замовлення додається і лист із вартістю надання послуг та розтратних матеріалах які були закуплені та використані під час виконання замовлення.

Зазвичай сервіси які доступні на ринку які надіють послуги CRM систем є широконаправленими та мають в меті дозволити використовувати програмний продукт для різних підприємств, рішення не цілком правильне, розглянемо основні плюси:

- можливість адаптації системи для різних типів завдань;

					ДППЗ. 180146.01.08.ВД	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

– автономність системи, зазвичай ПЗ виконується на серверах підприємства надання програмного продукту.

Із більшості наявних на ринку CRM систем можна виділити такі важливі недоліки:

– працівник має пройти весь курс навчання, для використання можливостей системи;

– система потребує для використання щомісячної абонплати;

– більшість індивідуальних можливостей додатково потребують щомісячної платні;

Для зберігання інформації в системах доступних на ринку зазвичай використовуються віддалені бази даних що у свою чергу приносить складність в оперування інформацією та міграцією даних, адже доступ для оперування зазвичай відбувається тільки через програмне забезпечення CRM системи. Всі вхідні дані записуються в таблиці баз даних, та при необхідності динамічно змінюються. Більшість систем мають обмеження кількості записів в базі даних та змушують видаляти застарілі записи. Це зменшує можливості аналізу підприємства, також зменшує можливості побудові звітностей.

Програмного забезпечення яке доступне на ринку майже немає можливості щоб покрити усі проблеми для підприємства сервісного обслуговування автомобілів, також наявні програмні застосунки не надають можливості використання одного програмного продукту для заміни декількох, що у свою чергу підтверджує необхідність даного розроблюваного програмного забезпечення .

Підсумовуючи потрібність даної інтернет платформи потрібно використати усі стандарти та позитивні моменти наведені вище у програмному забезпеченні, також виправити недоліки які мають наявні на ринку аналогічні програмні продукти.

Тим не менш, слід розуміти розвиток тенденцій та масштабування підприємства та готувати програмний продукт для збільшення навантаження та впровадження на різних серверних системах. Використання сучасних технологій

						ДППЗ. 180146.01.08.ВД	Арк.
							11
Змн.	Арк.	№ докум.	Підпис	Дата			

та надання можливостей для масштабування програного продукту, відкрита система для персоналу підтримки програмного забезпечення надає суттєву перевагу розроблювального програмного забезпечення серед аналогічних.

## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

На теперішній час існує безліч програм для створення веб-сторінок для ведення підприємства, але не всі з них підходять для станції сервісного обслуговування автомобілів. Це пов'язано із недостатньою кількістю можливостей заповнення інформації замовлення а оперування нею, також більшість програмних продуктів розроблені саме для використання для працівників та не надають жодної інформації для клієнтів за стан виконання замовлення. Також більшість сервісів надають можливостей створення рекламної інтернет-сторінки за допомогою конструктора сайтів, та слід відмітити суттєвий недолік, більшість даних сайтів виглядають однотипно.

Розглянемо популярні на ринку програмні продукти які можливо використати для цілі яка вкладена в розробку програмного забезпечення, найпопулярнішою системою в світі виступає інтернет-платформа “Wix.com”, На рисунку 1.1 зображено головну сторінку платформи. На головній сторінці зображено банер із закликом створення власного сайту, також зображено меню із списком сторінок сайту. Сервіс надає можливості використати шаблони для створення власного сайту, шаблони зображені на рисунку 1.2. Конструктор сайтів даного сервісу має ряд достоїнств:

- простота в використанні;
- різні типи блоків для побудови сайту;
- можливість анімації графічних елементів;
- середовище розробки достатньо ергономічне.

						ДППЗ. 180146.01.08.ВД	Арк.
							12
Змн.	Арк.	№ докум.	Підпис	Дата			

У сервісі доступні і безкоштовні тарифні плани, но в них є суттєвий недостаток, він підходить лише для створення рекламної сторінки, та містить посилання на сервіс, також не дозволяє використовувати власний домен.

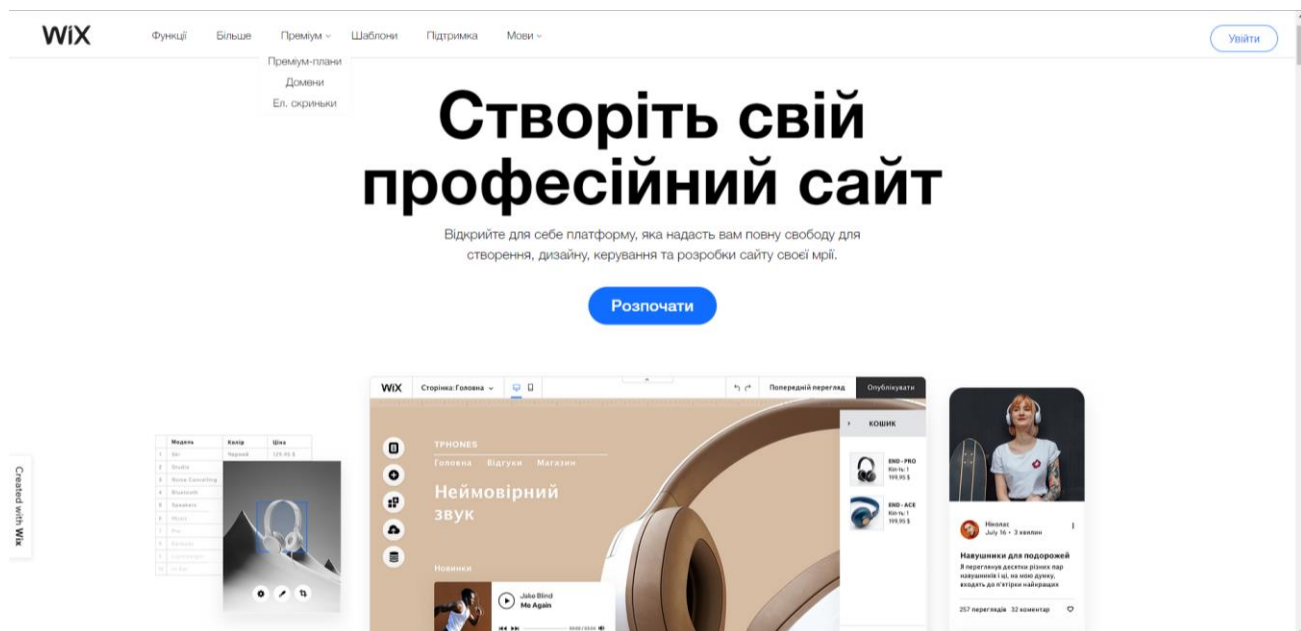


Рисунок 1.1 – Головна сторінка сервісу Wix.com

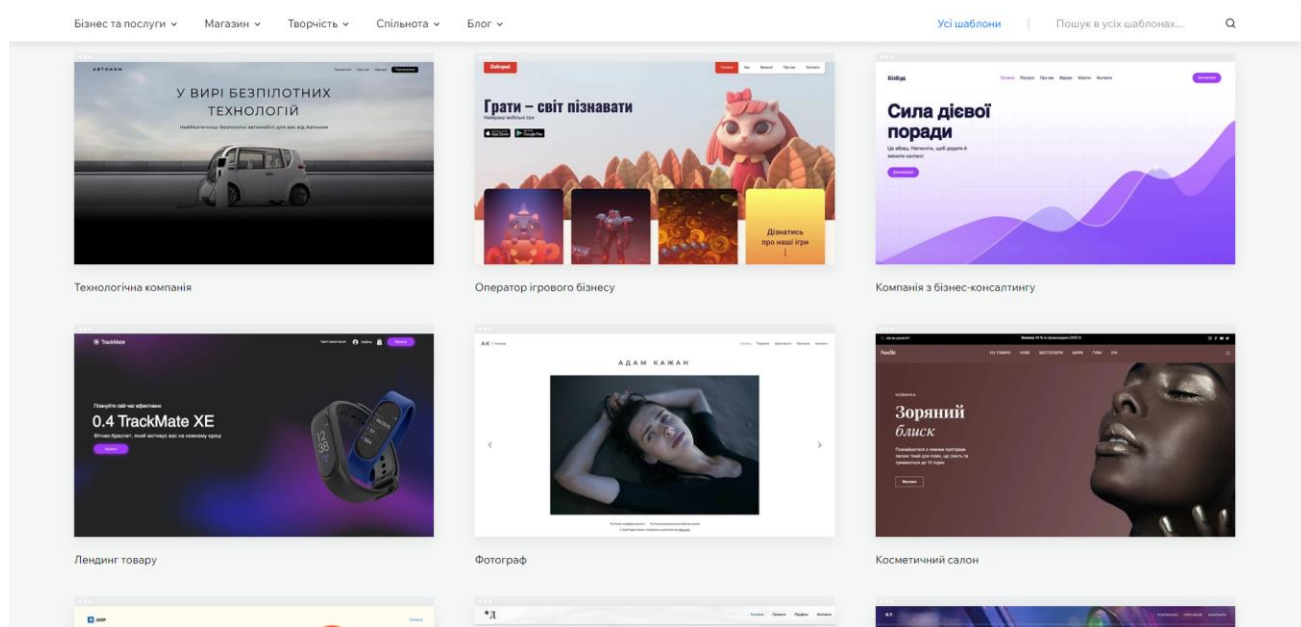


Рисунок 1.2 – Шаблони для створення сайту

В плані зовнішнього вигляду слід відзначити що сервіс володіє простим та ергономічним дизайном, що не містить нічого не потрібного для початку роботи

									Арк.
									13
Змн.	Арк.	№ докум.	Підпис	Дата					

із сервісом. Різні типи сторінок мають окремі типи оформлення та відчутно що дизайн сервісу не розроблявся копією одного шаблону. На сервісі доступні динамічні вікна для оперування різного типу інформацією, та для створення персоналізованих систем на основі даного сервісу.

Сервіс відповідно до тарифних планів малюнок 1.3 надає різні умови для ведення інтернет-платформи.

		популярний			
		VIP Пріоритетна підтримка	Unlimited Для підприємців та фрилансерів	Combo Персональний	Connect Domain Базовий
		—	—	—	ⓘ Цей план передбачає відображення реклами Wix
		<b>29 €</b> /міс.	<b>17 €</b> /міс.	<b>10 €</b> /міс.	<b>5.50 €</b> /міс.
Власний домен ⓘ	✓	✓	✓	✓	✓
Безкоштовний домен на 1 рік ⓘ	✓	✓	✓	✓	—
Сайт без реклами Wix ⓘ	✓	✓	✓	✓	—
Безкоштовний сертифікат SSL ⓘ	✓	✓	✓	✓	✓
Пропускна здатність ⓘ	Необмежена	Необмежена	Необмежена	2 ГБ	1 ГБ
Сховище ⓘ	35 ГБ	10 ГБ	10 ГБ	3 ГБ	500 МБ
Години відео ⓘ	5 годин	1 година	1 година	30 хвилин	—
Ваучер на рекламу на \$75 ⓘ	✓	✓	✓	✓	—
Додаток Site Booster Безкоштовно на 1 рік ⓘ	✓	✓	✓	—	—
Додаток Visitor Analytics Безкоштовно на 1 рік ⓘ	✓	✓	✓	—	—
Професійний логотип ⓘ	✓	—	—	—	—
Файли логотипів для соціальних мереж ⓘ	✓	—	—	—	—
Служба підтримки ⓘ	Пріоритетна підтримка клієнтів	Підтримка клієнтів 24/7	Підтримка клієнтів 24/7	Підтримка клієнтів 24/7	Підтримка клієнтів 24/7

Рисунок 1.3 – Тарифні плани сервісу “Wix.com”

Щодо ведення системи CRM система надає можливості використання програмних продуктів розроблених сторонніми розробниками. У свою чергу це приносить складність у використанні даного сервісу, адже немає спеціалізованого застосунку для вирішення даної проблеми, тобто клієнти сервісу змушені для створення системи під власне підприємство адаптуватися під стороннє програмне забезпечення.

Наступний сервіс який найпопулярніший в Україні виступає Bitrix 24, він більш спеціалізований саме під розробку CRM систем, також має простий конструктор сайтів, головна сторінка сервісу зображена на рисунку 1.4.



Рисунок 1.4 – Головна сторінка сервісу Бітрікс24

Сервіс спеціалізується на управлінні продажами в малих та середніх підприємствах, дозволяє використовувати канали зв'язку з клієнтами і автоматизує продажі. Програмне забезпечення доступне у віртуальному варіанті та у варіанті програмного забезпечення яке встановлюється на робочі комп'ютери. Серед плюсів сервісу можна виділити мобільний застосунок для операційних систем IOS та Android. Серед основних можливостей сервісу слід визначити наступні:

- керування лідами та замовленнями;
- налаштування власних стадій опрацювання продаж;
- виставлення рахунків за послугу;
- управління проектами та задачами між персоналом;
- автоматизація системи продажів та обробки замовлень;
- налаштування роботів для розсилання e-mail листів та смс повідомлень;
- можливість підключення відкритих ліній зв'язку із Facebook та Instagram,

та інших популярних месенджерів;

						ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			15

– налаштування планів продажу та звітів;  
 – формування документів із можливості вивантаження та друку із системи CRM;

– створення сайтів, лендінгів із влаштованою системою CRM для відображення необхідної інформації для замовників послуг.

Сервіс так як і попередній має доступний безкоштовний план. Тарифні плани сервісу Бітрікс24 наведені в рисунку 1.5 .

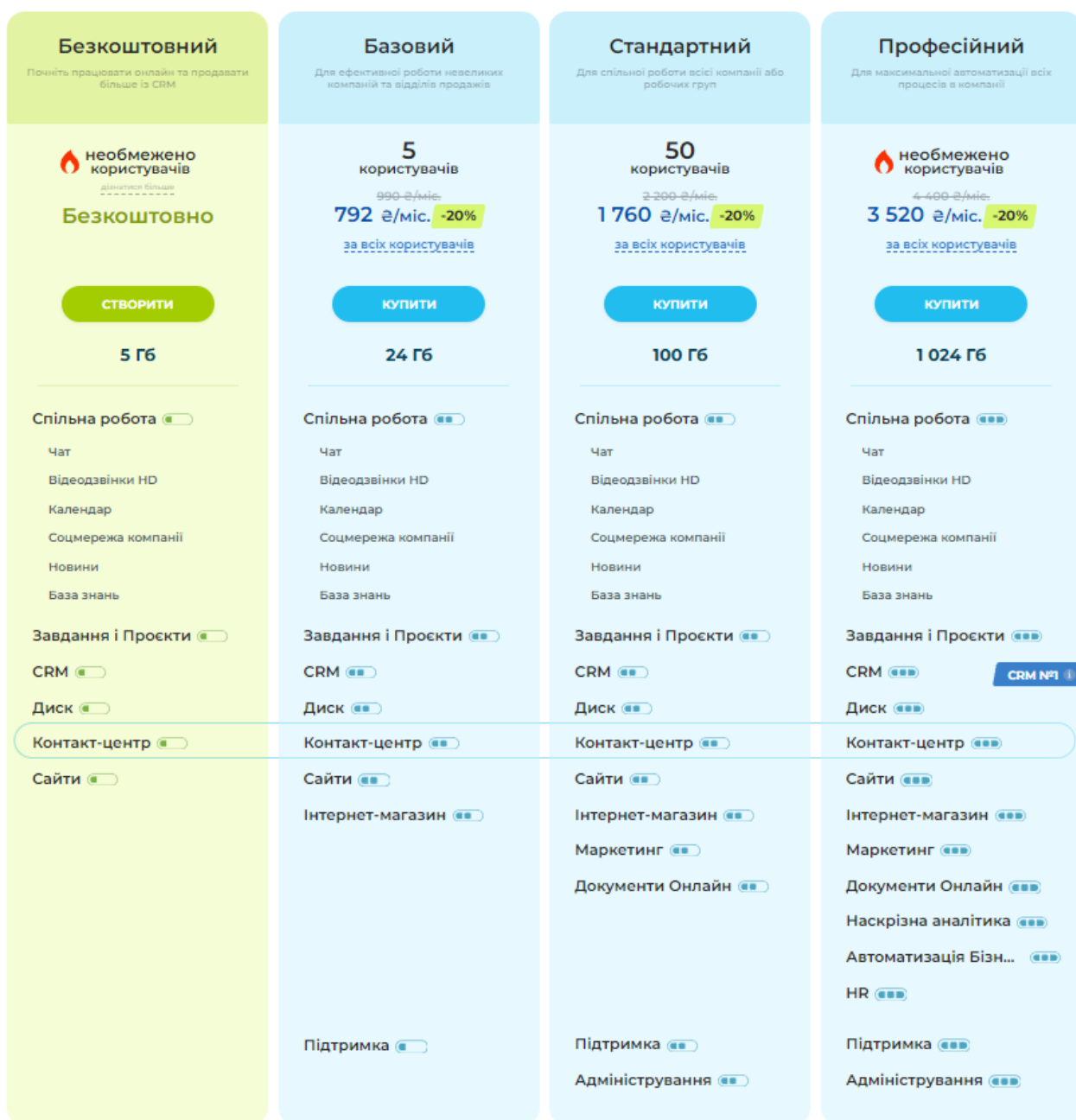
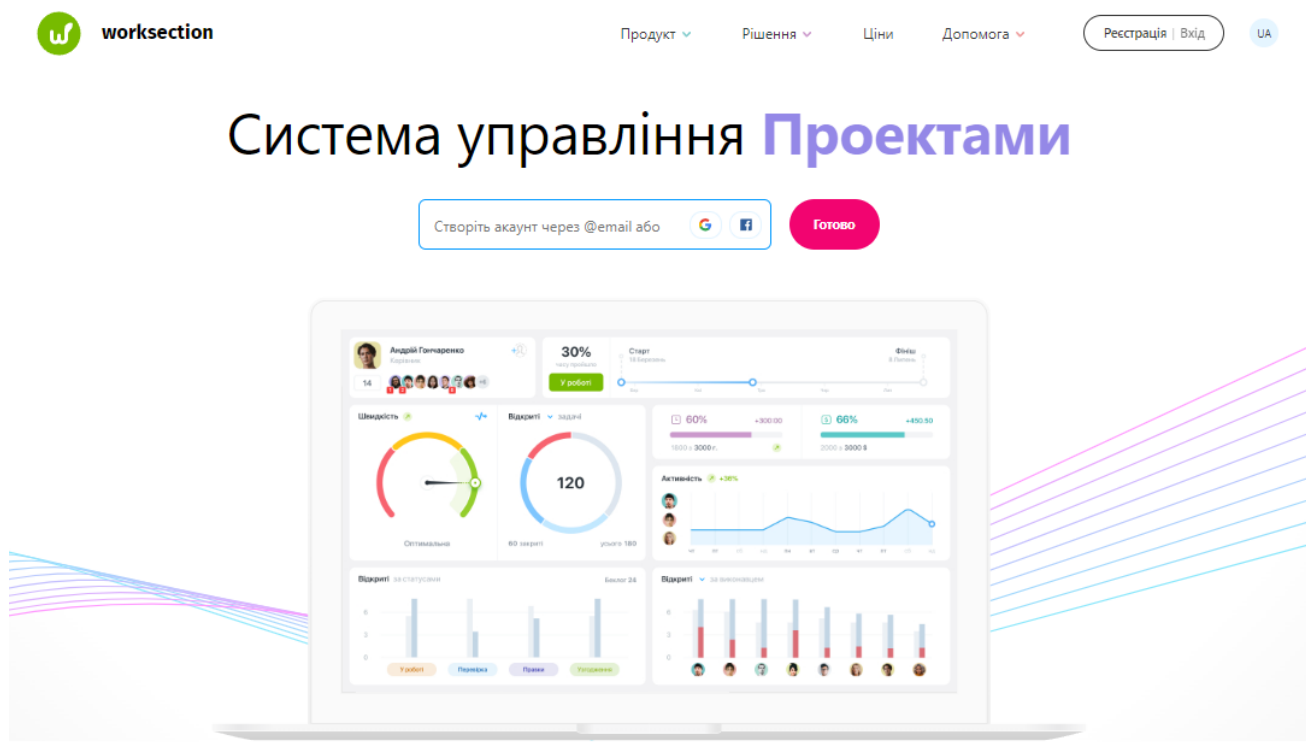


Рисунок 1.5 – Тарифні плани Бітрікс24

Розглянемо наступне програмне забезпечення від українських розробників Worksection.com, головна сторінка сервісу зображена на рисунку 1.6 .



### Система управління проектами для команд

*Worksection дозволяє легко спланувати навіть складний проект. Розділіть його на задачі та підзадачі, встановіть часові рамки задачам на діаграмі Ганта та відстежуйте прогрес і робоче навантаження команди.*

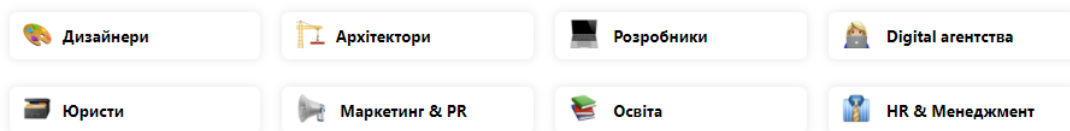


Рисунок 1.6 – Головна сторінка сервісу Worksection.com

Система виступає як CRM система для підприємства, але немає можливості побудови сайтів для клієнтів, також відсутня можливість взаємодії клієнтів із CRM системою. Серед можливостей сервісу слід відзначити наступні:

- можливість використовувати сервіс як хмарне сховище;
- систематизація завдань та процесів на підприємстві;
- контроль термінів виконання замовлення, та персональних завдань працівників;

									Арк.
									17
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ. 180146.01.08.ВД				

- можливість комунікації через внутрішній час сервісу;
- тайм-трекінг;
- можливість створення вибірки із колекцій замовлення та відображення інформації на діаграмі Ганта;
- створення звітностей по окремих замовленнях та за окремі періоди часу.

Сервіс має безкоштовний тарифний план який підходить тільки для ознайомлення з сервісом та навчання оперування на ньому, тарифні плани подані на рисунку 1.7 .

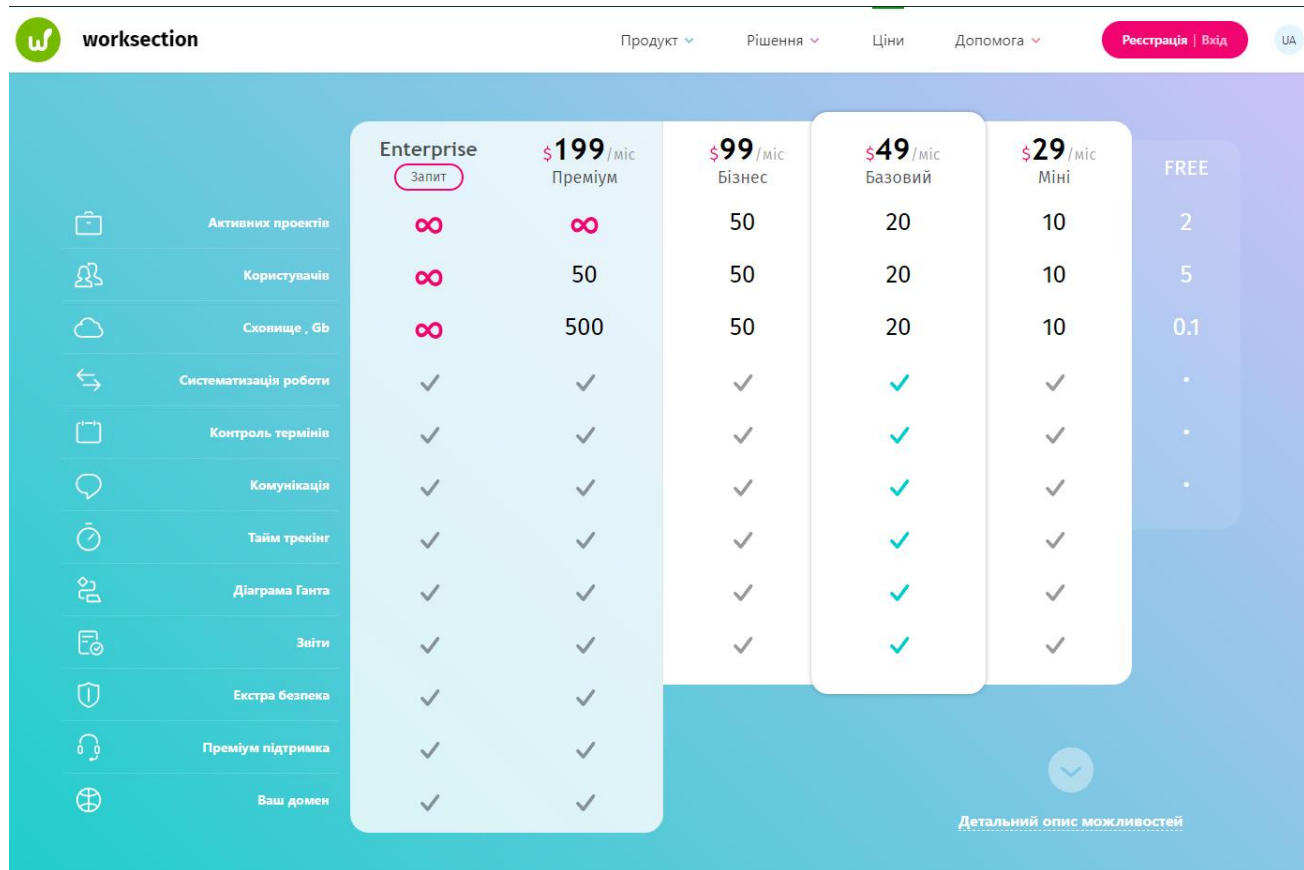


Рисунок 1.7 – Тарифні плани Worksection.com

Слід розглянути і застосунки світових корпорацій, які інтегруються з безліччю інших програмних застосунків. Найбільшим представником, та й найпродуктивнішою системою для малих, середніх та великих бізнесів виступає

програмне забезпечення від компанії Microsoft під назвою Microsoft Dynamics 365 CRM, головна сторінка програмного забезпечення зображена на рисунку 1.8 .

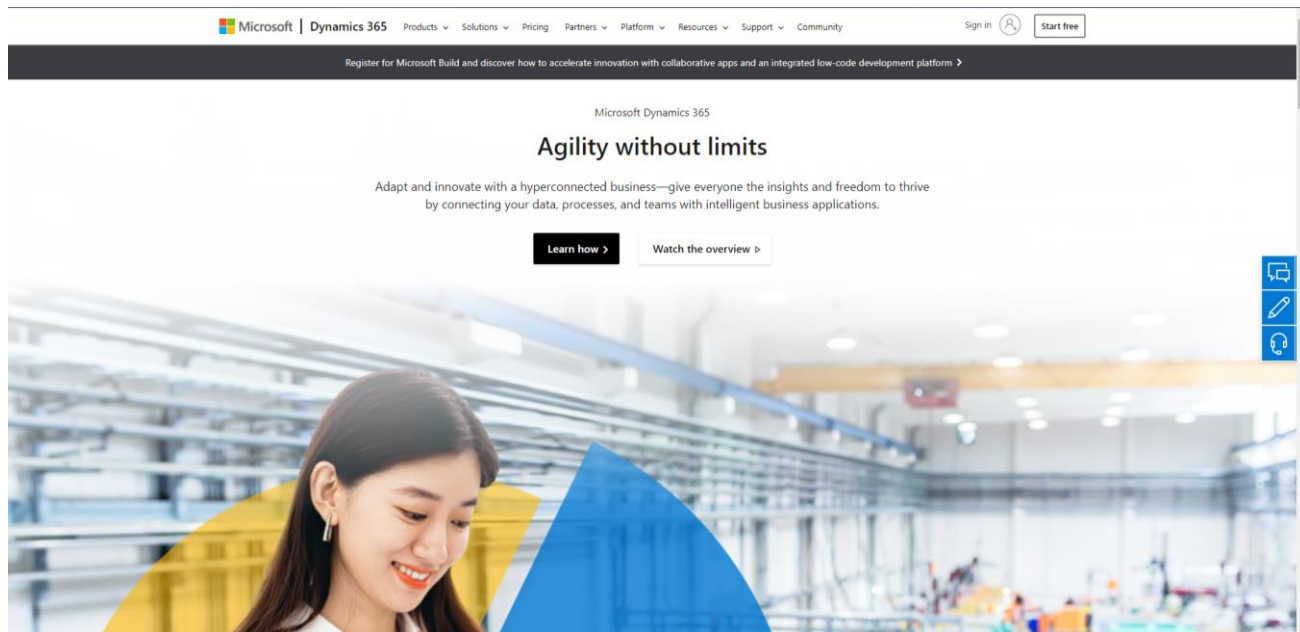


Рисунок 1.8 – Головна сторінка сервісу Microsoft Dynamics 365 CRM

Дане програмне забезпечення від компанії Microsoft надає такі ключові можливості:

- управління контактами із телефонної книги облікового запису Microsoft;
- підтримкою продажів та вбудованих можливостей аналізу для оптимізації продуктивності та збільшення прибутковості;
- вбудованими застосунками для маркетингу та реклами підприємства;
- візуалізація даних на дашбордах із можливістю персоналізації;
- автоматичне оброблення замовлень за допомогою сценаріїв виконання;
- створення шаблонів файлів Word та Excel із заповнення даними із системи;
- Застосунки для працівників під мобільні пристрої на операційних системах ISO та Android.

Слід відзначити що даний програмний продукт надає можливість створення лише CRM системи для підприємства, та немає можливості взаємодії системи із клієнтами. Також система немає конструкторів для побудови сайтів. Тому

										Арк.
										19
Змн.	Арк.	№ докум.	Підпис	Дата						

ДППЗ. 180146.01.08.ВД

підприємство змушене використовувати кілька програмних продуктів для ведення бізнесу.

Сервіс не надає безкоштовних тарифних планів. Також програмне забезпечення чітко поділене на можливості використання під різні типи завдань, на рисунку 1.9 наведено ціни на ліцензію програмного забезпечення та напрямки використання програмного продукту.

The screenshot shows the Microsoft Dynamics 365 website with a navigation bar and a grid of product categories. Below the grid is a pricing table for the Customer Data Platform.

Customer Data Platform	First Dynamics 365 app	Subsequent qualifying <sup>1</sup> Dynamics 365 app
<u>Customer Insights</u>	\$1,500 Per tenant/month	From \$1,000 per tenant/month
<u>Customer Voice</u>	From \$200 <sup>2</sup> 2,000 survey responses per tenant/month*	

<sup>1</sup> Subsequent pricing applies to any tenant in your organization. To learn about qualifying apps and other discounts, see the [licensing guide](#) or [contact us](#).  
<sup>2</sup> When purchased as a standalone application or additional capacity.

Рисунок 1.9 – Тарифні плани та напрямки використання Microsoft Dynamics 365 CRM



забезпечення не надає можливості створення сайту для інформації про власне підприємство.

Сервіс має безкоштовний тарифний план який підходить для ведення малого бізнесу, також слід відзначити не високу вартість у порівнянні з минулими, найпопулярніші тарифні плани наведено на рисунку 1.11 також вказано основний функціонал програмного забезпечення.

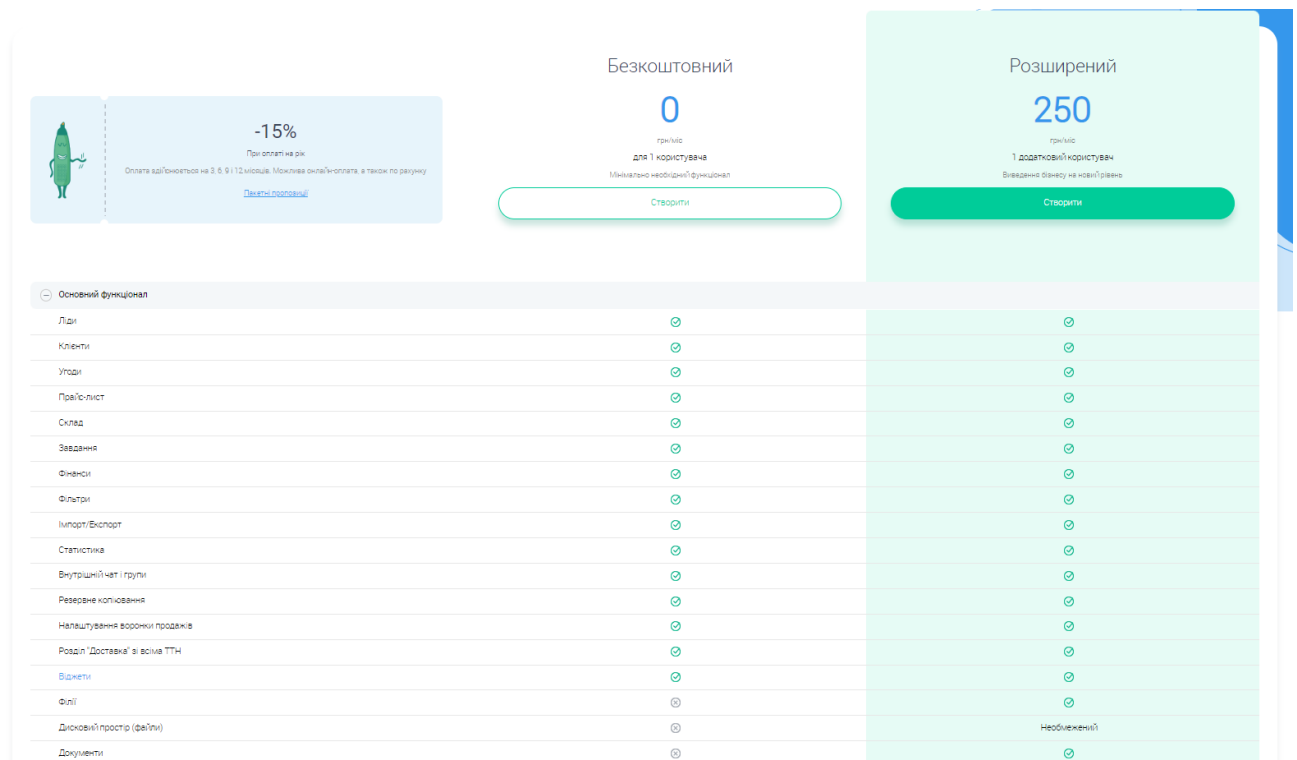


Рисунок 1.11 – Тарифні плани та основний функціонал сервісу Keerincrm

Проведений аналіз схожого програмного забезпечення показав, що на теперішній час мало аналогів на ринку які включають усі потрібні функції. Також керуючись аналізом визначено, що розроблюваний застосунок повинен:

- включати сторінку за інформацію про підприємство;
- мати вбудовану та спеціалізовану систему CRM;
- надавати клієнтам інформацію із системи CRM;
- містити методи керування персоналом;
- містити методи керування фінансами;

- зберігати всі замовлення які були отриманні підприємством та інформацію про користувачів, та виплати;
- застосунок повинен інтегруватися до різних форматів екранів;
- бути максимально надійним;
- мати простий інтерфейс та бути інтуїтивно зрозумілим.

### 1.3 Визначення вимог до інтернет-платформи із системою CRM

Опишемо вимоги розроблюваної інтернет- платформи та продеменструємо поведінку системи та можливості її використання відповідними користувачами. Діаграма складається з користувачів(акторів) та випадків користування елементами програмного застосунку і зв'язок який здійснюється при цьому в системі з іншими елементами. Покажемо які відбуваються зв'язки при використанні елементів програми, та продемонструємо можливості взаємодії з програмним забезпеченням.

Актор це будь-який об'єкт який взаємодіє з елементом програми за допомогою випадків використання. Діаграма відображає набір дій які здійснюється програмою для виконання завдання.

Таблиця 1.1 - Опис взаємодії акторів

Актор	Короткий опис
Гість	Виконує перегляд веб-сторінки СТО
Клієнт	Створює нові замовлення, переглядає статус виконання замовлення, переглядає список усіх замовлень, редагує особисту інформацію.
Працівник	Створює нові замовлення, переглядає статус виконання замовлення, переглядає список усіх замовлень, редагує особисту інформацію, отримає інформацію про нарахування заробітної платні.



## Кінець таблиці 1.1

Сервер	Реєструє або авторизує користувача, обробляє вхідні запити, вичитує інформацію з бази даних, записує інформацію в базу даних, надсилає відповідь на вхідний запит.
--------	--

Будуємо діаграму, на якій зобразимо найбільш популярні варіанти використання (див. додаток А).

Таблиця 1.2 - Головний розділ сценарію виконання варіанта використання

Варіанти використання	Перегляд сторінок веб-застосунку. Авторизація та автентифікація. Створення нового замовлення. Редагування замовлення. Редагування статусу користувача. Створення виплати. Редагування виплати. Запис в базу даних. Читання інформації з бази даних. Редагування інформації в базі даних. Видалення інформації з бази даних.
Актори	Гість, користувач, працівник, менеджер, кадровик, бухгалтер, директор, сервер.
Ціль	Введення звітності підприємства. Приймання нових замовлень. Контроль виконання замовлення.
Короткий опис	Гість авторизується на сервісі та стає користувачем. Клієнт створює замовлення. Менеджер керує інформацією замовлення. Бухгалтер створює виплати. Кадровик змінює статус користувача на працівника та встановлює посаду та навпаки. Директор виконує роботу менеджера, кадровика, бухгалтера. Сервер опрацьовує вхідну інформацію та надає відповіді на запити.
Тип	Базовий

Сервіс можна використати не тільки за даним сценарієм, адже він містить функції які дозволять використовувати його тільки для окремих завдань. Наприклад це може бути використання CRM системи або сторінки для бухгалтерського обліку.

Таблиця 1.3 - Типовий хід подій сценарію виконання варіанту

використання

Актори	Відклик системи
1. Гість входить на головну початкову сторінку сайту, або на профіль користувача .	2. Система відкриває вікно для реєстрації користувача.
3. Гість реєструється. Виняток №1: Невірна інформація	4. Система створює особистий профіль користувача.
5. Користувач створює замовлення.	6. Система опрацьовує та зберігає замовлення в базі даних.
7. Менеджер редагує замовлення. Додає працівників для виконання. Змінює статус замовлення	8. Система вносить зміни в базу даних.
9. Бухгалтер створює виплати на покупку запчастин та виплату заробітної плати.	10. Система вносить зміни у базу даних. Система рахує витрати та прибуток та визначає чистий дохід.

Таблиця 1.4 - Розділ виключення сценарію виконання варіанта виконання

Виняток №1: Невірна інформація	
Дії акторів	Відклик системи
	Система відмовляє у створенні облікового запису та повертає на форму введення інформації.

Проведено збір інформації подібних програмних застосунків та виконано аналіз. В результаті аналізу створено вимоги для власного програмного забезпечення із всіма недоліками та перевагами. Після аналізу проблематики

завдання створено діаграму використання програмного забезпечення із покращенням ергономіки в порівнянні до інших сторонніх програмних продуктів.

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

## 2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-ПЛАТФОРМИ

### 2.1 Архітектура та функціональна структура додатку

При розробці інтернет платформ використовують декілька основних архітектур побудови програмного забезпечення. Вибір архітектури здебільшого залежить від завдань які повинне виконувати ПЗ. Найпопулярнішою архітектурою розробки Web додатків стала архітектура MVC, адже вона найбільш універсальна та дозволяє пристосуватися під різні завдання виконання в програмних системах.

MVC архітектура, повна назва Model-View-Controller рисунок 2.1, становить загальноприйнятий стандарт у світі розробок Web застосунків. Шаблон дозволяє створювати зрозумілу логіку програмного забезпечення та поділяє програмне забезпечення на три рівня:

Модель – це об’єкт який містить задану кількість полів для інформації. Модель використовується для запису, зчитування, редагування, видалення інформації з БД. Також для формування запитів, наприклад JSON. Моделі використовуються для зберігання інформації яка може слугувати для опрацювання логіки програми без запису інформації з моделей в БД.

Представлення – це рівень який відповідає за відображення графічної складової Web застосунку. Також окрім графічного виведення доступні можливості виведення різних типів файлів.

Контролер – це рівень який містить логіку формування інформації конкретного представлення. Для кожного представлення використовується окремий контролер.

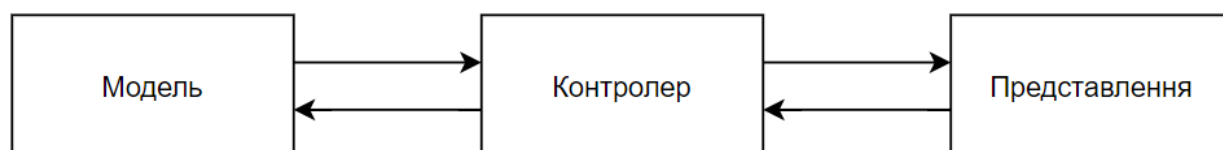


Рисунок 2.1 – Схема MVC шаблону

Усі Web застосунки використовують клієнт-серверну архітектуру. Це архітектура яка включає у себе серверну частину програми яка відтворюється на серверному комп'ютері. Сервер містить у собі весь програмний продукт та повністю опрацьовує логіку програми. Клієнтом виступає веб-браузер користувача. Взаємодія між сервером та клієнтом відбувається за допомогою спільної мережі. Для спілкування даних частин зазвичай використовуються протоколи http та https. У ролі сервера виступає один програмний продукт який відповідає за опрацювання логіки програми. Натомість клієнтів може бути необмежена кількість. На рисунку 2.2 зображено графічне приставлення архітектури.

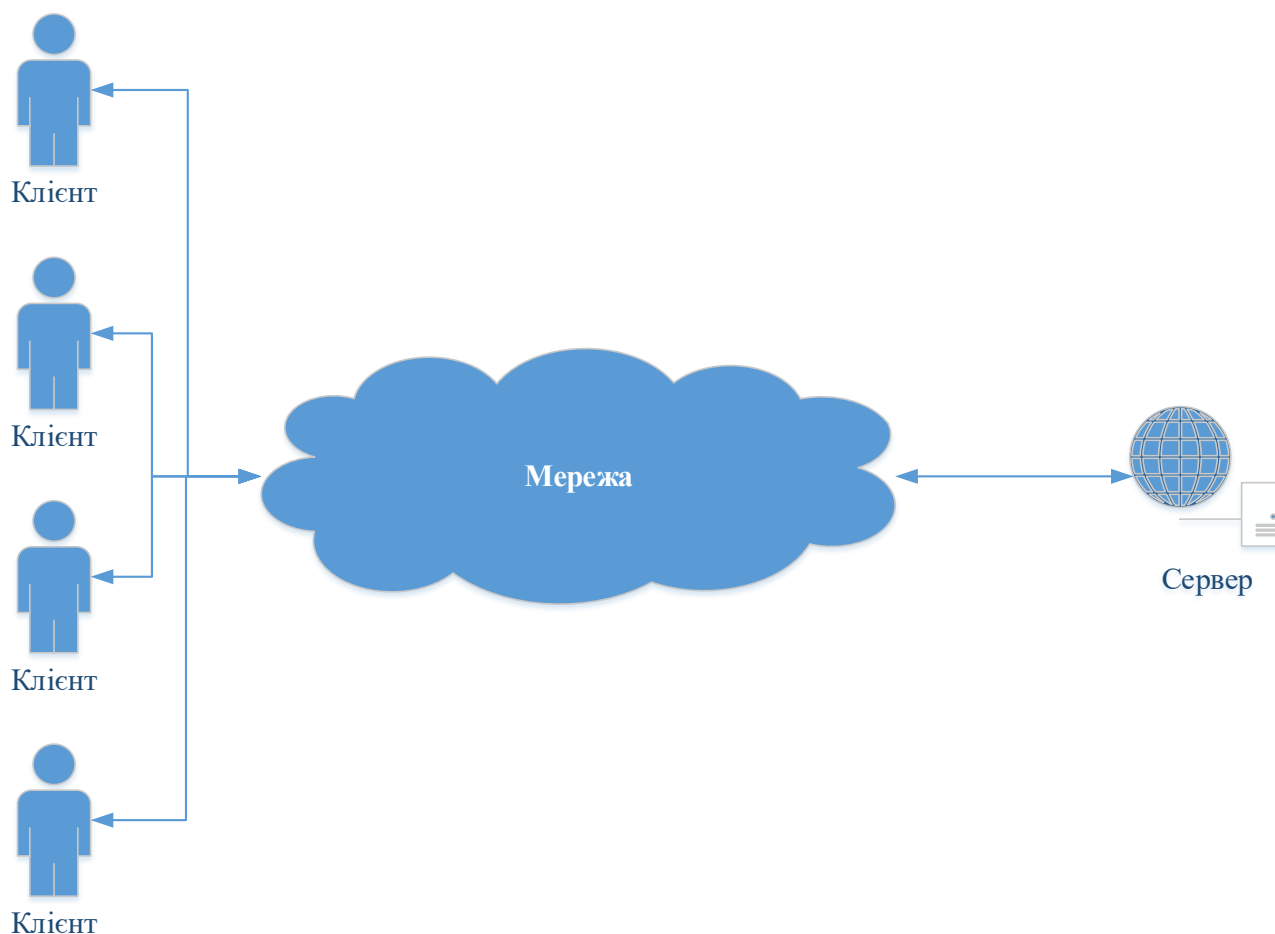


Рисунок 2.2 – Схема клієнт-серверної архітектури

Для розподілу навантаження на серверну частину, можливість розподілити навантаження на декілька комп'ютерів використовується мікросервісна

архітектура. Як і в клієнт-серверній архітектурі може бути безліч клієнтів які спілкуються через мережу з центральним мікросервісом API. Програмний продукт розділяється на менші програми які спілкуються між собою в різних мережах зазвичай через протоколи http та https, використовуючи RPC методи. Дана архітектура окрім можливості розподілу навантаження між серверними комп'ютерами дозволяє масштабувати програмний продукт, адже для нової функції потрібно створити мікросервіс та редагувати лише центральний мікросервіс для взаємодії, що у свою чергу дозволяє не вносити зміни в усі елементи програмного забезпечення. На рисунку 2.3 зображено графічне представлення мікросервісної архітектури.

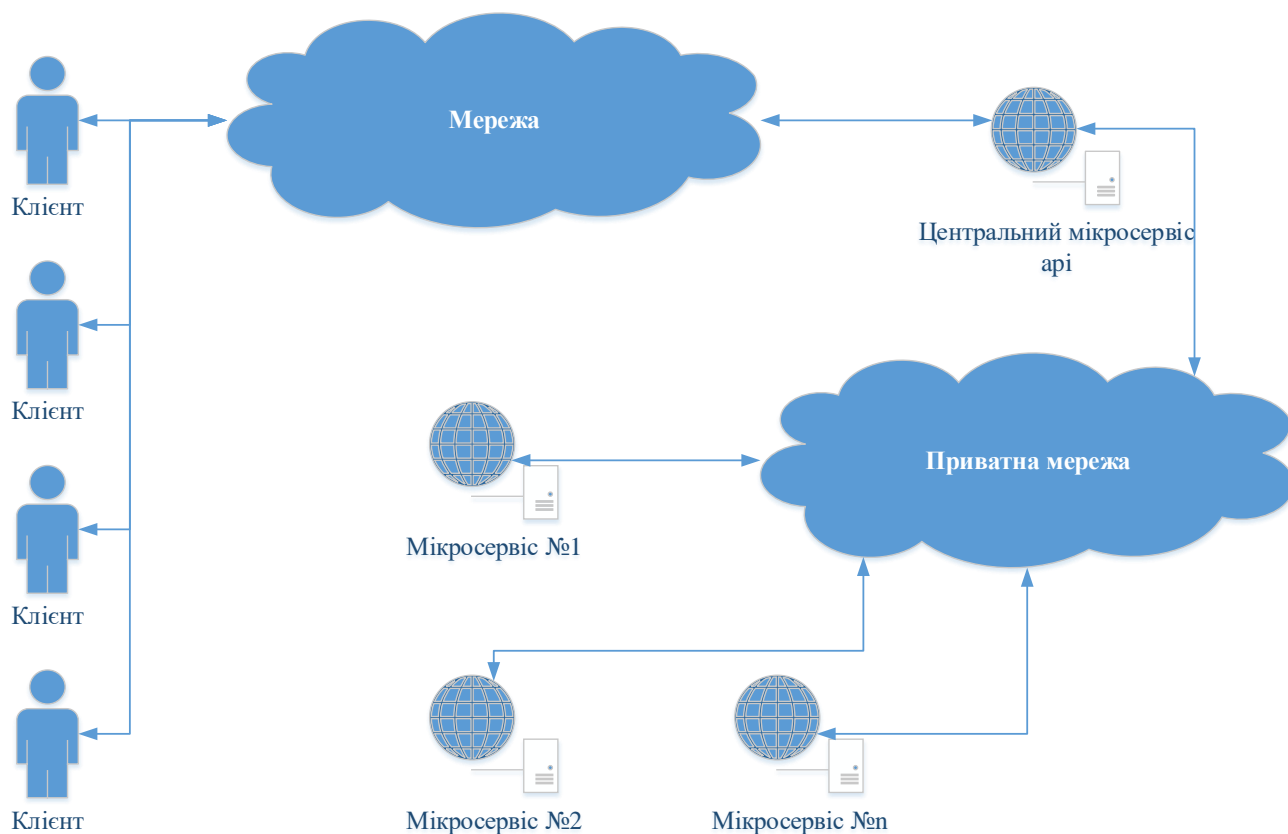


Рисунок 2.3 – Схема мікросервісної архітектури

HTTP – це протокол, який використовуються для передавання даних в форматі гіпертекстових повідомлень. Для кожного клієнта створюється особистий сеанс із сервером. Протокол дозволяє передавати вебсторінки та

текстові файли. Сеанс створюється під час з'єднання клієнта із сервером, та розривається якщо клієнтська сторона певний час, який встановлений серверним програмним забезпеченням, не надсилає запити серверу. Http протокол складається із таких частин:

- заголовка (headers);
- рядка запиту (body).

HTTP протокол містить дані методи взаємодії:

- GET – метод для отримання даних;
- POST – метод для надсилання даних;
- PUT – метод для редагування даних;
- DELETE – метод для видалення даних.

Для безпечного передавання даних використовуються HTTPS протокол. У порівнянні з HTTP протоколом, містить додатковий шар шифрування інформації. Даний шар створюється за допомогою SSL сертифікатів.

SSL протокол – це криптографічний протокол, який слугує для створення безпечного з'єднання між клієнтом та сервером. Та для забезпечення більшої надійності використовується TLS протокол який розроблений на основах SSL. SSL протокол невідмінно від SSL використовує асиметричне шифрування і сертифікати x.509.

Для взаємодії між мікросервісами використовується RPC протокол. RPC протокол використовує HTTP та HTTPS протоколи для встановлення з'єднання та передаванні інформації. Інформація передається у вигляді JSON об'єктів. Та отримує відповідь у формі JSON об'єкта.

Названі архітектури та протоколи ідеально підходять для використання в розроблювальному програмному продукті. Взаємодія клієнтів з програмним забезпеченням буде відбуватися за клієнт-серверною архітектурою та використовувати протокол HTTP для зв'язку та створення сесії. Побудова програмного забезпечення використає мікросервісну архітектуру, та для зв'язку у приватній мережі буде використовуватися RPC протокол, також за допомогою

										ДППЗ. 180146.01.08.ВД	Арк.
											31
Змн.	Арк.	№ докум.	Підпис	Дата							

RPC протоколу створюється підключення та взаємодія мікросервісів з базами даних.

## 2.2 Аналіз продукту для авторизації та аутентифікації Keycloak

Keycloak – продукт з відкритим кодом що впроваджує готові рішення для впровадження авторизації та аутентифікації за допомогою технології single sign-on, на рисунку 2.4 зображено головну сторінку програмного продукту.

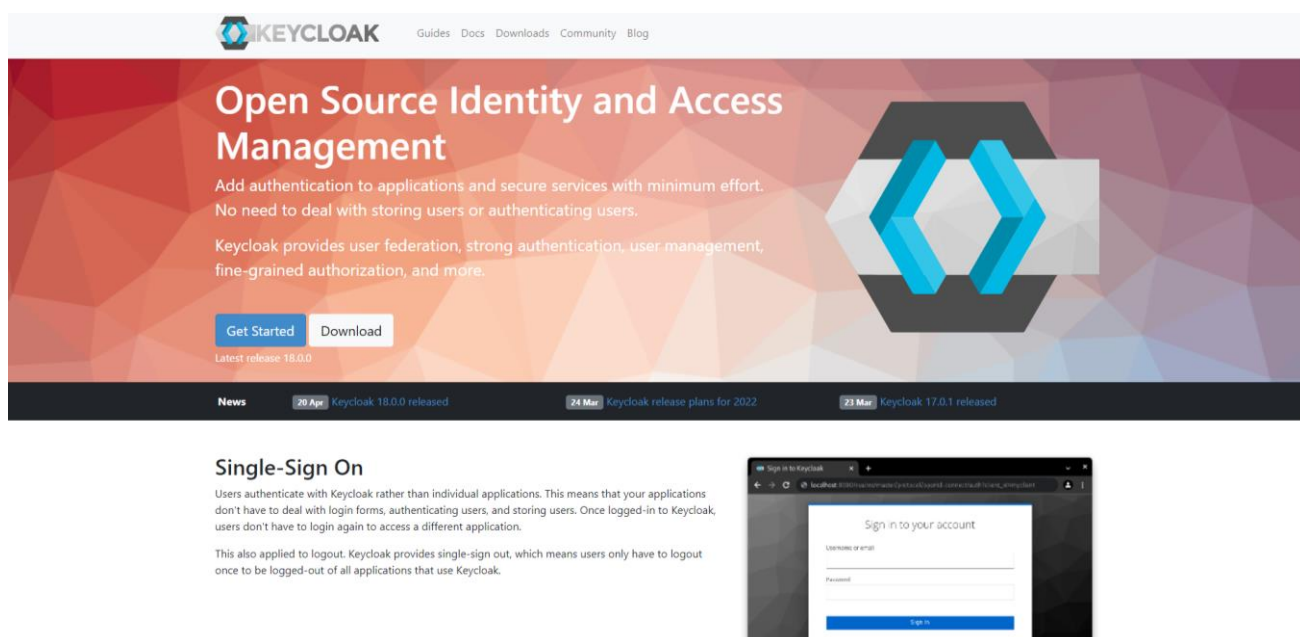


Рисунок 2.4 – Головна сторінка програмного продукту Keycloak

Програмний продукт націлений для впровадження у сучасних застосуваннях та сервісах. Метою застосунку виступає створення безпечних застосунків та сервісів із мінімальним написанням коду, та використання найсучасніших методів для надання безпеки. Keycloak виступає провайдером OAuth та дозволяє створювати OpenID Connect та використовує Security Assertion Markup Language.

Keycloak дає можливість запису інформації про облікові записи у даних базах даних:

					ДППЗ. 180146.01.08.ВД	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

- MarianDB;
- MsSql;
- MySql;
- Oracle DB;
- Postgres.

Основні функції ПЗ:

- реєстрація нових облікових записів;
- аутентифікація користувачів;
- авторизації та аутентифікація через соціальні мережі;
- Single Sign-On та Sign-Off для всіх мікросервісів одного реалму;
- видача JSON Web Token авторизованим користувачам;
- двохфакторна перевірка;
- інтеграція з LDAP протоколом;
- використання брокера Kerberos;
- персоналізація зовнішнього вигляду сторінок сервісу.

Keusloak складається із двох елементів:

- сервера;
- адаптера для програмного забезпечення.

Програмний продукт активно розвивається та підхоплює усі тенденції для захисту програмного забезпечення та конфіденційності користувачів. Активно розвивається та створює адаптери під різні мови програмування та фреймворки. Документація Google Console рекомендує використовувати дане програмне забезпечення, також Spring-Security засилається на використання Keusloak при розробці OAuth систем авторизації, та більше не розвиває власні методи взаємодії OAuth.

LDAP протокол – це полегшений протокол доступу до каталогів, входить до типу комунікаційних протоколів, який визначає методи в яких служба каталогів доступна та може бути доступною, на рисунку 2.5 графічно зображено роботу протоколу. LDAP формує спосіб, яким дані всередині служби директорії повинні бути представлені користувачам. Визначаються усі вимоги до компонентів які

									Арк.
									33
Змн.	Арк.	№ докум.	Підпис	Дата					

використовуються що для створення записів даних усередині служби директорії. Також описує спосіб яким різні примітивні елементи використовуються для складання записів.

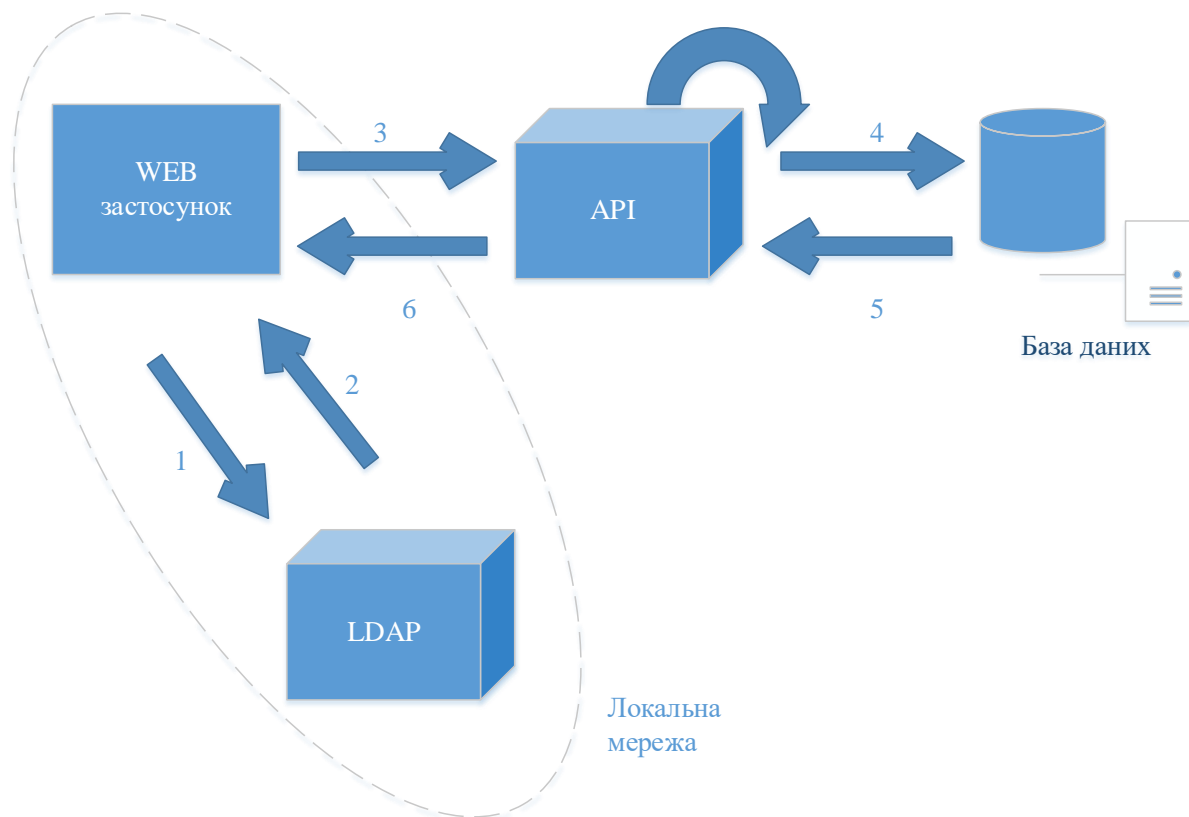


Рисунок 2.5 – Візуалізація роботи LDAP протоколу

Kerberos протокол – орієнтований для використання в клієнт-серверній архітектурі. Надає можливість взаємної аутентифікації клієнта із сервером в умовах небезпечного підключення, на рисунку 2.6 візуально продемонстровано роботу протоколу. В основу протоколу покладений протокол Нідгема-Шредера, тобто у ролі довіреної сторони виступає Key Distribution Center. Kerberos протокол реалізує такі принципи:

- пароль користувача ніколи не передається по мережі;
- пароль користувача ніколи не зберігається на клієнтській машині;

- пароль користувача зберігається тільки у зашифрованому вигляді;
- використання Single Sign-On;
- управління аутентифікацією здійснюється централізовано сервером аутентифікації;
- виконання процесу “Взаємної аутентифікації”;
- встановлення зашифрованого зв’язку між клієнтом та сервером.

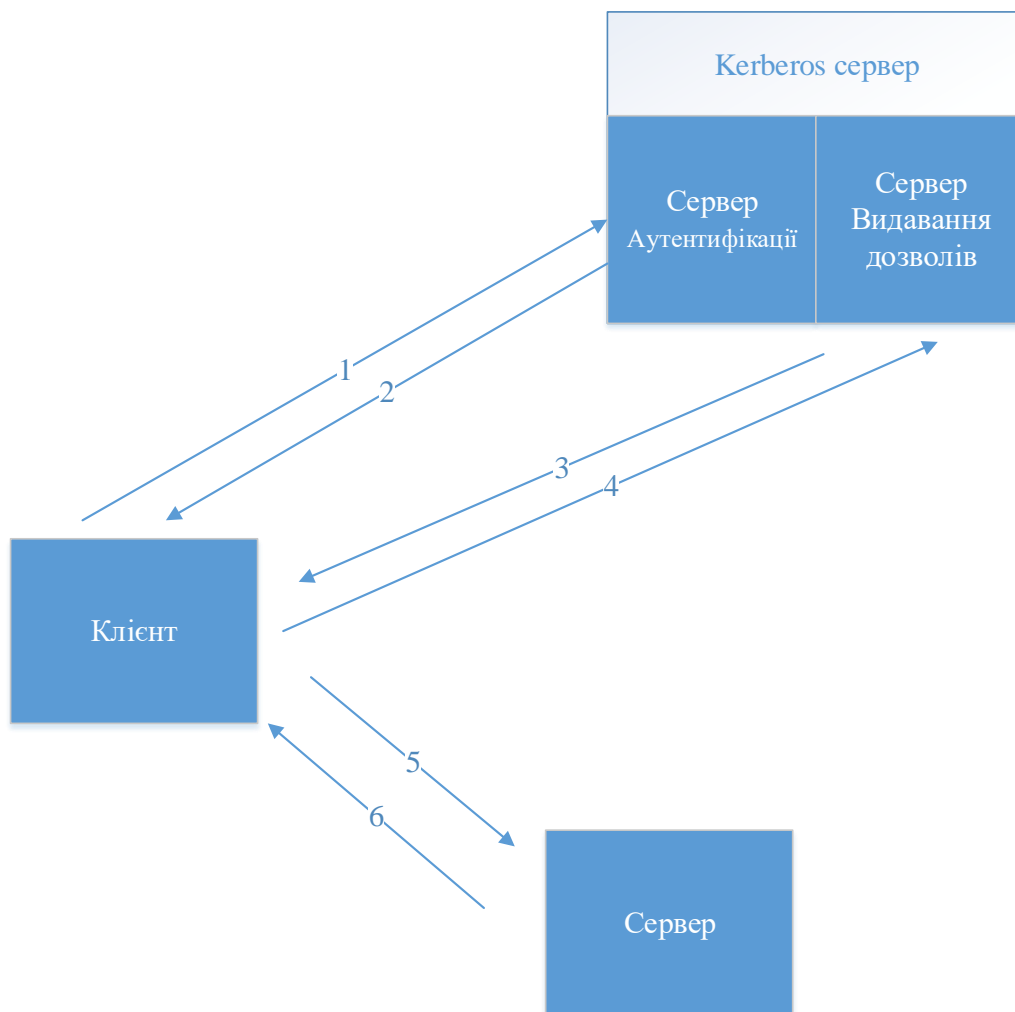


Рисунок 2.6 – Візуалізація роботи Kerberos протоколу

Даний програмний продукт ідеально підходить для впровадження у розроблювальному програмному продукті. Також присутній офіційний адаптер для фреймворка Spring. Гармонійне поєднання Spring-Security та Keycloak надають рішення для безпеки програмного забезпечення за найновішими стандартами та практиками захисту. Ключовим моментом виступає те що

програмний продукт встановлюється на сервер розроблювального програмного продукту. Також Keycloak розроблений із можливістю використання в мікросервісних архітектурах та забезпечує єдину авторизацію для всіх сервісів у системі.

### 2.3 Визначення основних модулів застосунку

Для реалізації застосунку завдяки попередньому аналізу функціональної структури, визначено чотири основних рівнів будови застосунку:

- рівень даних;
- рівень захисту;
- рівень застосунку;
- рівень користувацького інтерфейсу.

Рівень даних включає у себе створення БД для зберігання усієї потрібної інформації.

Рівень захисту виконується як окремий мікросервіс для виконання авторизації та аутентифікації користувача. Також мікросервіс перевіряє сесію та токени доступу для визначення рівня доступу для взаємодії із програмним продуктом. Виконує захист від зловмисних атак на ПЗ.

Рівень застосунку у архітектурі MVC та у фреймворку Spring відносять код який не входить у рівні MVC. Тобто рівень застосунку складає залежний від інтернет-платформи код. Сюди будуть відноситися усі імплементації та сервіси, а саме:

- сервіс для оперування інформацією з БД для використання у головній сторінці інтернет-платформи;
- сервіс для персональної сторінки, який створює параметри для роботи методів та функцій на персональній сторінці. Також даний сервіс створює можливість переходу на особисту сторінку за ідентичним адресом для всіх користувачів. Сервіс заповнює DTO для запису відгуків клієнтів у БД;

						ДППЗ. 180146.01.08.ВД	Арк.
							36
Змн.	Арк.	№ докум.	Підпис	Дата			

– сервіс обробки замовлень, його ключова мета полягає і перевірці правильності даних при створенні замовлення. Ордер оперує системними даними серверного комп’ютера та додає потрібні значення у DTO для подальшого запису в базу даних, також сервіс перевіряє правильність заданих значень DTO;

– сервіс бухгалтерських операцій, мета сервісу формувати та перевіряти правильність транзакцій, заповнення потрібно інформацією із системи. Сервіс надає можливість створення виплати в застосунку. Також сервіс використовується для читання та сумування інформації, створення статистики про прибутки та витрати підприємства;

– сервіс логування, сервіс заповнює DTO та створює запис логу в базу даних;

– сервіс безпеки, виконує перевірку користувача та встановлює рівень доступу для взаємодії із програмним продуктом. Також створює нові облікові записи у системі, містить методи для повернення користувача на особисту сторінку при такій потребі.

Рівень користувацького інтерфейсу відповідає за відображення графічного інтерфейсу користувача. також за взаємодію користувача із інтернет-платформою через графічний інтерфейс.

Діаграму модулів застосунку та їх взаємодію наведено на рисунку 2.7.

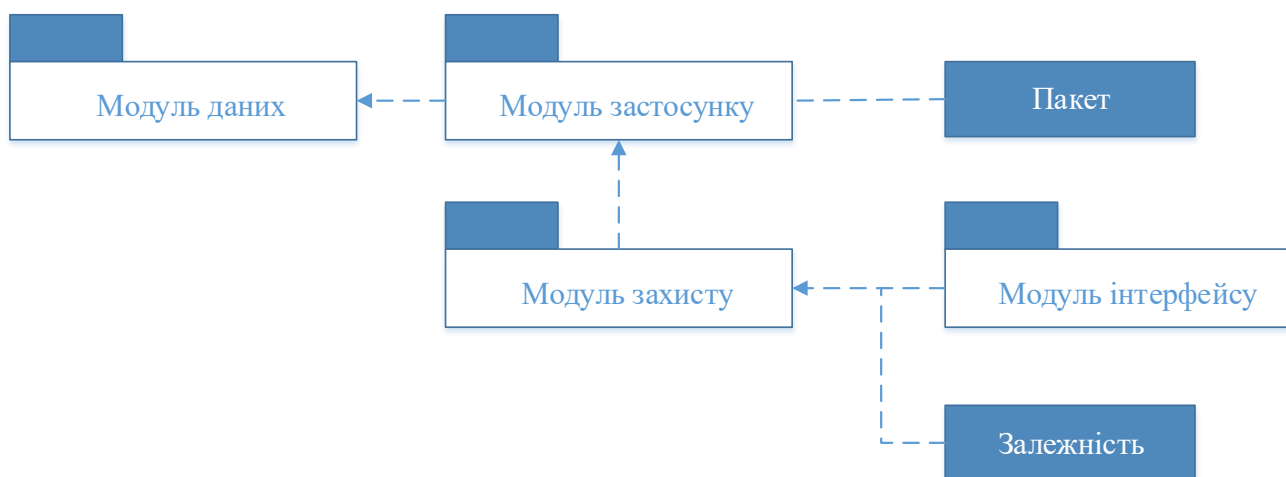


Рисунок 2.6 – Модулі застосунку

Сервіси програмного забезпечення використовують ін'єкцію задля поєднання із контролерами. Для створення ін'єкцію у фреймворку Spring використовується анотація @Autowired та при цьому сервіс має використовувати анотацію @Service, також є можливість використовувати сервіси без використання анотацій фреймворку Spring, користуючись тільки можливостями Java, а саме за допомогою інтерфейсів.

## 2.4 Проектування моделі бази даних

Інтернет-платформа буде використовувати дві бази даних реляційну MySQL та нереляційну MongoDB. Реляційна база даних використовується для опрацювання складних запитів, забезпечують гнучкість керування та дають можливість аналізу даних. Нереляційна база даних добре працює із великою кількістю інформації і має більшу стійкість до навантаження та більшу пропускну здатність.

MySQL слугуватиме сховищем для мікросервісу Keycloak. Реляційна база даних для зберігання інформації використовує таблиці. В БД записуються всі системні налаштування застосунку авторизації. Окрім того застосунок створює таблиці для запису інформацій та параметрів створених Realm. Усі дані за профілі користувачів із зашифрованими паролями теж записуються у базу даних.

MongoDB слугуватиме сховищем для мікросервісу інтернет платформи. Для запису в нереалізаційні бази даних використовуються документи у форматах JSON, XML, YAML, BSON. Відповідно до завдання та потреби зберігання інформації в БД будуть використовуватися дані колекції:

- Users;
- Transactions;
- ServicesList;
- Reviews;
- Partners;

						ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			38

- OrdersWorkTeam;
- Orders;
- OrderStatus;
- OrdersCheck;
- Logs;

Колекція «Users» призначення для зберігання інформації про облікові записи користувачі сервісу, та містить такі поля:

- \_id – id запису у колекції БД;
- name – ім'я користувача;
- personalIdent – унікальний ідифікатор, логін;
- phone – мобільний номер телефону;
- email – електронна адреса користувача;
- status – приналежність профілю до групи клієнтів або працівників;
- role – посада на підприємстві;
- photo\_url – посилання на світлинку профілю;
- \_class – посилання на DTO.

Колекція «Transactions» використовується для зберігання інформації про фінансові операції, поля колекції:

- \_id – id запису у колекції БД;
- number – унікальний номер замовлення;
- username – користувач якому здійснюється зачислення коштів;
- type – призначення виплати;
- clocks\_work – кількість пропрацьованих годин;
- info – коротка інформація про виплату;
- full\_info – додаткова інформація про виплату;
- money – кількість коштів;
- usernameCreator – унікальний ідифікатор працівника який створив виплату:

- \_id – id запису у колекції БД;
- date – повна дата створення виплати у типі запису timestamp

- logicDate – дата створення виплати у вигляді рік-місяць-день;
- link – посилання на сторінку з інформацією про фінансову операцію
- \_class – посилання на DTO.

Колекція «ServicesList» містить інформацію про послуги які надає підприємство та складається із полів:

- \_id – id запису у колекції БД;
- name – назва послуги;
- text – інформація про послугу;
- price – ціна виконання замовлення;
- image\_url – посилання на картинку послуги;

Колекція «Reviews» зберігає в БД відгуки клієнтів про підприємство, складається із полів:

- \_id – id запису у колекції БД;
- text – текст відгуку;
- timestamp – дата створення відгуку у форматі та вигляді Timestamp;
- username – персональний ідентифікатор користувача який створив відгук;
- user\_image – посилання на світлинку профілю користувача який створив відгук;
- \_class – посилання на DTO;
- logicId – унікальний номер відгуку.

Колекція «Partners» містить інформацію про партнерів підприємства, дана колекція складається із полів:

- \_id – id запису у колекції БД;
- image\_url – посилання на зображення про партнерську компанію;
- name – назву компанії;
- text – короткий опис компанії.

Колекція «OrdersWorkTeam» зберігає інформацію про виконавців замовлення, складається із полів:

- \_id – id запису у колекції БД;
- numberOrders – унікальний номер замовлення;

										ДППЗ. 180146.01.08.ВД	Арк.
											40
Змн.	Арк.	№ докум.	Підпис	Дата							

- name – персональний ідифікатор виконавця;
- statusUser – посада виконавця;
- link – посилання на особисту сторінку виконавця;
- \_class – посилання на DTO.

Колекція «Orders» містить інформацію про деталі замовлення які записуються у поля:

- \_id – id запису у колекції БД;
- orderNumber – унікальний номер замовлення;
- username – унікальний ідифікатор профіля замовника;
- brandCar – марка автомобіля;
- modelCar – модель автомобіля;
- yearsCar – дата виготовлення автомобіля;
- service – номер послуги по ремонту;
- price – ціна виконання замовлення;
- dateCreateOrder – дата створення замовлення;
- dateWork – дата виконання замовлення;
- phoneCall – інформація про зв'язок з клієнтом, розмова по телефону або повідомленнями в месенджері.

– Message – коментар замовника який вказується за бажанням при створені замовлення.

- statusWork – рівень виконання замовлення;
- archive – інформація чи замовлення активне або перенесене в архів.
- link – посилання на сторінку з інформацією про замовлення;
- netProfit – чистий прибуток підприємства від виконання замовлення
- \_class – посилання на DTO.

Колекція «OrderStatus» включає в себе інформацію про виконання замовлення для використання в CRM системі, колекція складається із полів:

- \_id – id запису у колекції БД;
- numberStatus – унікальний ідифікатор стану виконання;
- status – статус виконання замовлення;

					ДППЗ. 180146.01.08.ВД	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

– `_class` – посилання на DTO.

Колекція «OrdersCheck» містить у собі інформацію про витрати при виконанні замовлення та складається із полів:

- `_id` – id запису у колекції БД;
- `numberOrder` – унікальний ідифікатор замовлення;
- `name` – назва витрати;
- `numberCheck` – унікальний ідифікатор окремої витрати;
- `_class` – посилання на DTO.

Колекція «Logs» включає у себе інформацію про взаємодії клієнтів із сервісом та побудована із даних полів:

- `_id` – id запису у колекції БД;
- `type` – тип логування;
- `name` – назва дії користувача на сервісі;
- `title` – детальна інформація про дію;
- `message` – текст логу;
- `date` – дата виконання дії у вигляді формату Timestamp;
- `logicDateDays` – дата яка використовується для логіки програми у вигляді рік-місяць-день;
- `_class` – посилання на DTO.

## 2.5 Проектування графічного інтерфейсу користувача

Дизайн інтерфейсу виконано у трьох основних кольорах:

- жовтий;
- білий;
- чорний.

Завдання інтерфейсу бути ергономічним, мати максимально легкий дизайн інтерфейсу. Розглянемо головну сторінку інтернет платформи по блоках.

						ДППЗ. 180146.01.08.ВД	Арк.
							42
Змн.	Арк.	№ докум.	Підпис	Дата			

На рисунку 2.7 зображений меню взаємодії з сервісом та перший блок головної сторінки який включає у себе банер на якому розміщено агітаційні картинки для приваблювання нових клієнтів.

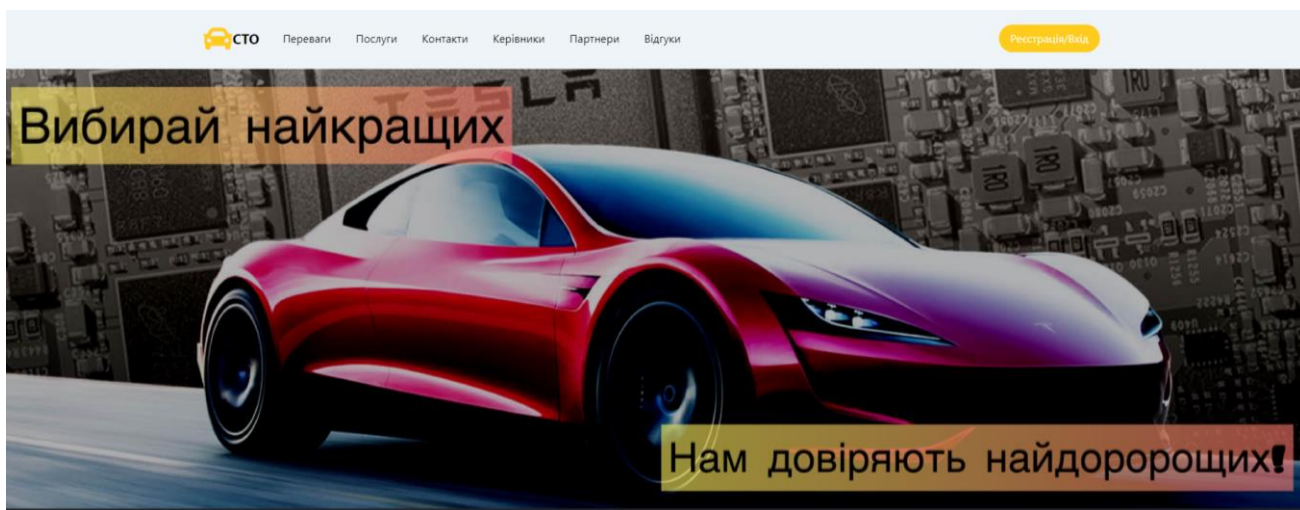


Рисунок 2.7 – Меню взаємодії з сервісом та перший блок головної сторінки

На другому по черзі блокові інтерфейсу розміщено інформацію про переваги підприємства, рисунок 2.8.

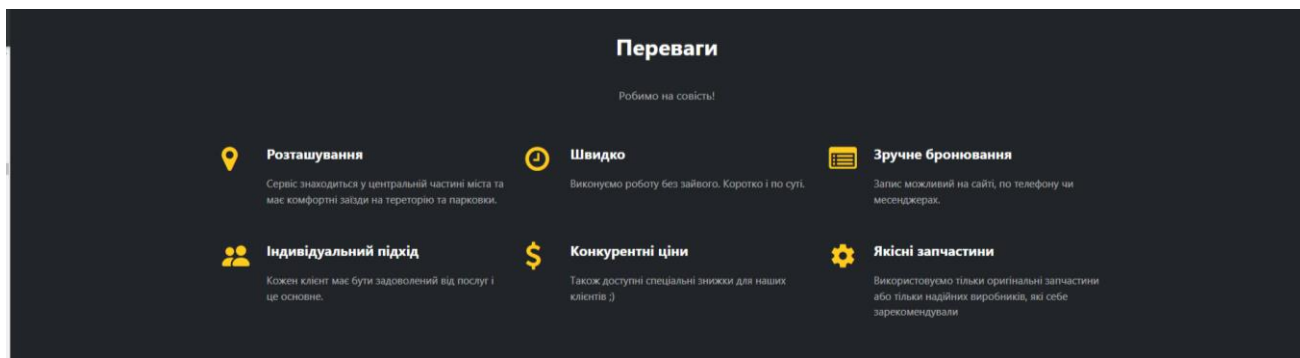


Рисунок 2.8 – Другий блок головної сторінки

Третій блок головної сторінки містить список послуг які надає підприємство. Колонки з послугами анімовано та при наведенні курсора колонка виділяється кольором, на рисунку 2.9 зображено третій блок інтерфейсу головної сторінки.

						ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			43

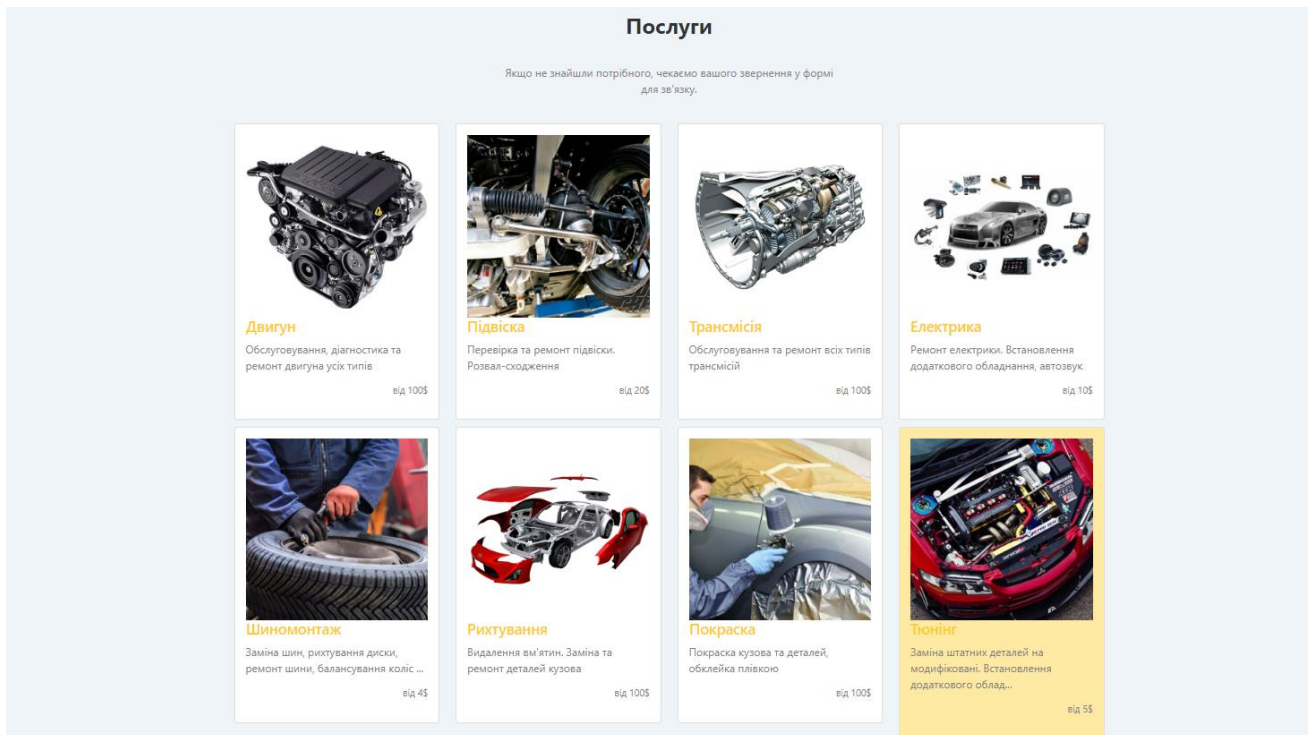


Рисунок 2.9 – Третій блок головної сторінки

Четвертий блок головної сторінки включає у себе контактну інформацію із активними кнопками для створення нового замовлення та входу в особистий кабінет. Також використана карта яка використовує Google Maps API, блок зображено на рисунку 2.10.

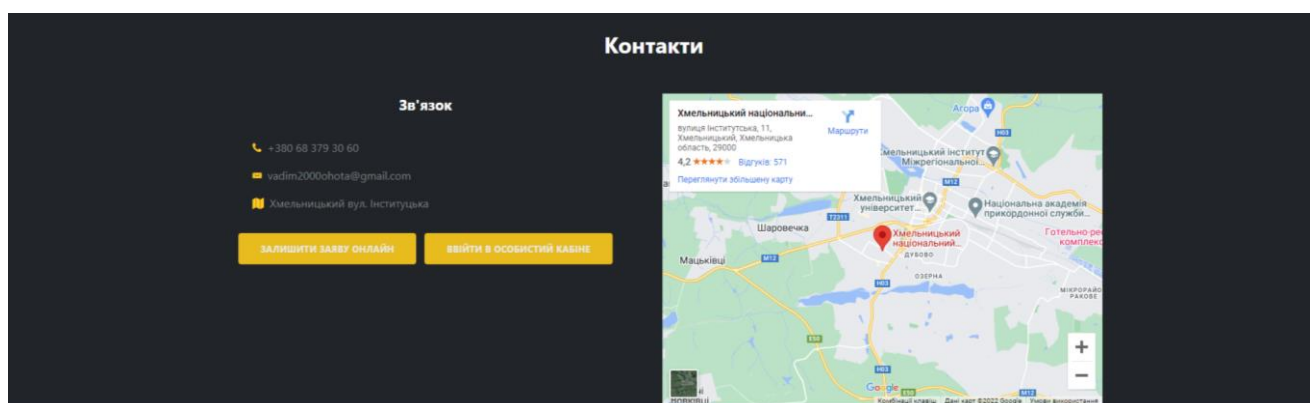


Рисунок 2.10 – Четвертий блок головної сторінки

На рисунку 2.11 зображено п'ятий блок головної сторінки який містить інформацію про керівників та керівників підрозділів підприємства.

									Арк.
									44
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ. 180146.01.08.ВД				

## Керівники

Для нас робота - це хобі



**Vadym Okhota**  
ДИРЕКТОР  
VADIM2000DHOTA@GMAIL.COM

Рисунок 2.11 – П'ятий блок головної сторінки

На рисунку 2.12 зображено шостий блок головної сторінки який містить інформацію про партнерів підприємства, та має анімацію при наведенні курсором на картинку партнера.

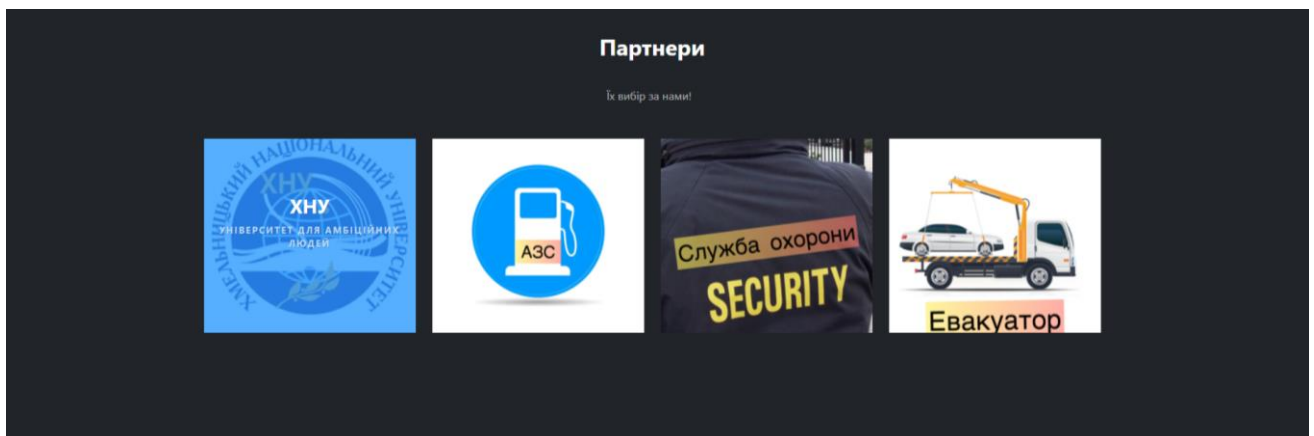


Рисунок 2.12 – Шостий блок головної сторінки

На рисунку 2.13 зображено сьомий блок сайту який містить відгуки клієнтів пр підприємство. Також на рисунку зображено футер сайту який містить посилання на сторінки сайту, лозунг сервісу та посилання на соціальні мережі через графічні іконки.

					ДППЗ. 180146.01.08.ВД	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

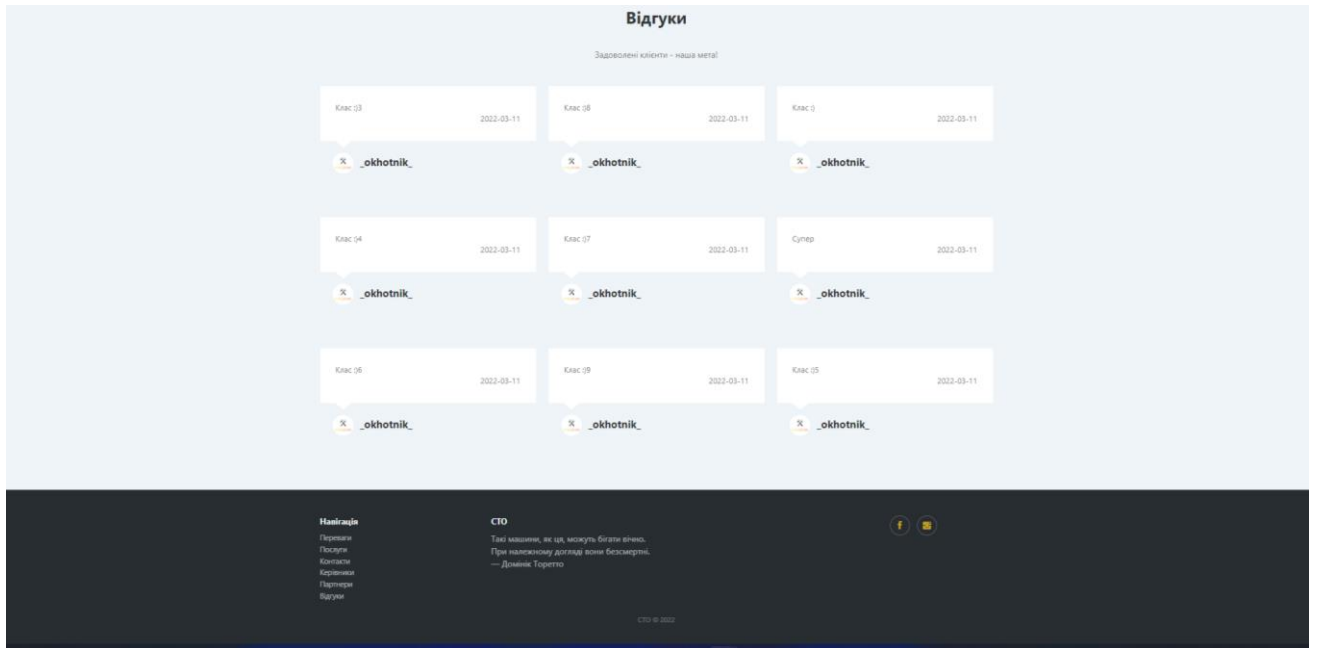


Рисунок 2.13 – Сьомий блок головної сторінки та футер

Для використання застосунку Keycloak створено сторінки із формами для створення нового профіля користувача та входу у наявний. У формах присутня анімація при наведенні курсора та відображаються підказки при помилках введення та невірній валідації, на рисунку 2.14 зображено сторінки авторизації та аутентифікації.

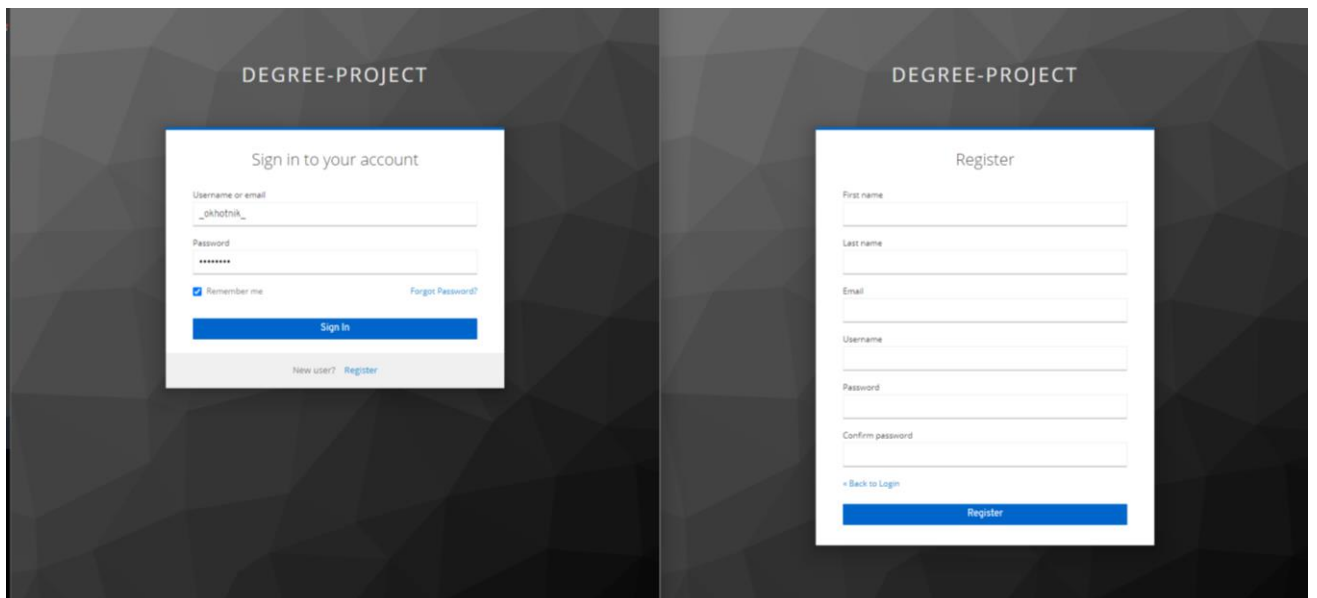


Рисунок 2.14 – Сьомий блок головної сторінки та футер

На рисунку 2.15 показано особисту сторінку користувача. Особиста сторінка має різні графічні елементи в залежності від статусу користувача елементи .

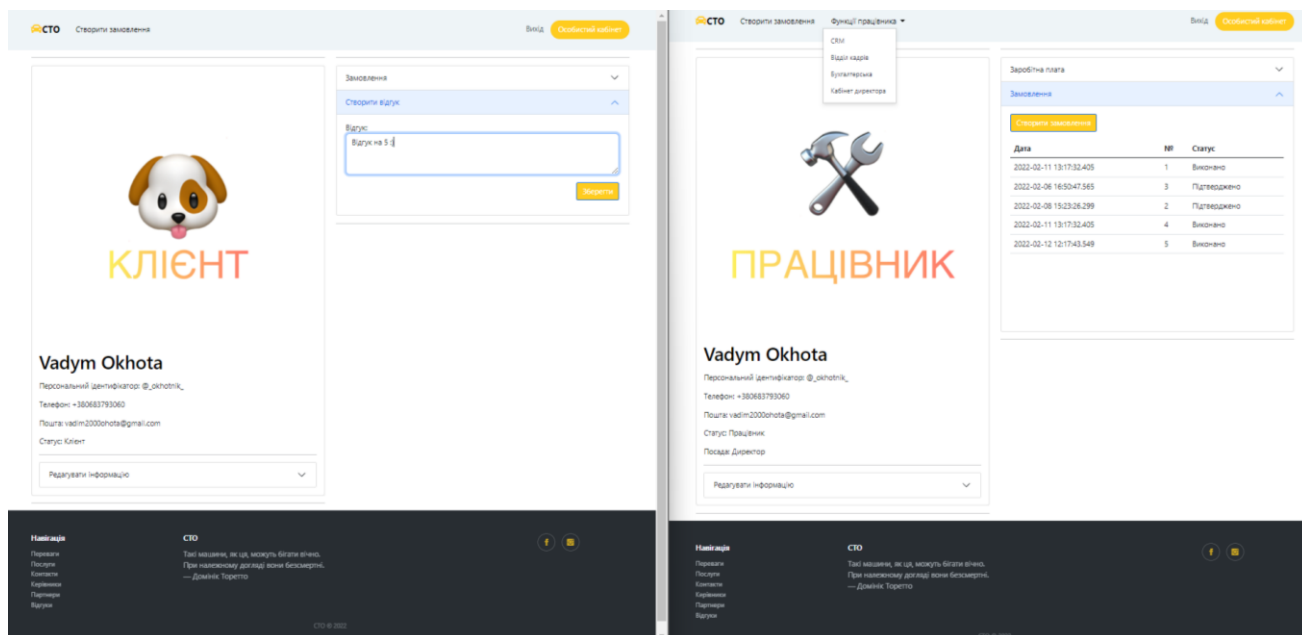


Рисунок 2.15 – Вигляд особистої сторінки для клієнтів та працівників

Для створення нового замовлення використовується сторінка із формою, сторінка містить автозаповнюючі поля. Для редагування замовлення створена сторінка з можливостями для додавання виконавців замовлення, та формування списку витрат (див. рисунок 2.16).

Дизайн CRM системи простий та інтуїтивно зрозумілий, головна сторінка розділена на чотири колони які відповідають статусу виконання замовлення. З сторінки CRM доступне швидке оперування замовленням. CRM система містить кнопку яке переходить на сторінку яка містить суцільний список архівованих замовлень (див. рисунок 2.17). CRM система має акордеоний елемент який упускає шторку із формою для пошуку замовлення по унікальному номеріві замовлення. На рисунку 2.18 зображено графічний інтерфейс CRM системи. Дизайн системи дозволяє одночасно упускати всі шторки акордеонів кожного окремого замовлення.

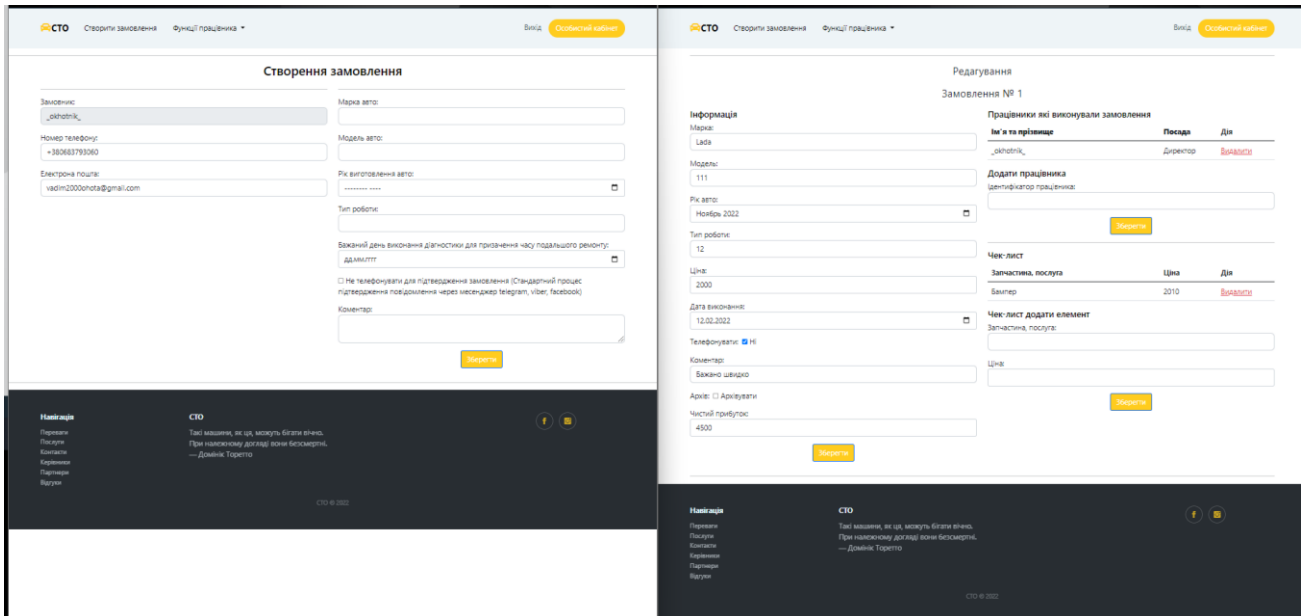


Рисунок 2.16 – Дизайн сторінок для створення та редагування замовлення

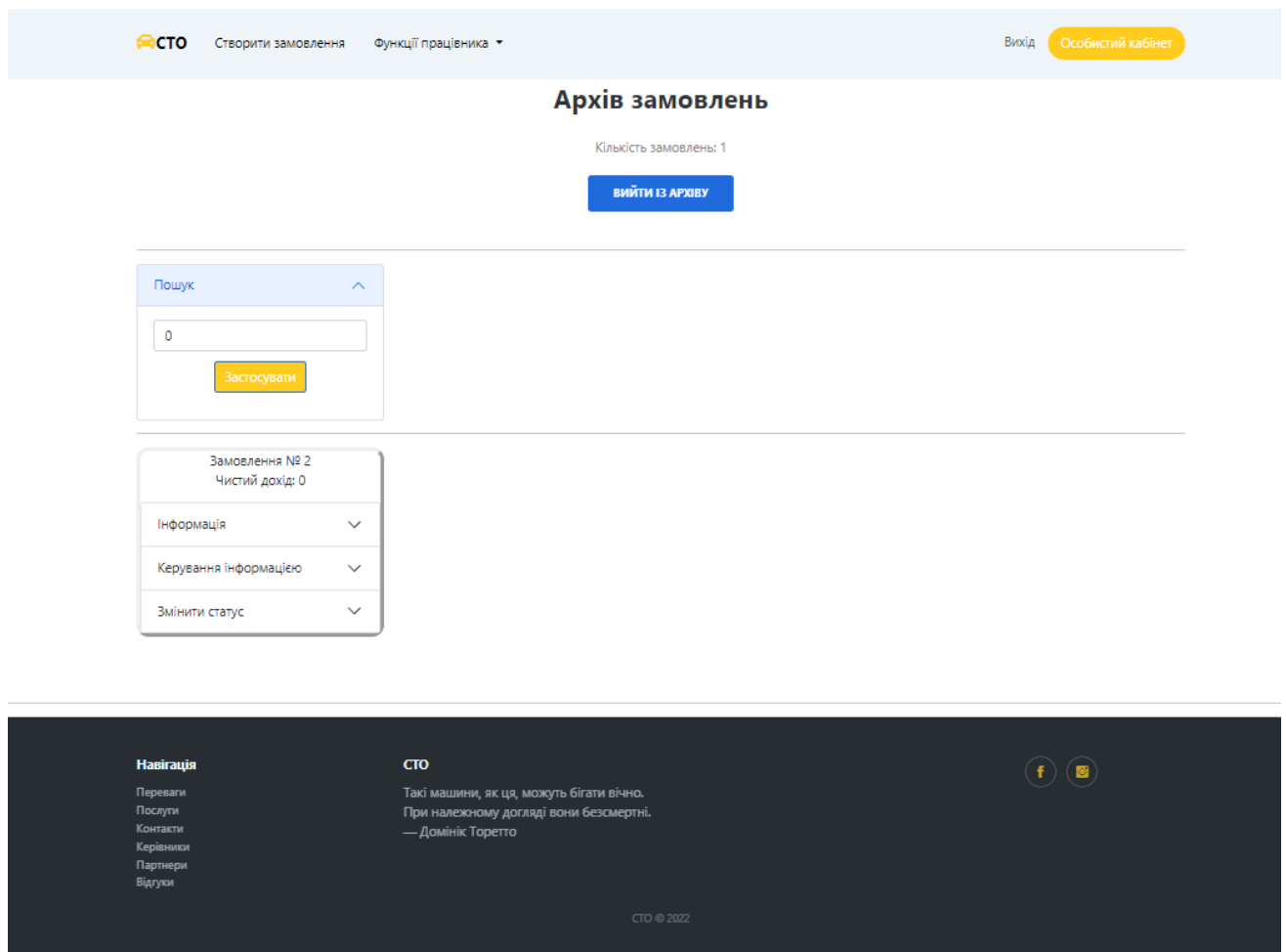


Рисунок 2.17 – Дизайн архіву замовлень

### Керування замовленнями

Кількість усіх замовлень: 5

**ПЕРЕЙТИ В АРХІВ**

Пошук ^

**Застосувати**

Опрацювання	Підтвержені	Виконання	Виконано
Кількість замовлень: 1	Кількість замовлень: 1	Кількість замовлень: 0	Кількість замовлень: 3
<p>Замовлення № 6</p> <p>Інформація <span>^</span></p> <p>Індифікатор клієнта: test</p> <p>Марка та модель авто:</p> <p>Тип роботи:</p> <p>Дата створення заявки: 2022-03-24 16:54:49.071</p> <p>Дата бажаного початку роботи:</p> <p>Керування інформацією <span>^</span></p> <p><a href="#">Детальніше</a></p> <p><a href="#">Редагувати</a></p> <p><a href="#">Архівувати</a></p> <p><a href="#">Видалити</a></p> <p>Змінити статус <span>^</span></p> <p><b>Підтверджено</b></p> <p><b>Виконується</b></p> <p><b>Виконано</b></p>	<p>Замовлення № 3</p> <p>Інформація <span>^</span></p> <p>Керування інформацією <span>^</span></p> <p><a href="#">Детальніше</a></p> <p><a href="#">Редагувати</a></p> <p><a href="#">Архівувати</a></p> <p><a href="#">Видалити</a></p> <p>Змінити статус <span>^</span></p>		<p>Замовлення № 1</p> <p>Чистий дохід: 4500</p> <p>Інформація <span>^</span></p> <p>Керування інформацією <span>^</span></p> <p>Змінити статус <span>^</span></p> <p><b>Опрацювання</b></p> <p><b>Підтверджено</b></p> <p><b>Виконується</b></p> <hr/> <p>Замовлення № 4</p> <p>Чистий дохід: 4500</p> <p>Інформація <span>^</span></p> <p>Керування інформацією <span>^</span></p> <p>Змінити статус <span>^</span></p> <hr/> <p>Замовлення № 5</p> <p>Чистий дохід: 1000000</p> <p>Інформація <span>^</span></p> <p>Керування інформацією <span>^</span></p> <p>Змінити статус <span>^</span></p>

**Навігація**

- Перевали
- Послуги
- Контакти
- Коріння
- Партнери
- Відгуки

**СТО**

Такі машини, як ця, можуть бігати вічно.  
При належному догляді вони безсмертні.  
— Домінік Торретто

СТО © 2022

Рисунок 2.18 – Дизайн CRM системи

Для управління працівниками створено одну сторінку, яка містить у собі акордеон який опускає шторку для пошуку працівника у списку, або для додавання нового працівника, також дана форма використовується для зміни

посади працівника. Нище розташований список користувачів із можливістю перегляду інформації та керування (див. рисунок 2.19).

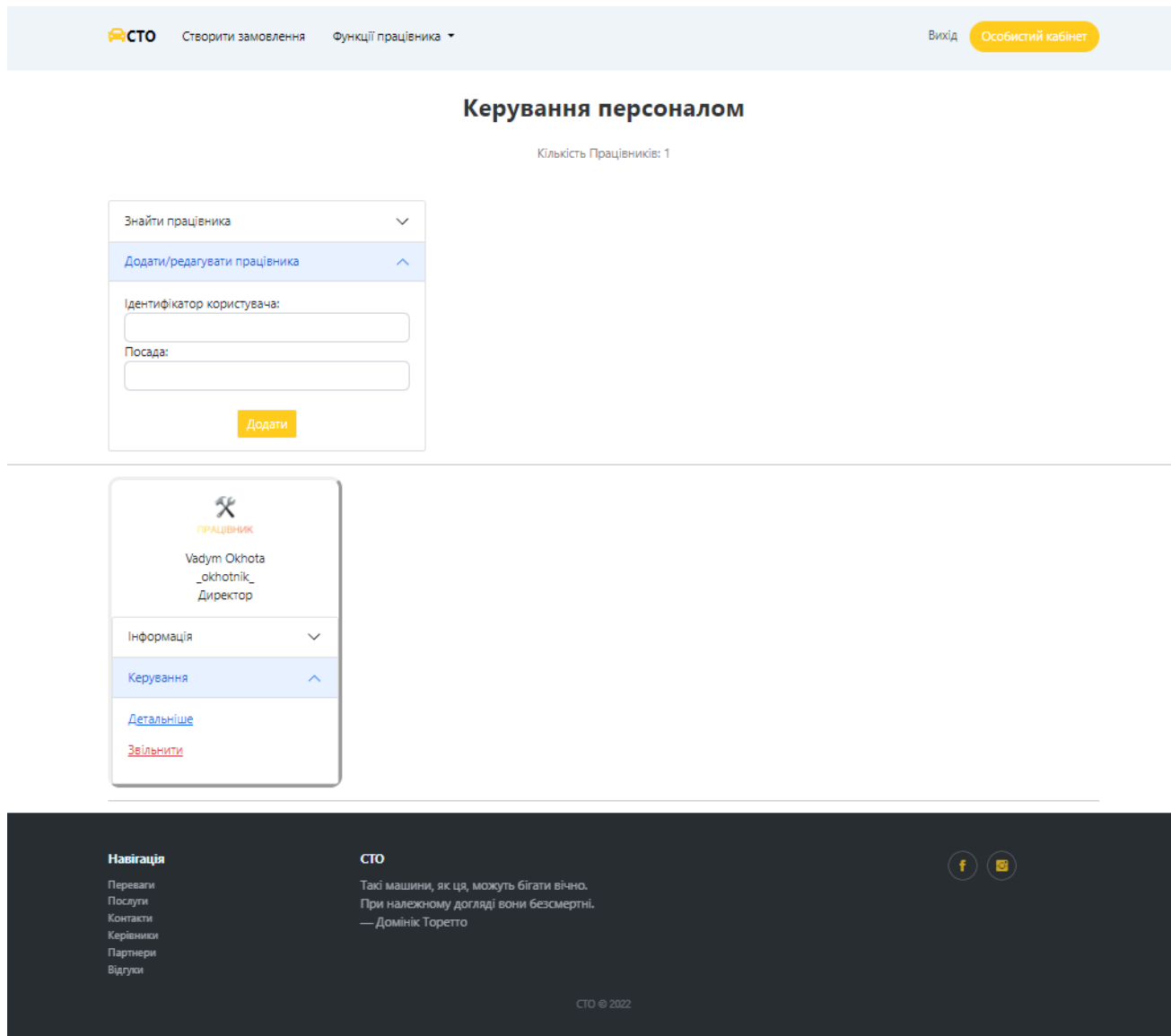


Рисунок 2.19 – Дизайн сторінки керування працівниками

Для створення нових витрат створено сторінку для бухгалтерського обліку (див. рисунок 2.20). Сторінка містить форми для створення виплат різних призначень. Також на сторінці розміщена форма для вибору місяця відображення статистики. На сторінці розміщено елементи які відображають фінансову статистику підприємства. Також на сторінці розміщено два списки для відображення списку транзакцій які приносять прибуток та окремий список для відображення витрат.

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

## Бухгалтерська

Статистика:  
2022-02

Змінити дату для статистики

Виберіть місяць:

Февраль 2022

Застосувати

<b>ЧИСТИЙ ПРИБУТОК</b> 987800	<b>ПРИБУТОК</b> 1009000	<b>ВИТРАТИ</b> 21200	<b>ВИКОНАНО ЗАМОВЛЕНЬ</b> 3
----------------------------------	----------------------------	-------------------------	--------------------------------

Подати звіт за зарплату

Зарплата Премія Аванс

Ідентифікатор працівника:

Кількість годин праці:

Сума:

Дата:

Подати

Подати звіт за витрату

Призначення:

Короткі деталі:

Повна інформація:

Сума:

Дата:

Подати

**Обіг коштів прибуток**

Призначення	Деталі	Сума	Дата
Надання послуг	Ремонт/діагностика №1	+4500	2022-02-12
Надання послуг	Ремонт/діагностика №4	+4500	2022-02-12
Надання послуг	Ремонт/діагностика №5	+1000000	2022-02-12

**Обіг коштів витрати**



Призначення	Деталі	Сума	Дата
Виплата	Виплата зарплати	-15000	2022-02-13T22:54
Виплата	Виплата зарплати	-1200	2022-02-13T02:58
Виплата	Виплата премії	-5000	2022-02-13T23:01

**Навігація**

- Переваги
- Послуги
- Контакти
- Кар'єра
- Партнери
- Відгуки

**СТО**

Такі машини, як ця, можуть бігати вічно.  
При належному догляді вони безсмертні.  
— Домінік Торетто

СТО © 2022

Рисунок 2.20 – Дизайн сторінки бухгалтерського кабінету

Для відображення транзакцій створено окрему сторінку яка містить інформацію про рух коштів, та якщо працівник має відповідний доступ відкривається доступ для редагування інформації та її видалення (див. рисунок 2.21).

СТО Створити замовлення Функції працівника Вихід Особистий кабінет

Транзакція Id: 3

Призначення: Виплата  
 Ідентифікатор користувача: okhotnik  
 Пропрацьовано годин: 120  
 Короткі деталі: Виплата зарплати  
 Повна інформація: Виплата зарплати okhotnik  
 Сума: 15000

Видано: Директор okhotnik  
 Дата: 2022-02-13T22:54

Редагувати

Призначення:

Ідентифікатор користувача:

Пропрацьовано годин:

Короткі деталі:

Повна інформація:

Сума:

Видано (ідентифікатор працівника):

Дата:

Зберегти

Видалити

**Навігація**  
 Перевали  
 Послуги  
 Контакти  
 Керівники  
 Партнери  
 Відеореєстри

**СТО**  
 Такі машини, як ця, можуть бігати вічно.  
 При належному догляді вони безсмертні.  
 — Дрейвік Торетто

СТО © 2022

Рисунок 2.21 – Дизайн сторінки транзакції

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

Та остання сторінка інтернет-платформи це особиста сторінка директора. Сторінка містить кнопки для переходу на сторінки CRM, керування персоналом та бухгалтерського обліку. На даній сторінці розміщено також список логу взаємодії користувачів із інтернет платформою. Сторінка містить також елемент із формуою для виконання пошуку взаємодій користувачів із сервісом за визначений день (див. рисунок 2.22).

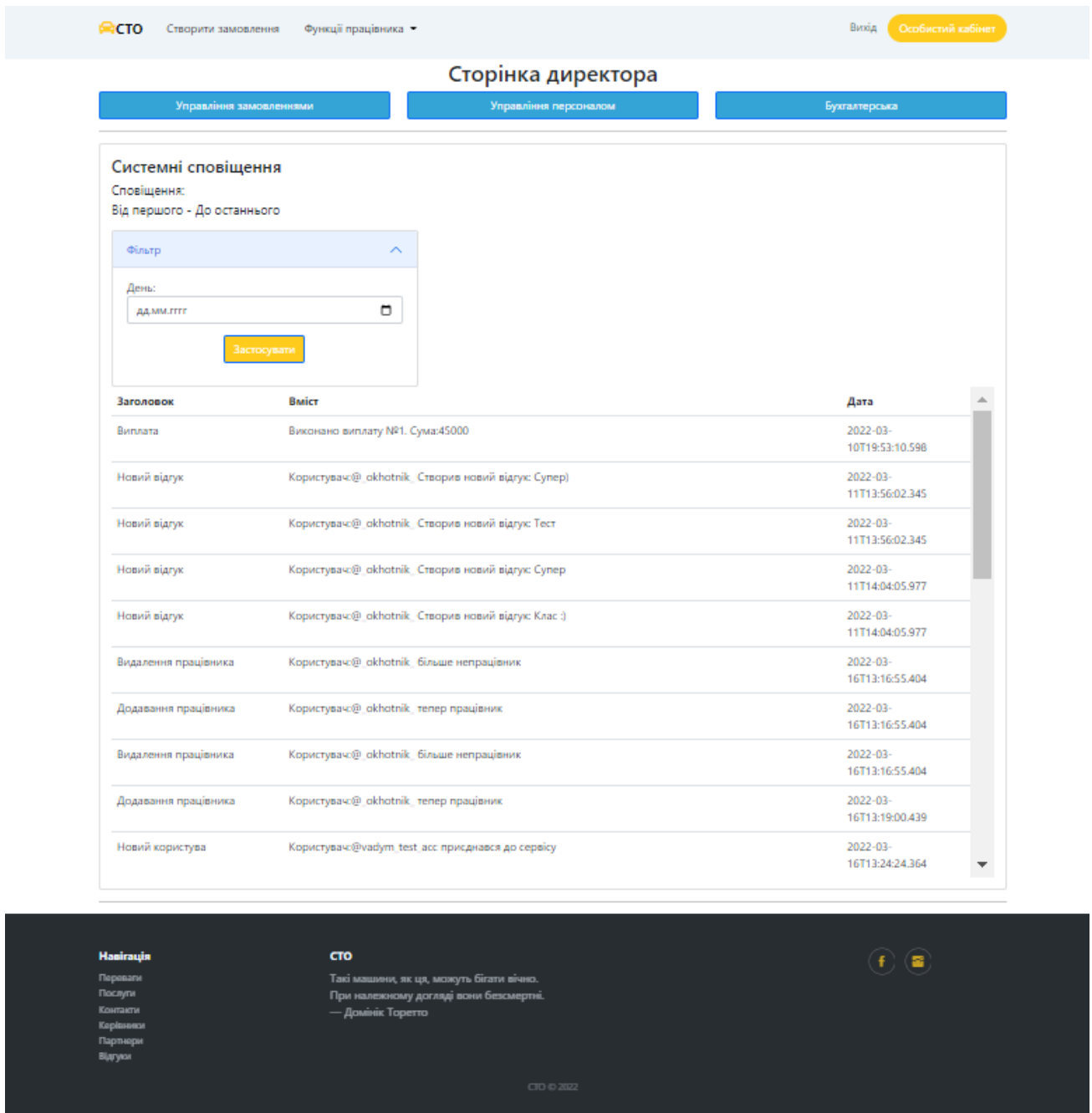


Рисунок 2.21 – Дизайн сторінки директора

## 2.7 Аналіз та вибір технології та методів для реалізації застосунку

Для реалізації програмного продукту як раніше зазначалося буде використана мова програмування Java та найпопулярніший фреймворк для даної мови Spring. Фреймворк кросплатформений та дозволяє створювати застосунки під операційні системи:

- Windows;
- Linux;
- Android.

Фреймворк Spring має у собі вбудовані можливості для взаємодії із різними базами даних, дана складова фреймворка має назву Spring Date. Також фреймворк має вбудовані можливості для впровадження системи безпеки та керування рівнями доступу Spring Security. Фреймворк ідеально працює у програмах контейнеризації програмного забезпечення, а саме Docker.

Для впровадження системи аутентифікації та авторизації використається програмний продукт Keycloak, інформація про даний програмний продукт була подана у роботі. Сервіс буде відтворюватися в мікросервійсній архітектурі в контейнері Docker та використовувати спільну мережу для взаємодії між собою.

Для збереження даних про користувачів включно із зашифрованими паролями буде використано реляційну БД MySQL.

Для збереження даних для опрацювання логіки програмного продукту та іншої інформації яка була наведене в аналізі бази даних, буде використано базу даних MongoDB. Взаємодія мікросервісу розробленого на фреймворку Spring відповідає спеціальний адаптер.

Для контейнеризації програмного продукту буде використано спеціальне програмне забезпечення для контейнеризації програмного забезпечення Docker. Дане програмне забезпечення створює Image програмного забезпечення та всі елементи які потрібні для запуску ПЗ. Docker контейнер – це ізольоване середовище для виконання програмного продукту. Для взаємодії мікросервісів між собою та із базами даних буде створено спеціальну мережу у Docker. Для

										Арк.
										54
Змн.	Арк.	№ докум.	Підпис	Дата						

контейнеризації мікросервісів та баз даних із створенням спеціальної мережі буде використано Docker Compose. Що дозволить виконати дані дії за допомогою одного файлу із налаштуваннями формату YAML за допомогою однієї команди.

У результаті детального аналізу всіх складових системи було визначено та описано основні особливості програмного продукту. Також було описано основні модулі програмного забезпечення.

Проведено проєктування та опис елементів користувацького інтерфейсу, також було продемонстровано дизайн сторінок інтернет-платформи.

## 2.7 Розробка Алгоритмів роботи застосунку

Після вибору архітектури та проєктування інтерфейсу, створимо алгоритм роботи застосунку. визначимо найбільш використовуванні процеси:

- Відображення сторінки (див. рисунок 2.22);
- пошук/фільтрування інформації (див. рисунок 2.23);
- створення, редагування, видалення інформації (див. рисунок 2.24);

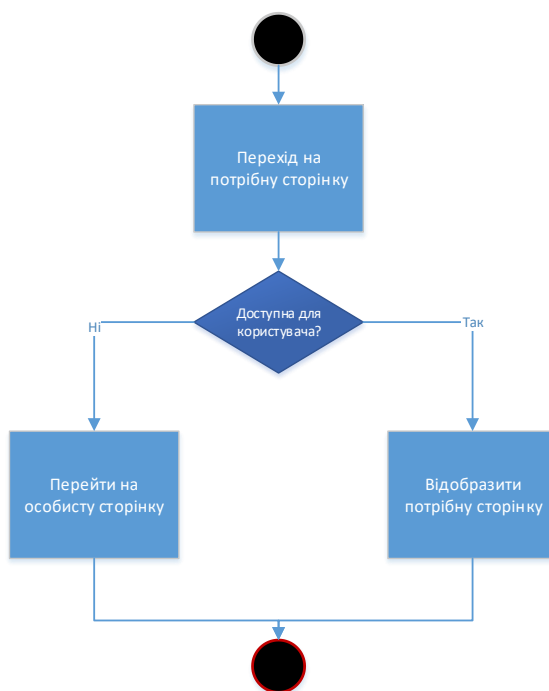


Рисунок 2.22 – Діаграма переходів станів відображення сторінки



Рисунок 2.23 – Діаграма переходів станів пошуку/фільтрування інформації

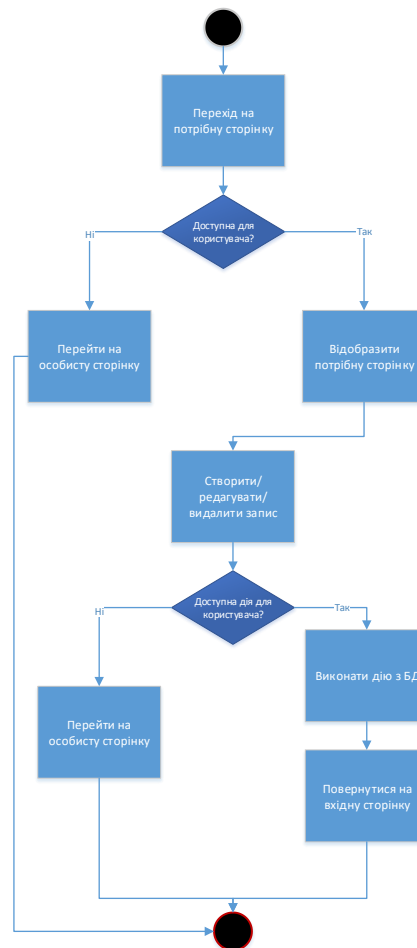


Рисунок 2.24 – Діаграма переходів станів додавання, редагування, вилучення інформації

Змн.	Арк.	№ докум.	Підпис	Дата

Мікросервіси взаємодіють між собою і окремо з сервером. Для кожного мікросервісу використаний сервер БД. На рисунку 2.25 зображено діаграму потоків даних. Діаграми потоків даних додатку показує переміщення даних між різними компонентами програмного забезпечення.

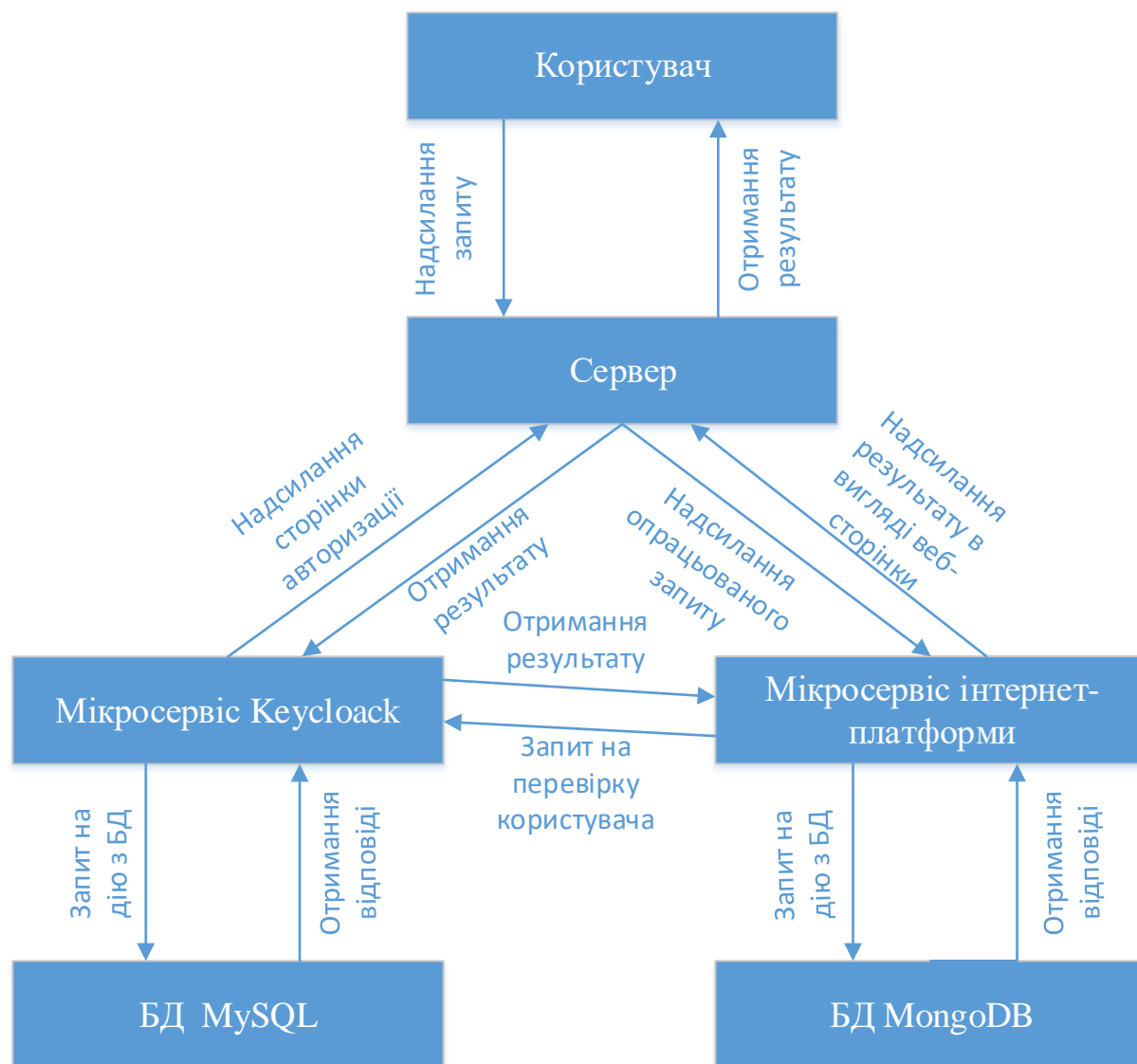


Рисунок 2.25 – Діаграма потоів даних

Функціональна діаграма зображує залежності між різними процесами програмного забезпечення. На діаграмі (див. рисунок 2.26) представлено основний процес всього додатку та необхідні для його роботи елементи.

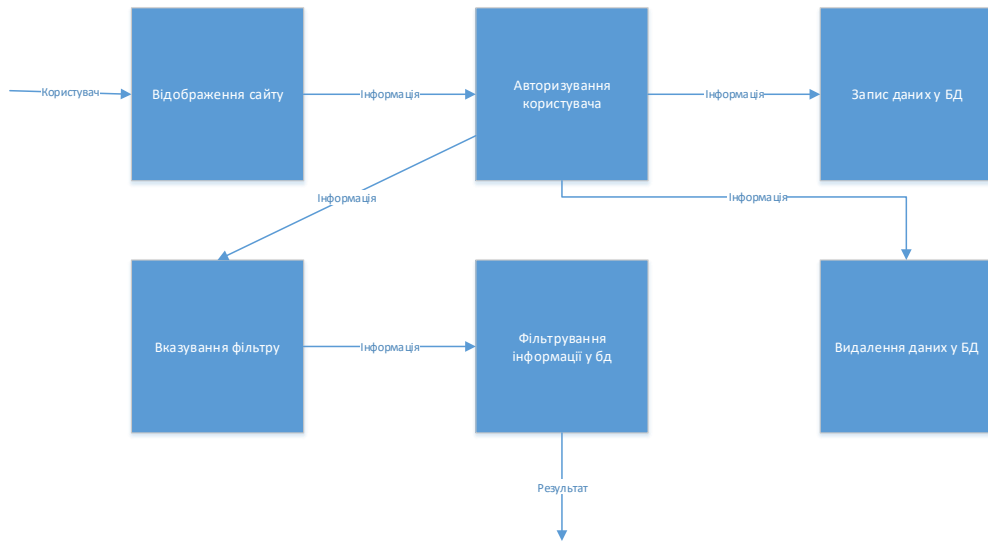


Рисунок 2.25 – Функціональна діаграма

Система безпеки рахується однією з основних у програмних продуктах. Розроблений наступний алгоритм для мікросервісу Keycloak (див рисунок 2.26).

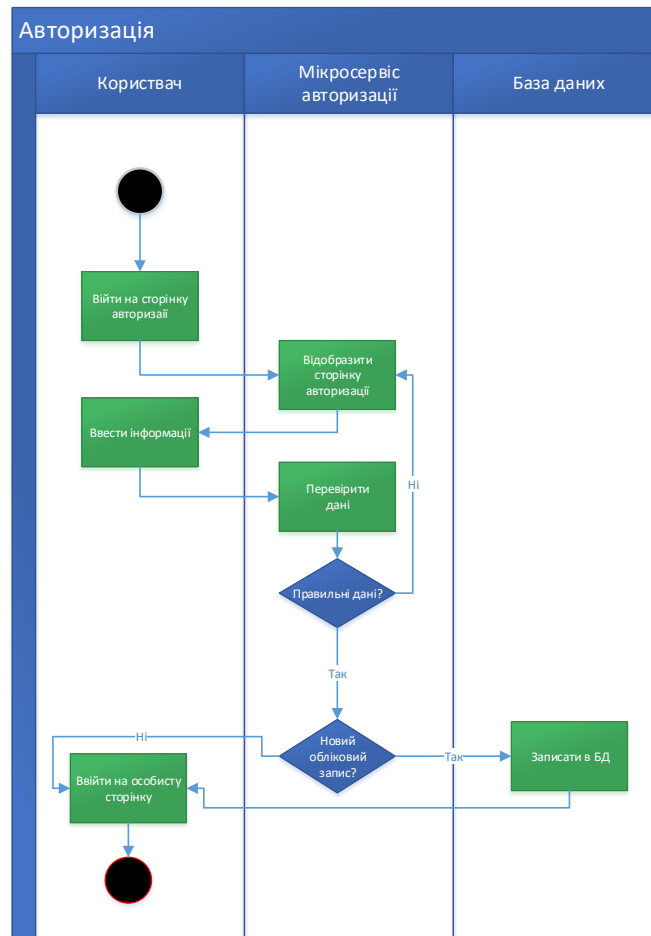


Рисунок 2.26 – діаграма діяльності алгоритму авторизації

Змн.	Арк.	№ докум.	Підпис	Дата

Для кращого розуміння зобразимо алгоритм авторизації діаграмою кооперації (див. рисунок 2.27) та діаграмою послідовності (див. рисунок 2.28).

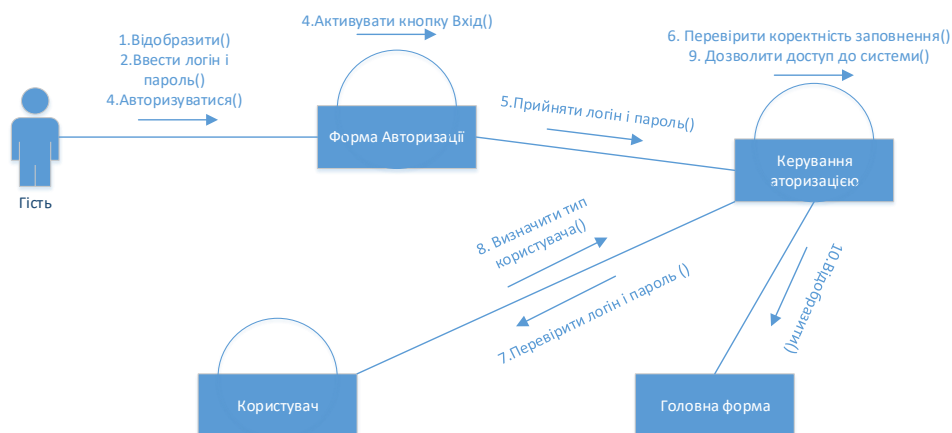


Рисунок 2.27 – діаграма кооперації алгоритму авторизації

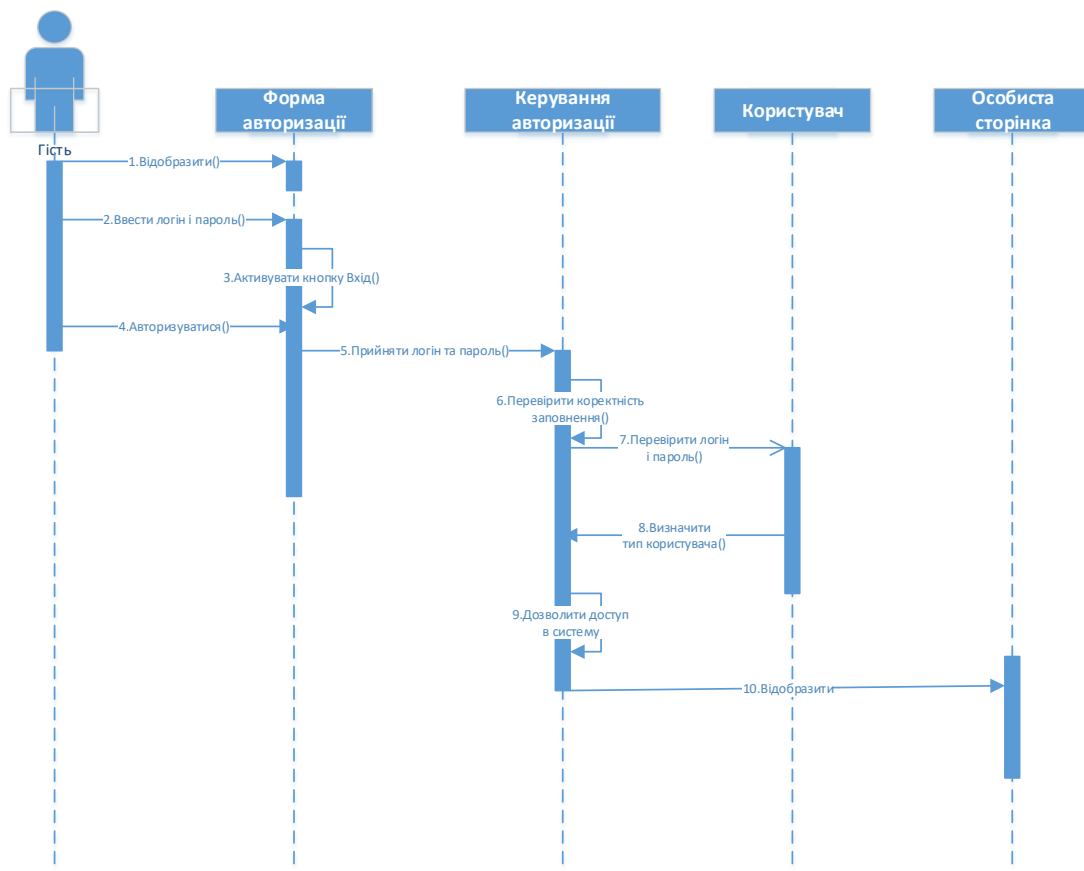


Рисунок 2.28 – діаграма послідовності алгоритму авторизації

Алгоритм взаємодії користувача із інтернет-платформою зображено на рисунку 2.29. Система при оперуванні користувача інформацією перевіряє доступ користувача для даних дій, що створює додаткову захищеність застосунку.

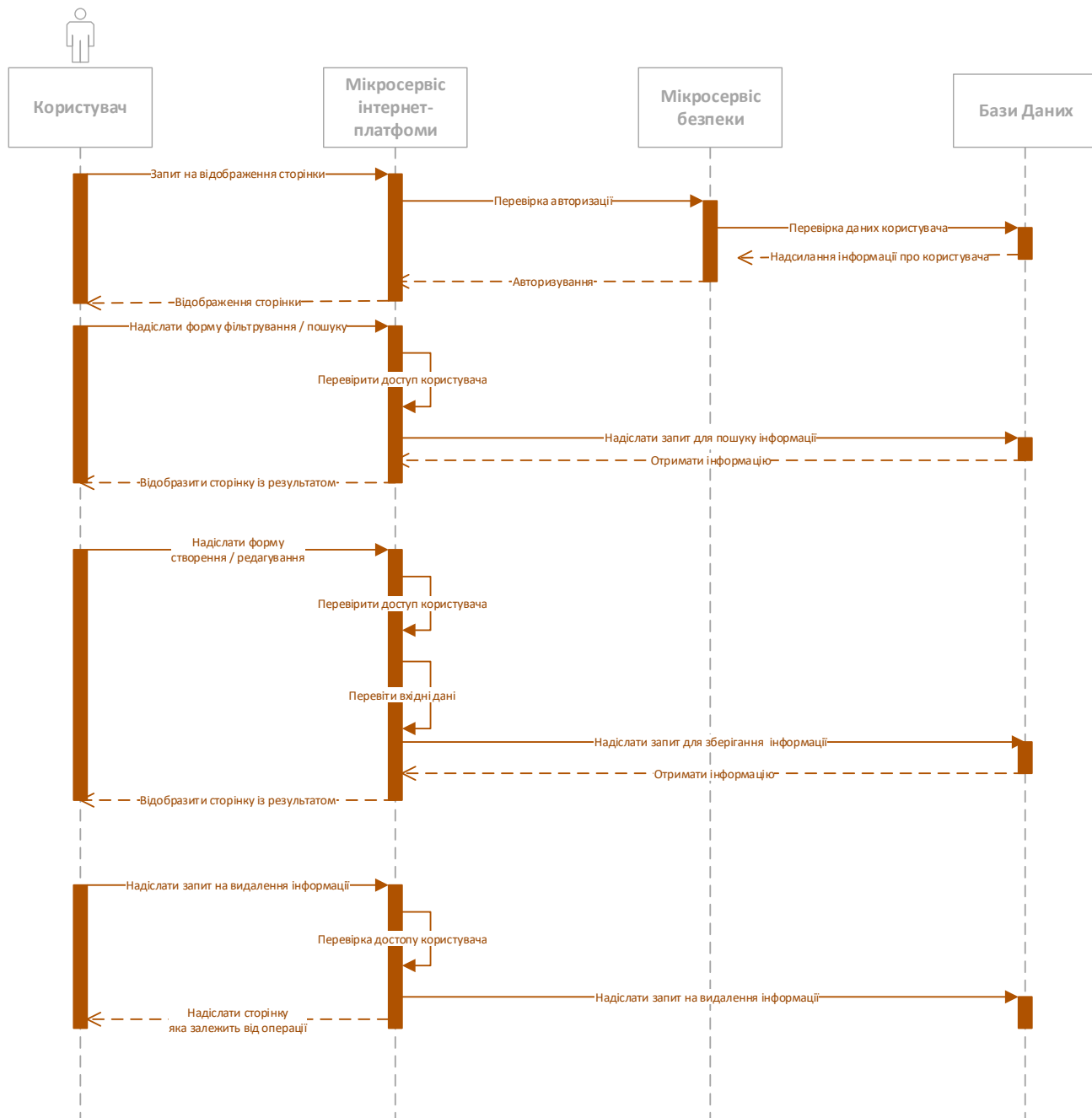


Рисунок 2.29 – діаграма послідовності алгоритму взаємодії користувача із інтернет-платформою

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Реалізація серверної частини програмного забезпечення

Створення та написання програмного коду для розроблювального програмного продукту буде виконано у середовищі розробки IntelliJ IDEA. Для використання сторонніх бібліотек використано Apache Maven – це засіб автоматизації роботи із програмними проєктами які розроблені іншими розробниками та використовують ліцензію Apache-2.0 license. Щоб побудувати стартовий шаблон Spring застосунку використовуємо сервіс Spring Initialize.

Для того щоб розроблювальний продукт міг взаємодіяти із базами даних та використовувати функції які не надає для використання Java, вказуємо залежності які ми будемо використовувати в програмному продукті, Maven завантажить та інтегрує для використання бібліотеки. Вміст файлу Maven pom.xml:

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.keycloak.bom</groupId>
      <artifactId>keycloak-adapter-bom</artifactId>
      <version>13.0.1</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-mongodb</artifactId>
    <version>2.5.6</version>
  </dependency>

  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.22</version>
    <scope>provided</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
    <version>2.6.1</version>
  </dependency>

  <dependency>
    <groupId>org.thymeleaf</groupId>
```

						ДППЗ. 180146.01.08.ВД	Арк.
							61
Змн.	Арк.	№ докум.	Підпис	Дата			

```

    <artifactId>thymeleaf</artifactId>
    <version>3.0.14.RELEASE</version>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>

<dependency>
    <groupId>org.keycloak</groupId>
    <artifactId>keycloak-spring-boot-starter</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-test</artifactId>
    <scope>test</scope>
</dependency>
</dependencies>

```

#### Опис деталей використання даних залежностей:

- keycloak-adapter-bom – використовуються для взаємодії з мікросервісом Keycloak;
- spring-boot-starter-data-mongodb – бібліотека для взаємодії із БД MongoDB
- lombok – бібліотека яка замінює стандартні методи та інші інструменти на анотації;
- spring-boot-starter-thymeleaf – бібліотека для використання движка шаблонів Thymeleaf, що дозволяє використовувати для представлення програмний код HTML, CSS, JavaScript.
- spring-boot-starter-web – бібліотека яка інтегрує програмне забезпечення для моделі MVC та інші загальні стандарти, моделі WEB застосунків.

					ДППЗ. 180146.01.08.ВД	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

– `spring-boot-starter-test` - бібліотека для тестування додатків написаних на фреймворку Spring. Містить методи як для модульного тестування так і інтеграційного;

– `spring-boot-starter-security` – бібліотека для підключення конфігурацій безпеки при запуску сервера;

– `keycloak-spring-boot-starter` – бібліотека для встановлення зв'язку мікросервісом Keycloak при запуску сервера;

– `spring-security-test` – бібліотека яка включає у себе методи тестування безпеки застосунків написаних за допомогою Spring Security.

Початок роботи фреймворка Spring розпочинається викликом методом `main`:

```
@SpringBootApplication
public class CtoApplication {
    public static void main(String[] args) {
        SpringApplication.run(CtoApplication.class, args);
    }
}
```

Анотація `@SpringBootApplication` вказує що код буде використовувати для свого запуску фреймворк Spring. При запуску фреймворк ініціалізує усі компоненти та завантажує контекст програмного забезпечення.

Для відображення сторінок інтернет-платформи контролери використовують метод:

```
@Controller
public class HomeController {
    ...
    @Autowired
    private HomeService homeService;
    ...
    @GetMapping("/")
    public String home(Model model, HttpServletRequest request) {
        ...
        return "home";
    }
    ...
}
```

Анотація @Controller вказує фреймворку що даний клас контролер. @Autowired створює ін'єкцію для можливості використання сервісу та його методів у контролері. Анотація @GetMapping вказує посилання на сторінку та метод для звернення. Аргумент model використовується для запису та передачі інформації, об'єктів у представлення. HttpServletRequest дозволяє зчитувати заголовки хедера та використовувати її у логіці програми.

У програмному забезпеченні розроблено наступні контролери (код контролерів див. додаток В):

- HomeController містить – відображення сторінки;
- AccountantPageController – відображення сторінок бухгалтерського обліку, фільтрування відображення за визначеним місяцем. Створення нових виплат та редагування наявних;
- DirectorPageController – відображення сторінки директора та фільтрування виведеної інформації за визначеним днем;
- OrderController містить – відображення заданого замовлення, також включає у себе методи для створення, редагування, видалення замовлень. Містить методи для роботи CRM системи.
- PersonalPageController – для відображення персональної сторінки користувача та метод редагування особистої інформації;
- TeamManagementController - зміна ролі користувача на працівника. Метод пошуку працівника.
- TransactionsInfoController – відображення сторінки з інформацією про відповідну транзакцію та методи для редагування та видалення.
- SecurityController – створення нового користувача, перевірки профілю користувача.

Для виконання обчислювальної логіки використовують сервіси. Анотація @Service вказує що даний клас виконує роль сервісу. Фрагмент сервісу AccountantPageService:

```
@Service  
public class AccountantPageService {
```

						ДППЗ. 180146.01.08.ВД	Арк.
							64
Змн.	Арк.	№ докум.	Підпис	Дата			

```

@Autowired
private OrdersRepository ordersRepository;
...
public String defaultDate() {
    Timestamp timestamp = new Timestamp(System.currentTimeMillis());
    String month = null;
    if (timestamp.toLocalDateTime().getMonthValue() < 10) {
        month = "0" + timestamp.toLocalDateTime().getMonthValue();
    } else {
        month = String.valueOf(timestamp.toLocalDateTime().getMonthValue());
    }
    return timestamp.toLocalDateTime().getYear() + "-" + month;
}
...
}

```

У програмному забезпеченні розроблені наступні сервіси (код сервісів див. додаток В), сервіси та методи подано в таблиці 3.1.

Таблиця 3.1 – сервіси та методи

Сервіс	Методи
AccountantPageService	<ul style="list-style-type: none"> <li>– defaultDate – виконує зчитування дати із комп'ютера та змінює під потрібний формат для подальшого використання у логіці програми;</li> <li>– ordersDTOListFindAndTransactionsMoney – виконує пошук у базі даних інформацію про прибуток та витрати, для пошуку використовує дату;</li> <li>– transactionsInfoDTOS – виконує пошук транзакцій за датою;</li> <li>– ordersDTOList – шукає замовлення за датою;</li> <li>– addCoast – записує у БД нову виплату;</li> <li>– transactionsNumberCheck – перевіряє на доступність унікальний номер транзакції, унікальний номер.</li> </ul>

Продовження таблиці 3.2

HomeService	<ul style="list-style-type: none"> <li>– managementList – створює список із посад;</li> <li>– reviewsGetList – зчитує відгуки з БД;</li> <li>– reviewsGetLogicId – перевіряє доступність унікального номера відгуку, повертає доступний унікальний номер.</li> </ul>
OrderService	<ul style="list-style-type: none"> <li>– orderCreateSave – записує та редагує замовлення в БД;</li> <li>– orderCreateNumber – перевіряє доступність унікального номера замовлення, повертає доступний унікальний номер;</li> <li>– orderAddTeam – додає нового виконавця замовлення у БД;</li> <li>– orderDellTeam – видаляє виконавця замовлення із БД;</li> <li>– orderAddCheck – записує в БД витрату про витрату при виконанні замовлення;</li> <li>– orderPrice – підсумовує ціну послуги сумуючи витрати при виконанні;</li> <li>– orderCreateNumberCheck - перевіряє доступність унікального номера витрати, повертає доступний унікальний номер;</li> <li>– orderDellCheck – видалення витрати з БД.</li> </ul>
PersonalPageService	<ul style="list-style-type: none"> <li>– createReviews – записує відгук у БД;</li> <li>– createLogicId - перевіряє доступність унікального номера, повертає доступний унікальний номер;</li> <li>– editAccess – перевіряє доступ користувача для зміни інформації на особистих сторінках.</li> </ul>

### Кінець таблиці 3.1

SecurityService	<ul style="list-style-type: none"> <li>– profileCheck – перевіряє наявність облікового запису користувача;</li> <li>– createProfile – створює особистий профіль користувача;</li> <li>– customAccess – переспрямовує користувача на особисту сторінку.</li> </ul>
LogService	<ul style="list-style-type: none"> <li>– addLog – створює та записує в БД лог;</li> <li>– findFilterList – знаходить в БД логуювання за днем, якщо логуювання відсутні повертає за увесь час;</li> <li>– findAllLogs – знаходить в БД усі логуювання.</li> <li>– createLogicDate – зчитує дату та форматує її для використані у логіці;</li> <li>– findByLogicDateDays – знаходить в БД логуювання за днем.</li> </ul>

Для виконання логіки ПЗ використовують об'єкти (DTO) для запису та зчитування інформації. Для запису логуювання в БД використовується клас LogDTO:

```

@Getter
@Setter
@Document(collection = "Logs")
public class LogDTO {

    @Id
    public ObjectId objectId;
    public String type;
    public String name;
    public String title;
    public String message;
    public String date;
    public String logicDateDays;
}
    
```

Анотації @Getter та @Setter викликають Lombok, це дозволяє зменшити написання програмного коду. Анотація @Document використовується для

взаємодії з БД MongoDB та вказує назву документу для дій із БД. Сервіс має наступні DTO (код див. додаток В) (вміст полів об'єктів див. розділ 2.4):

- LogDTO;
- FindLogDTO;
- OrdersCheckDTO;
- OrdersDTO
- OrderStatusDTO;
- OrdersWorkTeamDTO;
- PartnersDTO;
- ReviewsDTO;
- ServicesListDTO;
- TransactionsInfoDTO;
- UsersDTO.

Для взаємодії з БД використовуються репозиторії, репозиторій OrdersCheckRepository:

```
@Repository
public interface OrdersCheckRepository extends MongoRepository<OrdersCheckDTO,
ObjectId> {
    List<OrdersCheckDTO> findAllByNumberOrder(long number);
    OrdersCheckDTO findByNumberCheck(long number);
    OrdersCheckDTO findByNumberOrder(long number);
    OrdersCheckDTO findByNumberOrderAndNumberCheck(long numberOrder, long
numberCheck);
}
```

Репозиторій створюється в інтерфейсі який наслідує модель для зв'язку із БД, та включає анотацію @Repository яка дозволяє фреймворку визначити інтерфейс репозиторієм. Розроблено наступні репозиторії (код див. додаток В):

- OrdersCheckRepository;
- OrdersRepository;
- OrderStatusRepository;
- OrdersWorkTeamRepository;
- PartnersRepository;
- ReviewsRepository;

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

- ServicesListRepository;
- TransactionsInfoRepository;
- UsersRepository.

Для взаємодії із мікросервісом Keycloak використовується конфігуратор **BeanKeycloakConfiguration**:

```
@Configuration
public class BeanKeycloakConfiguration {
    @Bean
    public KeycloakSpringBootConfigResolver keycloakConfigResolver() {
        return new KeycloakSpringBootConfigResolver();
    }
}
```

Анотація **@Configuration** налаштовує бібліотеку при старті сервера для взаємодії з Keycloak.

Для забезпечення та впровадження безпеки та рівнів доступу до окремих сторінок інтернет-ресурсу використовується клас налаштування Spring Security з назвою **SecurityConfig**:

```
@Configuration
@EnableWebSecurity
@ComponentScan(basePackageClasses = KeycloakSecurityComponents.class)
public class SecurityConfig extends KeycloakWebSecurityConfigurerAdapter {

    @Autowired
    public void configureGlobal(
        AuthenticationManagerBuilder auth) throws Exception {

        KeycloakAuthenticationProvider keycloakAuthenticationProvider
            = keycloakAuthenticationProvider();
        keycloakAuthenticationProvider.setGrantedAuthoritiesMapper(
            new SimpleAuthorityMapper());
        auth.authenticationProvider(keycloakAuthenticationProvider);
    }

    @Bean
    @Override
    protected SessionAuthenticationStrategy sessionAuthenticationStrategy() {
        return new RegisterSessionAuthenticationStrategy(
            new SessionRegistryImpl());
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        super.configure(http);
        http
            .authorizeRequests()
            .antMatchers("/resources/**", "/static/**",
```

						ДППЗ. 180146.01.08.ВД	Арк.
							69
Змн.	Арк.	№ докум.	Підпис	Дата			

```

"/assets/**").permitAll()
    .antMatchers("/").permitAll()
    .anyRequest().authenticated()
    .and()
    .logout()
    .permitAll();
}
}

```

Анотація `@EnableWebSecurity` впроваджує для використання додаткові методи забезпечення безпеки для Web застосунків.

Код (лістинг) програми подано у додатку В.

### 3.2 Реалізація сторінок інтернет-платформи

Для створення сторінок використано набір інструментів Bootstrap. Для відображення сторінок та з'єднання із серверною частиною програмного забезпечення використано шаблонізатор Thymeleaf. Список сторінок інтернет платформи (код сторінок подано у додатку В):

- accountant-page;
- director-page;
- home;
- order;
- order-archive;
- order-create;
- order-edit;
- order-management;
- personal-page;
- team-management;
- transactions-info;

Опис сторінок та графічне представлення (див. розділ 2.5).

						ДППЗ. 180146.01.08.ВД	Арк.
							70
Змн.	Арк.	№ докум.	Підпис	Дата			

Для підключення шаблонізатора використовується спеціальна залежність Maven вказана у розділі 3.1, та потребує вказання на залежність у заголовку документа:

```
<!DOCTYPE html>
<html
    lang="uk" xmlns:th="http://www.thymeleaf.org">
</html>
```

Шаблонізатор працює разом із мікросервісом інтернет-платформи, тобто використовуються в спільному мікросервісу.

### 3.3 Керівництво користувача

Дана інтернет-платформа використовується для відображення сторінки з інформацією про підприємство з надання послуг СТО. Та надає послуги для створення профілів клієнтів та користувачів. У інтернет-платформі розроблена власна система CRM, сторінка для бухгалтерського обліку, керування персоналом та відстеження взаємодії із програмним забезпеченням.

Особиста сторінка користувача (див. рисунок 3.1). Користувачеві доступні функції для змінити особистої інформації. На даній сторінці доступний список створений замовлень із можливістю переходу на сторінки з детальною інформацією про виконання. Для клієнтів підприємства доступна можливість створення відгуку який буде відображатися на головній сторінці інтернет-платформи. Для користувачів застосунку із статусом Працівник недоступна функція створення відгуків, але з'являється список із нарахуванням заробітної плати з можливості переходу на сторінки з деталями операції.

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

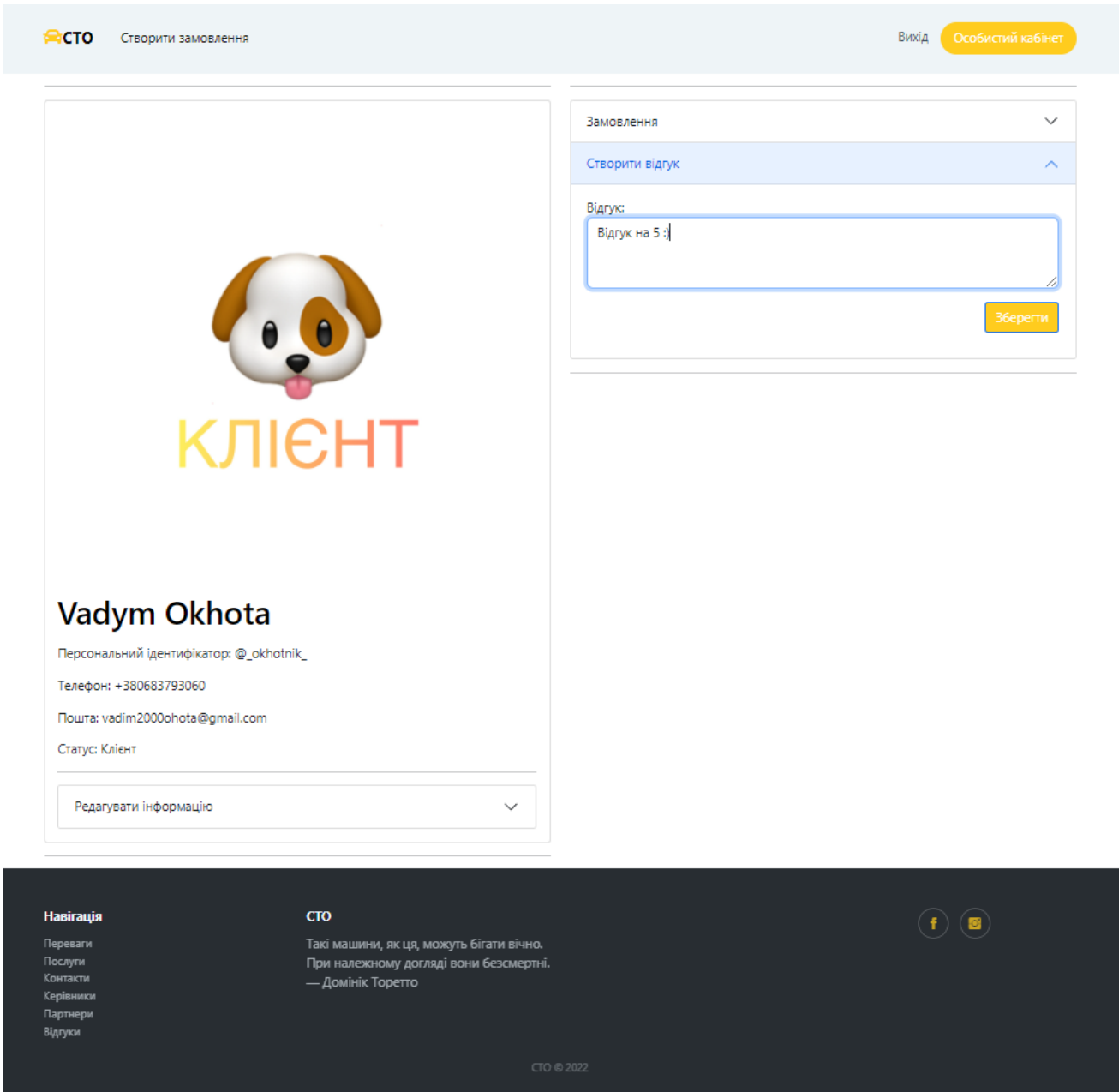


Рисунок 3.1 – Особиста сторінка користувача

Для створення замовлення замовник заповнює форму на спеціальній сторінці (див. рисунок 3.2). Поле замовник автозаповнюється персональним ідентифікатор користувача та не дозволяє вносити зміни у себе. Поля з номером телефону та електронною поштою теж автоматично заповнюються із даних профілю користувача, проте дозволяють вносити у себе зміни. Інші поля є необов'язковими та можуть бути вказані працівниками підприємства. Для створення замовлення після заповнення інформацією натискається кнопка «Зберегти».

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

СТО
Створити замовлення
Функції працівника ▾
Вихід
Особистий кабінет

---

### Створення замовлення

Замовник:

Номер телефону:

Електронна пошта:

Марка авто:

Модель авто:

Рік виготовлення авто:

Тип роботи:

Бажаний день виконання діагностики для призначення часу подальшого ремонту:

Не телефонувати для підтвердження замовлення (Стандартний процес підтвердження повідомлення через месенджер telegram, viber, facebook)

Коментар:

Зберегти

**Навігація**

- Перезаги
- Послуги
- Контакти
- Керівники
- Партнери
- Відгуки

**СТО**

Такі машини, як ця, можуть бігати вічно.  
При належному догляді вони безсмертні.  
— Домінік Торетто

f
s

СТО © 2022

Рисунок 3.2 – Сторінка створення замовлення

Для редагування інформації замовлення використовується відповідна сторінка (див. рисунок 3.3). Поля для зміни інформації заповнюються уже раніше вказаною інформацією, для збереження змін натискається кнопка «Зберегти». Для внесення додаткової інформації про виконавців замовлення та чек-лист використовуються відповідні форми, у формах доступна можливість видалення інформації. Аналогічно виконується процес для редагування інформації про транзакції. Доступ до даної сторінки мають користувачі із статусом «Працівник» та роллю «Менеджер», «Директор».

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

СТО Створити замовлення Функції працівника
Вихід Особистий кабінет

### Редагування Замовлення № 1

#### Інформація

Марка:

Модель:

Рік авто:

Тип роботи:

Ціна:

Дата виконання:

Телефонувати:  Ні

Коментар:

Архів:  Архівувати

Чистий прибуток:

[Зберегти](#)

#### Працівники які виконували замовлення

Ім'я та прізвище	Посада	Дія
_okhotnik_	Директор	<a href="#">Видалити</a>

**Додати працівника**

Ідентифікатор працівника:

[Зберегти](#)

#### Чек-лист

Запчастина, послуга	Ціна	Дія
Бампер	2010	<a href="#">Видалити</a>

**Чек-лист додати елемент**

Запчастина, послуга:

Ціна:

[Зберегти](#)

**Навігація**

- Перевали
- Послуги
- Контакти
- Керівники
- Партнери
- Відгуки

**СТО**

Такі машини, як ця, можуть бігати вічно.  
При належному догляді вони безсмертні.  
— Домінік Торетто

СТО © 2022

Рисунок 3.3 – Сторінка створення замовлення

CRM система (див. рисунок 3.4) містить функцію пошуку замовлення за використанням унікального номеру замовлення. Замовлення містить у собі поля для взаємодії для переходу на сторінку з інформацією про відповідне замовлення, перехід на сторінку для редагування замовлення та видалення замовлення із БД. Зміна статусу відповідає для переміщення у списках виконання системи CRM. Взаємодія із архівом замовлень виконується аналогічно як із системою головною системою CRM. Доступ до даної сторінки мають користувачі із статусом «Працівник» та роллю «Директор».

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

### Керування замовленнями

Кількість усіх замовлень: 5

**ПЕРЕЙТИ В АРХІВ**

Пошук

**Застосувати**

Опрацювання	Підтвержені	Виконання	Виконано
Кількість замовлень: 1	Кількість замовлень: 1	Кількість замовлень: 0	Кількість замовлень: 3
<p>Замовлення № 6</p> <p>Інформація</p> <p>Індифікатор клієнта: test</p> <p>Марка та модель авто:</p> <p>Тип роботи:</p> <p>Дата створення заявки: 2022-03-24 16:54:49.071</p> <p>Дата бажаного початку роботи:</p> <p>Керування інформацією</p> <p><a href="#">Детальніше</a></p> <p><a href="#">Редагувати</a></p> <p><a href="#">Активувати</a></p> <p><a href="#">Видалити</a></p> <p>Змінити статус</p> <p><b>Підтверджено</b></p> <p><b>Виконується</b></p> <p><b>Виконано</b></p>	<p>Замовлення № 3</p> <p>Інформація</p> <p>Керування інформацією</p> <p><a href="#">Детальніше</a></p> <p><a href="#">Редагувати</a></p> <p><a href="#">Активувати</a></p> <p><a href="#">Видалити</a></p> <p>Змінити статус</p>		<p>Замовлення № 1</p> <p>Чистий дохід: 4500</p> <p>Інформація</p> <p>Керування інформацією</p> <p>Змінити статус</p> <p><b>Опрацювання</b></p> <p><b>Підтверджено</b></p> <p><b>Виконується</b></p> <hr/> <p>Замовлення № 4</p> <p>Чистий дохід: 4500</p> <p>Інформація</p> <p>Керування інформацією</p> <p>Змінити статус</p> <hr/> <p>Замовлення № 5</p> <p>Чистий дохід: 1000000</p> <p>Інформація</p> <p>Керування інформацією</p> <p>Змінити статус</p>

**Навігація**

- Перевірити
- Послуги
- Контакти
- Керування
- Партнери
- Відгуки

**СТО**

Такі машини, як ця, можуть бігати вічно.  
При належному догляді вони безсмертні.  
— Домінік Торретто

СТО © 2022

Рисунок 3.4 – CRM система

Для керуванням працівниками використовується спеціальна сторінка (див. рисунок 3.5). Відповідний працівник на даній сторінці може виконати пошук відповідного користувача. Також доступна форма яка використовується для зміни статусу потрібного користувача на статус «Працівник», за допомогою даної

					ДППЗ. 180146.01.08.ВД	Арк. 75
Змн.	Арк.	№ докум.	Підпис	Дата		

форми вказується посада нового працівника та змінюється посада наявного. У формі відображення користувача доступні дії для звільнення користувача. Доступ до даної сторінки мають користувачі із статусом «Працівник» та роллю «Рекрутер», «Директор».

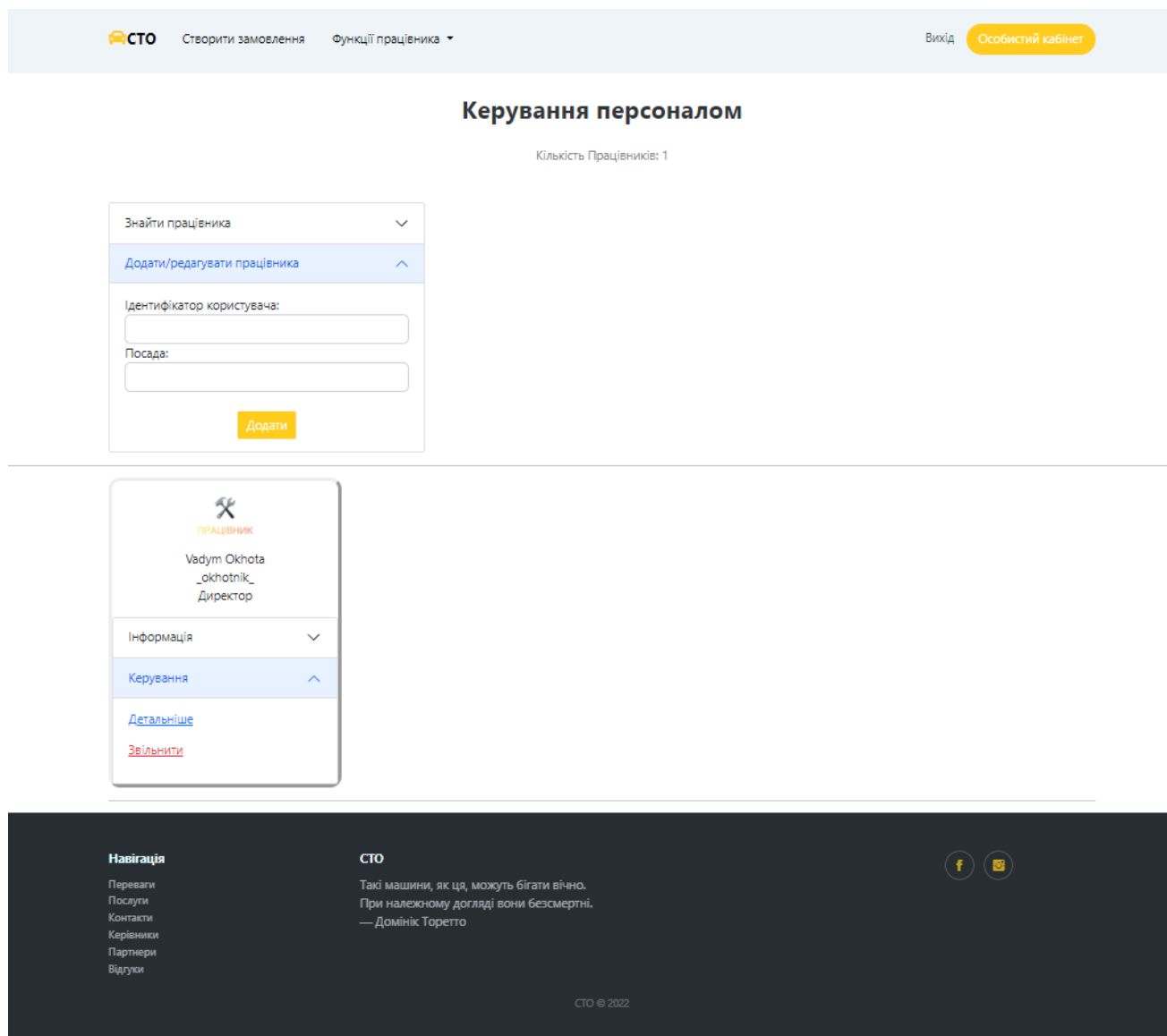


Рисунок 3.5 – сторінка керування персоналом

Сторінка для бухгалтерського обліку (див. малюнок 3.6) надає можливості для вибору місяця для відображення статистики. Також дана сторінка дозволяє створити нову виплату за допомогою двох форм. Перша форма «Подати звіт за витрату» відповідає для створення виплат: зарплати, премії, авансу працівникам підприємства, друга форма «Подати звіт за витрату» слугує для вказування витрат

					ДППЗ. 180146.01.08.ВД	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

на: покупку матеріалів, оплату податків та інших витрат. Доступ до даної сторінки мають користувачі із статусом «Працівник» та роллю «Бухгалтер», «Директор».

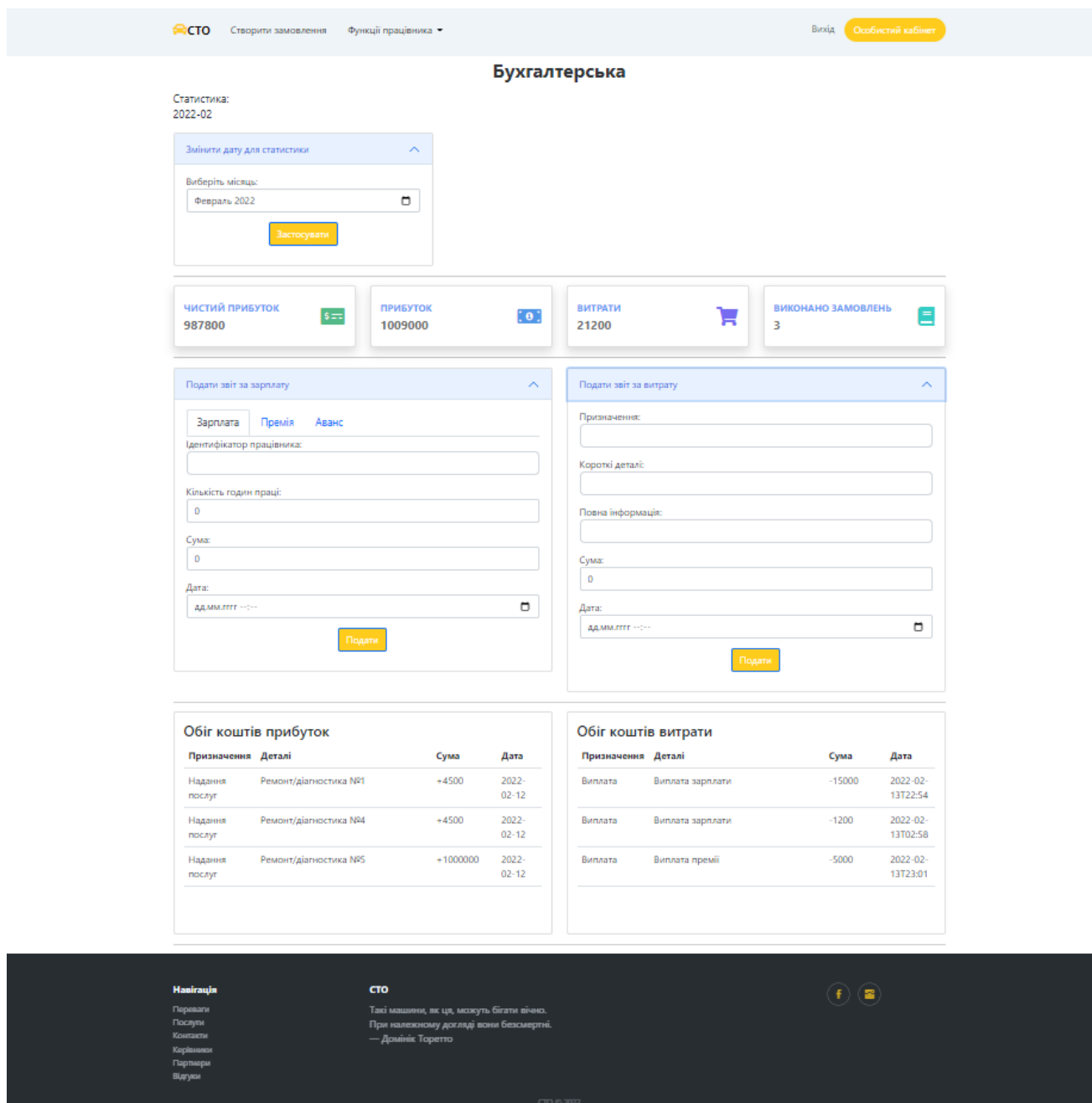


Рисунок 3.6 – Сторінка бухгалтера

Для керування відображення взаємодії користувачів ПЗ використовується сторінка директора (див. рисунок 3.7). Дана сторінка містить форму для фільтрування та відображення інформації за заданим днем.

## Сторінка директора

Управління замовленнями      Управління персоналом      Бухгалтерська

**Системні сповіщення**  
Сповіщення:  
Від першого - До останнього

Фільтр

День:  
ДД.ММ.ГГГГ

**Застосувати**

Заголовок	Вміст	Дата
Виплата	Виконано виплату №1. Сума:45000	2022-03-10T19:53:10.598
Новий відгук	Користувач:@_okhotnik_ Створив новий відгук: Супер!	2022-03-11T13:56:02.345
Новий відгук	Користувач:@_okhotnik_ Створив новий відгук: Тест	2022-03-11T13:56:02.345
Новий відгук	Користувач:@_okhotnik_ Створив новий відгук: Супер	2022-03-11T14:04:05.977
Новий відгук	Користувач:@_okhotnik_ Створив новий відгук: Клас :)	2022-03-11T14:04:05.977
Видалення працівника	Користувач:@_okhotnik_ більше непрацівник	2022-03-16T13:16:55.404
Додавання працівника	Користувач:@_okhotnik_ тепер працівник	2022-03-16T13:16:55.404
Видалення працівника	Користувач:@_okhotnik_ більше непрацівник	2022-03-16T13:16:55.404
Додавання працівника	Користувач:@_okhotnik_ тепер працівник	2022-03-16T13:19:00.439
Новий користувач	Користувач:@vaadym_test_acc приєднався до сервісу	2022-03-16T13:24:24.364

**Навігація**  
Перевали  
Послуги  
Контакти  
Керівники  
Партнери  
Відгуки

**СТО**  
Такі машини, як ця, можуть бігати вічно.  
При належному догляді вони безсмертні.  
— Драйв'єк Torretto

СТО © 2022

Рисунок 3.7 – Сторінка директора

Якщо користувач виконає перехід на сторінку для якої потрібна індивідуальний статус, ПЗ направить даного користувача на особисту сторінку. Якщо користувач не авторизований в сервісі для нього доступний тільки перегляд головної сторінки, якщо гість виконає запит на іншу дію система його перенаправить на сторінку для авторизації.

### 3.3 Вимоги до технічних та програмних засобів

Для використання інтернет-платформи користувачем, комп'ютер користувача має відповідати наступним мінімальним вимогам:

- операційна система: Windows 7, Linux Ubuntu 18.04, Android 4, IOS 7, MacOS 10.5 Leopard;

- встановлений браузер: Chrome, Firefox, Edge, Safari;

- 2 Гб оперативної пам'яті;

- 1 Гб вільного простору на жорсткому диску;

- З'єднання в інтернет на швидкості 0,1 Мбіт/с

Для використання мікросерверів ПЗ, серверний комп'ютер має відповідати наступним мінімальним вимогам:

- операційна система: Windows 7, Linux Ubuntu 18.04, MacOS 10.14;

- 64 бітний процесор із функцією Second Level Address Translation (SLAT);

- 6 Гб оперативної пам'яті;

- активні функції Hyper-V і Containers Windows (Для ОС Windows);

- Linux ядро не старіше версії 3:10;

- Встановлене програмне забезпечення Docker.

### 3.4 Розгортання та встановлення системи

Для розгортання системи потрібно використати програмний застосунок Docker Compose. Застосунок у контейнерах розверне:

- базу даних MySQL;

- базу даних MongoDB;

- мікросервіс Keycloak;

- мікросервіс інтернет-платформи.

										Арк.
										79
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ. 180146.01.08.ВД					

Для взаємодії компонентів системи використовується мережа Docker. Щоб вдало розгорнути ПЗ потрібно використати файл «docker-compose.yml» із налаштуваннями системи:

```
version: "3"

services:

  mongodb:
    container_name: mongo-devnet
    image: mongo
    ports:
      - "2001:27017"
    environment:
      MONGO_INITDB_DATABASE: devmongo
    volumes:
      - ./mongodb/mongod.conf:/etc/mongod.conf
      - ./mongodb/initdb.d:/docker-entrypoint-initdb.d/
      - ./mongodb/data/db:/data/db/
      - ./mongodb/mongodb/data/log:/var/log/mongodb/
    restart: always

  mysql:
    container_name: mysql-devnet
    image: mysql
    environment:
      MYSQL_ROOT_USER: root
      MYSQL_ROOT_PASSWORD: devnet
      MYSQL_DATABASE: keycloak
    volumes:
      - ./mysql/dbdata:/var/lib/mysql
    ports:
      - "2002:3306"
    restart: always

  keycloak:
    image: jboss/keycloak
    container_name: keycloak-devnet
    depends_on:
      - mysql
    environment:
      - DB_VENDOR=MYSQL
      - DB_ADDR=mysql
      - DB_PORT=3306
      - DB_DATABASE=keycloak
      - DB_USER=root
      - DB_PASSWORD=devnet
      - KEYCLOAK_USER=admin
```

					ДППЗ. 180146.01.08.ВД	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```
- KEYCLOAK_PASSWORD=Pa55w0rdOkhotnikPOyt52
- KEYCLOAK_STATISTICS=all
- PROXY_ADDRESS_FORWARDING=true
- VIRTUAL_HOST=https://okcy.pp.ua
volumes:
  - ./keycloak:/data
ports:
  - "2003:8080"
restart: always

site:
  image: okhotnik/sto:latest
  container_name: site
  depends_on:
    - mysqldb
    - keycloak
    - mongodb
  environment:
    SPRING_DATA_MONGODB_URI: mongodb://mongodb:27017/Degree-Project
  ports:
    - "8010:8010"
  restart: always
```

Для початку контейнеризації потрібно із директорії файлу використовуючи командний рядок виконати команду «docker-compose up -d». Після виконання команди Docker завантажить Images із Docker HUB та розгорне елементи системи у контейнерах.

Далі потрібно виконати налаштування мікросервісу Keycloak, потрібні дії для налаштування із адмін консолі Keycloak:

- створюємо новий Realm «Degree-Project»;
- створюємо Clients «Degree-Project-Client»;
- у поле «Valid Redirect URIs» вказуємо посиланням на мікросервіс інтернет-платформи;

Після виконання даних дій інтернет платформа готова до використання та успішно розгорнена.

У розділі було реалізовано програмну систему. Описано усі модулі та графічний інтерфейс. Також описано та продемонстровано реалізацію інтернет-платформи. Створено інструкцію користувача, визначено мінімальні

									Арк.
									81
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ. 180146.01.08.ВД				

характеристики для використання ПЗ. Подано спосіб розгортання та встановлення розробленого програмного забезпечення.

					ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

## 4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вибір та обґрунтування методів тестування застосунку

Для перевірки правильності програмного продукту використовується тестування. Для того щоб протестувати усю систему і зв'язки системи використовують інтеграційне тестування. Інтеграційне тестування – це тестування декількох модулів, методів програмного забезпечення одночасно. Для даного програмного забезпечення даний спосіб тестування ідеальний, адже дозволяє виконати перевірку усіх елементів системи з мінімальним написанням коду тестування.

Оскільки застосунок написаний на фреймворку Spring доцільно використати вбудовані можливості фреймворку. Для можливості використання даних методів тестування використовується бібліотека Spring Test. Саме за допомогою даної бібліотеки можливо відтворювати програмне забезпечення із завантаженням всього контексту для інтеграційного тестування.

Також для тестування доцільно використати фреймворк Mockito, адже дана бібліотека дозволяє знущатися над вибраними елементами програмного продукту. Тобто ми можемо створити заглушку для БД щоб не відбувалося запису тестової інформації, аналогічно ми можемо використовувати дану із сервісами контролерами та методами. У нашому тестуванні основна роль бібліотеки буде використана щоб створити власний тестовий шаблон MVC для можливості повного керування процесу відтворення програми.

Для написання тестування використаємо також бібліотеку Junit, дана бібліотека надає методи для порівняння кінцевих результатів, основні методи бібліотеки:

- assertEquals;
- assertFalse;
- assertNotNull;
- assertNull;
- assertNotSame;

						ДППЗ. 180146.01.08.ВД	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			83

- assertSame;
- assertTrue.

Також буде використано Spring Boot Starter Test – це стартовий набір бібліотек, який включає у себе методи та анотації для створення тесового середовища і реалізації тестових сценаріїв.

#### 4.2 Розробка тестових наборів даних

Для виконання інтеграційного тестування створюємо тестові сценарії (див. таблицю 4.1), завдяки даним сценаріям буде перевірено роботу застосунку. Інтегроване тестування буде відбуватися за створенням та надсилання запиту на контролери, аналогічно із діями користувача що призведе до використання інших елементів ПЗ та отримання статус коду. Якщо статус код дорівнює 200 це означає що дія виконана повністю, допускаються статус 302 та 403 коди, вони означають що інформація яка використовується для тестування не вірна, та користувач не має доступу до даної сторінки. Тест рахується не виконаним якщо статус код має інше значення.

Таблиця 4.1 – тестові сценарії

№	Контролер	Дія тесту	Очікуваний результат
1	2	3	4
1	AccountantPageController	1. Відкрити сторінку бухгалтера; 2. вибрати місяць для відображення статистики; 3. створити нову витрату.	Статус код 200
2	DirectorPageController	1. Відкрити сторінку директора; 2. Виконати фільтрацію логу за днем.	Статус код 200
3	HomeController	1. Відкрити головну сторінку; 2. вийти із облікового запису.	Статус код 200

Продовження таблиці 4.1

4	OrderController	<ol style="list-style-type: none"> <li>1. Відкрити сторінку замовлення №1;</li> <li>2. відкрити сторінку редагування замовлення №1;</li> <li>3. редагувати замовлення №1</li> <li>4. додати виконавця замовлення №1;</li> <li>5. видалити виконавця замовлення №1;</li> <li>6. створення витрати при виконанні замовлення №1;</li> <li>7. вилучення витрати при виконанні замовлення №1;</li> <li>8. відкрити сторінку створення замовлення;</li> <li>9. створити нове замовлення;</li> <li>10. відкрити сторінку керування замовленнями;</li> <li>11. знайти замовлення;</li> <li>12. змінити статус замовлення;</li> <li>13. архівувати замовлення;</li> <li>14. відкрити архів;</li> <li>15. знайти замовлення в архівові.</li> </ol>	Статус код 200
5	PersonalPageController	<ol style="list-style-type: none"> <li>1. Відкрити особисту сторінку користувача «@_okhotnik_»;</li> <li>2. змінити персональну інформацію.</li> </ol>	Статус код 200
6	TeamManagementController	<ol style="list-style-type: none"> <li>1. Відкрити сторінку керування персоналом;</li> <li>2. знайти працівника;</li> <li>3. додати нового працівника;</li> <li>4. вилучити працівника.</li> </ol>	Статус код 200

Змн.	Арк.	№ докум.	Підпис	Дата

## Кінець таблиці 4.1

7	TransactionsInfoController	1. Відкрити сторінку із транзакцією №1; 2. видалити транзакцію №1; 3. змінити інформацію транзакції №1.	Статус код 200
---	----------------------------	---	-------------------

### Фрагмент коду реалізації тесту:

```

@Test
void homePage() throws Exception {
    basicIntegrationsGetTest("/", statusCode[0]);
    basicIntegrationsGetTest("/logout", statusCode[0]);
}

@Test
void accountantPage() throws Exception {
    basicIntegrationsGetTest("/accountant", statusCode[0]);
    basicIntegrationsPostTest("/accountant", statusCode[0]);
    basicIntegrationsPostTest("/accountant/addCoast", statusCode[0]);
}

@Test
void directorPage() throws Exception {
    basicIntegrationsGetTest("/director", statusCode[0]);
    basicIntegrationsPostTest("/director", statusCode[0]);
}

@Test
void orderPages() throws Exception {
    basicIntegrationsGetTest("/order/1", statusCode[0]);
    basicIntegrationsGetTest("/order/1/edit", statusCode[0]);
    basicIntegrationsPostTest("/order/1/edit", statusCode[2]);
    basicIntegrationsPostTest("/order/1/edit/team/add", statusCode[0]);
    basicIntegrationsPostTest("/order/1/edit/team/dell", statusCode[0]);
    basicIntegrationsPostTest("/order/1/edit/check/add", statusCode[0]);
    basicIntegrationsPostTest("/order/1/edit/check/dell", statusCode[0]);
    basicIntegrationsGetTest("/order/create", statusCode[0]);
    basicIntegrationsPostTest("/order/create", statusCode[0]);
    basicIntegrationsGetTest("/order/management", statusCode[0]);
    basicIntegrationsPostTest("/order/management/find", statusCode[0]);
    basicIntegrationsGetTest("/order/management/archive/add", statusCode[0]);
    basicIntegrationsGetTest("/order/management/archive/dell", statusCode[0]);
    basicIntegrationsGetTest("/order/management/order/dell", statusCode[0]);
    basicIntegrationsGetTest("/order/management/order/status/new",
statusCode[0]);
    basicIntegrationsGetTest("/order/management/order/status/confirmed",
statusCode[0]);
    basicIntegrationsGetTest("/order/management/order/status/work",
statusCode[0]);
    basicIntegrationsGetTest("/order/management/order/status/complete",
statusCode[0]);
    basicIntegrationsGetTest("/order/management/archive", statusCode[0]);
    basicIntegrationsGetTest("/order/management/archive", statusCode[0]);
}

```

										Арк.
										86
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ. 180146.01.08.ВД					

```

        basicIntegrationsPostTest("/order/management/archive/find", statusCode[0]);
    }

    @Test
    void personalPage() throws Exception {
        basicIntegrationsGetTest("/@_okhotnik_", statusCode[0]);
        basicIntegrationsPostTest("/@_okhotnik_/edit", statusCode[0]);
    }

    @Test
    void teamManagementPages() throws Exception {
        basicIntegrationsGetTest("/team/management", statusCode[0]);
        basicIntegrationsPostTest("/team/management/find", statusCode[0]);
        basicIntegrationsPostTest("/team/management/add", statusCode[0]);
        basicIntegrationsGetTest("/team/management/dell", statusCode[0]);
    }

    @Test
    void transactionsPage() throws Exception {
        basicIntegrationsGetTest("/transactions/1", statusCode[0]);
        basicIntegrationsGetTest("/transactions/1/dell", statusCode[0]);
        basicIntegrationsPostTest("/transactions/1/edit", statusCode[0]);
    }
}

```

### 4.3 Аналіз результатів тестування

Інформацію про виконання тестів отримуємо за допомогою бібліотеки Junit. Тести проведено у середовищі розробки – IntelliJ IDEA. На рисунку 4.1 зображено виконання тестів.

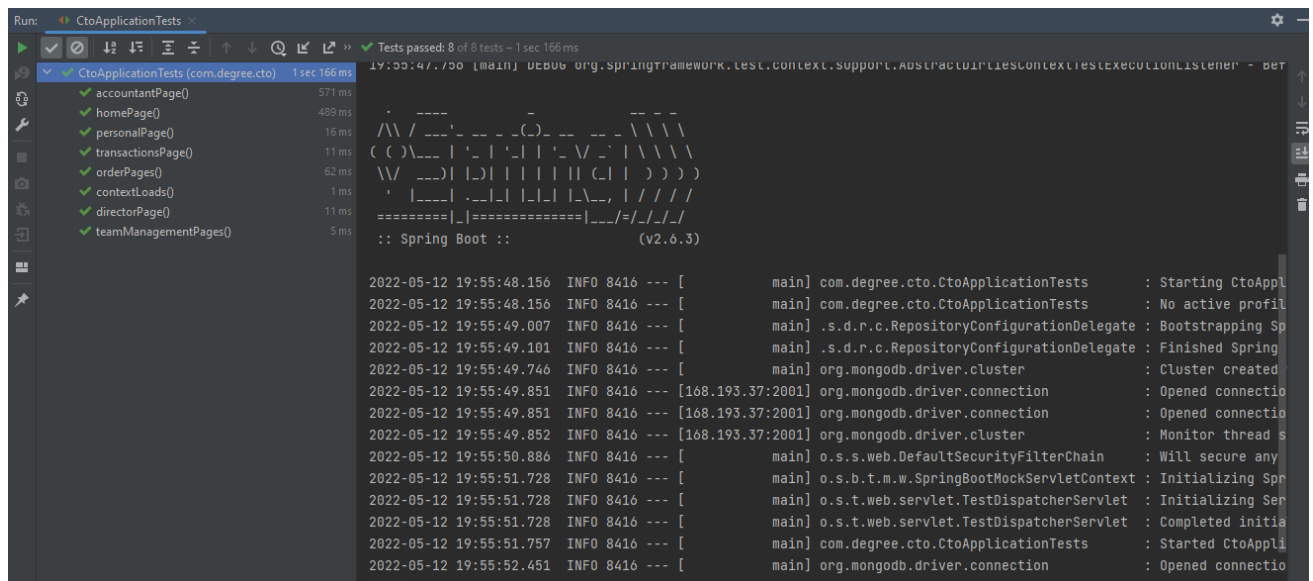


Рисунок 4.1 – Виконання тестів

						ДППЗ. 180146.01.08.ВД	Арк.
							87
Змн.	Арк.	№ докум.	Підпис	Дата			

Усі тестові сценарії виконано успішно (див. таблицю 4.2), тобто всі модулі програмного забезпечення коректно працюють.

Таблиця 4.2 – результати тестових сценаріїв

№ тестового сценарію	Результат	Час виконання
1	2	3
1	Правильно	571 мс
2	Правильно	489 мс
3	Правильно	16 мс
4	Правильно	11 мс
5	Правильно	62 мс
6	Правильно	1 мс
7	Правильно	5 мс

В даному розділі було проаналізовано методи тестування для використання у розробленому програмному забезпеченні. В результаті інтеграційного тесту підтверджено коректну роботу ПЗ, та визначено що програмне забезпечення готове для впровадження.

## ВИСНОВКИ

Із збільшенням кількості автомобілів аналогічно збільшується і кількість сервісних центрів обслуговування, це призводить і до збільшення конкуренції. Тож щоб бізнес мав популярність та привертав увагу нових клієнтів йому потрібна веб-сторінка. А щоб покращити та оптимізувати роботу клієнтів та зменшити використання спеціальних програмних продуктів, можна розробити потрібні функції для інтернет-платформи.

При виконанні дипломного проекту у першому розділі проаналізовано предметну область, визначені основні потреби від подібних систем, Аналіз показав що розробка даного програмного забезпечення затребувана. Програмне забезпечення. В ході дослідження сторонніх подібних програмних засобів визначено основні переваги та недоліки, що допомогло в розробленому програмному продукту виправити недоліки даних сервісів, також перейняти досвід для унаслідування переваг. В результаті було визначено вимоги до програмного забезпечення, створено варіанти використання системи.

Виконуючи другий розділ дипломного проекту було проведено аналіз та опис архітектур які будуть використовуватися в програмному забезпеченні. Це допомогло визначити найсучасніші та найпопулярніші технології розробки для подібних програмних продуктів, щоб забезпечити максимальну оптимізацію та швидкодію програмного забезпечення. Проаналізовано сторонній сервіс авторизації та аутентифікації Keycloak, визначено переваги сервісу та можливості використання у власному ПЗ. Визначено та описано основні модулі програми, для кращого розуміння розробки. Спроектовано БД для оптимального процесу взаємодії з розробленим програмним забезпеченням. Визначено основні сторінки інтерфейсу та побудований дизайн сторінок інтернет-платформи, для максимальної ергономічності, що у свою чергу надасть перевагу серед інших подібних програмних застосунків.

У третьому розділі описане розроблене програмне забезпечення. Також створено керівництво користувача, що у свою чергу полегшить процес

									Арк.
									89
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ. 180146.01.08.ВД				

ознайомлення користувача із програмним забезпеченням. Описано вимоги для використання програмних засобів та процес розгортання ПЗ.

У четвертому розділі визначено підхід та застосунки для тестування програмного забезпечення. В результаті інтеграційного тестування визначено що застосунок коректною працює та готовий для впровадження.

Використання розробленої інтернет-платформи покращить хід роботи на підприємстві. Дозволить замінити стороннє програмне забезпечення. Та оптимізувати процеси у центрі сервісного обслуговування автомобілів. Все це дозволить правильно розподіляти ресурси підприємства та збільшити прибутковість.

					ДППЗ. 180146.01.08.ВД	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Spring Data JPA [Електронний ресурс] / Spring documentation – режим доступу до ресурсу: <https://spring.io/projects/spring-data-jpa> - назва з екрана.

2. Spring Data MongoDB [Електронний ресурс] / Spring documentation – режим доступу до ресурсу: <https://spring.io/projects/spring-data-mongodb> - назва з екрана.

3. Client-Server Architecture for QA Engineers [Електронний ресурс] / Madnes – режим доступу до ресурсу: <https://intellipaat.com/blog/what-is-client-server-architecture/> - назва з екрана.

4. Client Server Architecture [Електронний ресурс] / IntelliPaat – режим доступу до ресурсу: <https://www.qamadness.com/knowledge-base/client-server-architecture-for-qa-engineers/> - назва з екрана.

5. Serving Web Content with Spring MVC [Електронний ресурс] / Spring documentation – режим доступу до ресурсу: <https://spring.io/guides/gs/serving-web-content> - назва з екрана.

6. What is microservices architecture? [Електронний ресурс] / Google Cloud – режим доступу до ресурсу: <https://cloud.google.com/learn/what-is-microservices-architecture> - назва з екрана.

7. Keycloak documentation [Електронний ресурс] / Keycloak documentation – режим доступу до ресурсу: <https://www.keycloak.org/documentation> - назва з екрана.

8. Pattern: Microservice Architecture [Електронний ресурс] / Microservice Architecture – режим доступу до ресурсу: <https://microservices.io/patterns/microservices.html> - назва з екрана.

9. Kubernetes documentation [Електронний ресурс] / Kubernetes documentation – режим доступу до ресурсу: <https://kubernetes.io/uk/docs/home> - назва з екрана.

10. Docker documentation [Електронний ресурс] / Docker documentation – режим доступу до ресурсу: <https://docs.docker.com/> - назва з екрана.

						ДППЗ. 180146.01.08.ВД	Арк.
							91
Змн.	Арк.	№ докум.	Підпис	Дата			

11. What Is Web Architecture? (With Definition and Tips) [Електронний ресурс] / Indeed – режим доступу до ресурсу: <https://www.indeed.com/career-advice/finding-a-job/what-is-web-architecture> - назва з екрана.

12. Web security explained — Understanding web threats [Електронний ресурс] / Mimecast – режим доступу до ресурсу: <https://www.mimecast.com/content/web-security/> - назва з екрана.

13. 10 Most Common Web Security Vulnerabilities [Електронний ресурс] / Toprtal – режим доступу до ресурсу: <https://www.toptal.com/security/10-most-common-web-security-vulnerabilities> - назва з екрана.

14. Spring Security with JWT for REST API [Електронний ресурс] / Toprtal – режим доступу до ресурсу: <https://www.toptal.com/spring/spring-security-tutorial> - назва з екрана.

15. What is CRM? [Електронний ресурс] / Keap – режим доступу до ресурсу: <https://keap.com/product/what-is-crm> - назва з екрана.

16. What is CRM? A definition of CRM and why it's so important for your business? [Електронний ресурс] / Act – режим доступу до ресурсу: <https://www.act.com/what-is-crm/> - назва з екрана.

17. The beginner's guide to CRM: Customer relationship management [Електронний ресурс] / Zapier – режим доступу до ресурсу: <https://www.act.com/what-is-crm/> - назва з екрана.

18. Web Application Testing Complete Guide (How To Test A Website) [Електронний ресурс] / Software Testing Help – режим доступу до ресурсу: <https://www.softwaretestinghelp.com/web-application-testing/> - назва з екрана.

19. A 6-Step Guide to Web Application Testing [Електронний ресурс] / Usersnap – режим доступу до ресурсу: <https://usersnap.com/blog/web-application-testing/> - назва з екрана.

20. Spring Testing [Електронний ресурс] / Spring documentation – режим доступу до ресурсу: <https://docs.spring.io/spring-framework/docs/current/reference/html/testing.html> - назва з екрана.

						ДППЗ. 180146.01.08.ВД	Арк.
							92
Змн.	Арк.	№ докум.	Підпис	Дата			

21. What is Docker? [Электронный ресурс] / IBM – режим доступа до ресурсу: <https://www.ibm.com/cloud/learn/docker> - назва з екрана.

22. What Is Bootstrap? [Электронный ресурс] / Hostinger tutorials – режим доступа до ресурсу: <https://www.hostinger.com/tutorials/what-is-bootstrap/> - назва з екрана.

23. What is Bootstrap? 5 Reasons to love it [Электронный ресурс] / Code instrtute – режим доступа до ресурсу: <https://codeinstitute.net/global/blog/five-reasons-to-love-bootstrap/> - назва з екрана.

24. Tutorial: Using Thymeleaf [Электронный ресурс] / Thymeleaf – режим доступа до ресурсу: <https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html> - назва з екрана.

25. What is MongoDB? [Электронный ресурс] / AWS Amazon – режим доступа до ресурсу: <https://aws.amazon.com/ru/documentdb/what-is-mongodb/> - назва з екрана.

26. Docker MySQL Container: 3 Easy Steps for Setup and Configuration [Электронный ресурс] / Hevo – режим доступа до ресурсу: <https://hevodata.com/learn/docker-mysql/> - назва з екрана.

27. How To Run MongoDB as a Docker Container [Электронный ресурс] / bmc – режим доступа до ресурсу: <https://www.bmc.com/blogs/mongodb-docker-container/> - назва з екрана.

28. Docker Basics and Easy Steps to Dockerize Spring Boot Application [Электронный ресурс] / Medium – режим доступа до ресурсу: <https://medium.com/geekculture/docker-basics-and-easy-steps-to-dockerize-spring-boot-application-17608a65f657> - назва з екрана.

29. Microservices with Spring [Электронный ресурс] / Spring documentation – режим доступа до ресурсу: <https://spring.io/blog/2015/07/14/microservices-with-spring> - назва з екрана.

30. Spring Security [Электронный ресурс] / Spring documentation – режим доступа до ресурсу: <https://docs.spring.io/spring-security/reference/index.html> - назва з екрана.

						ДППЗ. 180146.01.08.ВД	Арк.
							93
Змн.	Арк.	№ докум.	Підпис	Дата			

## ДОДАТОК А (обов'язковий)

### ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ

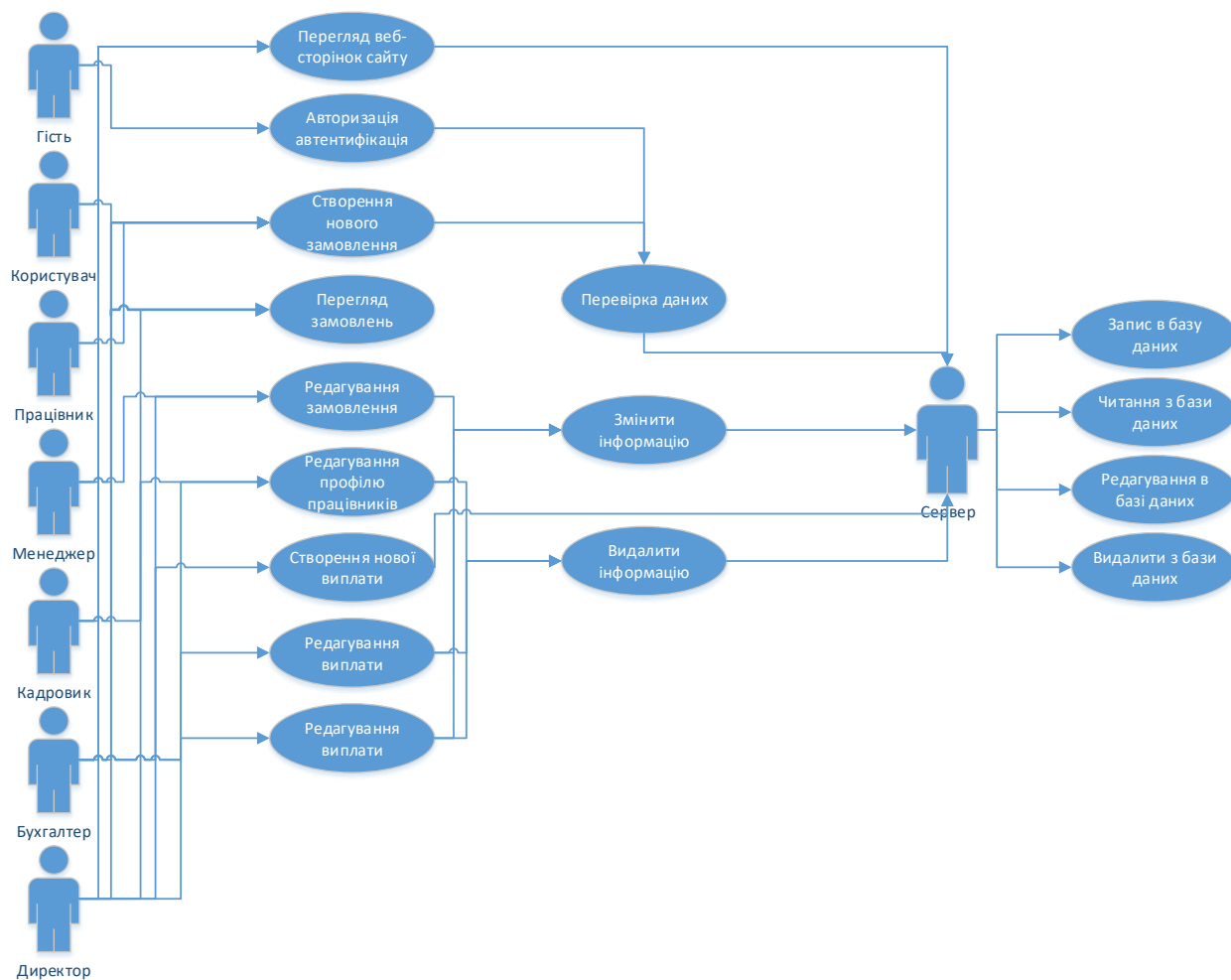


Рисунок А.1 – Діаграма варіантів використання

ДОДАТОК Б  
(обов'язковий)

**ТЕХНІЧНЕ ЗАВДАННЯ**

## **Введення**

Інтернет-платформа із CRM системою включає в себе процеси:

- Створення замовлення;
- Редагування замовлення;
- Керування працівниками;
- Створення та редагування бухгалтерських розрахунків;
- Відображення загальної інформації на сторінках веб-застосунку;
- Відображення активності на сервісі.

Сервіс призначений для використання у сервісах технічного обслуговування автомобілів.

### **1 Ціль розробки**

Створення веб-застосунку для відображення інформації про сервісний центр обслуговування автомобілів, із CRM системою для контролю виконання замовлення, контролю працівників та фінансових операцій.

### **2 Призначення застосунку**

Повний перелік призначення веб-застосунку:

- Відображення інформації про сервісний центр обслуговування автомобілів;
- Створення онлайн заявок на надання послуг;
- Контроль виконання замовлення;
- Створення віртуального чеку;
- Керування працівниками СТО;
- Керування фінансовими операціями;
- Розрахунок сумарних витрат;
- Розрахунок чистого прибутку;
- Ведення списку виконаних замовлень;
- Ведення списку фінансових операцій;
- Ведення списку дій взаємодії із системою.

### **3 Вимоги до сервісу**

- Відображення загальної інформації на головній сторінці доступне усім гостям.
- Авторизація та аутентифікація виконується із валідацією полів.
- Сервіс включає особливі сторінки користувачів.
- Користувачі можуть редагувати особисту інформацію
- Користувачі можуть переглядати інформацію та стан замовлення із електронним чеком.
- Працівники можуть переглянути інформацію за фінансові операції якщо у вони стосуються особистого профілю працівника.
- Сервіс має особливу сторінку для менеджера для керування замовленнями, доступ до сторінки та дій з ними має тільки менеджер та директор.
- Сервіс має особливу сторінку для кадровика для керування працівниками, зміни їх посади та заробітної платні, доступ до сторінки та дій з ними має тільки кадровик та директор
- Сервіс має особливу сторінку для Бухгалтера для керування фінансовими операціями: виплата зарплати, премії, авансу та інших типів фінансових операцій. Керування списком операцій: зміна інформації, видалення. Підрахунок витрат та доходів, виведення чистого прибутку. Доступ до сторінки та дій з ними має тільки бухгалтер та директор.
- Сервіс має особливу сторінку для Директора для відстеження взаємодії користувачів із сервісом. Доступ до сторінки має тільки Директор.
- Програмне забезпечення розроблено для виконання в контейнері Docker.

### **4 Вимоги до надійності**

- Забезпечувати безпеку передачі даних;
- Надавати доступ до заданих сторінок тільки користувачам які мають відповідний рівень доступу.
- Заборона несанкціонованого доступу до даних;
- Оброблення помилок без завершення праці програмного забезпечення.

## **5 Умови експлуатації**

Умови експлуатації мають відповідати загальним технічним нормам експлуатування програмного забезпечення для обчислювальних(комп'ютерних) машин.

## **6 Вимоги до інформаційної та програмної сумісності**

Засіб реалізації – мова програмування JAVA із використанням фреймворка Spring.

Docker – засіб для створення закритого середовища виконання програмного продукту.

Дотримання стандартів ООП за правилами SOLID.

## **7 Стадії та етапи роботи**

- Аналіз роботи: пошук аналогічних сервісів та дослідження позитивних та негативних сторін;
- Створення ескізного проекту
- Визначення методів та проектування системи залежно від закладених потреб сервісу;
- Написання програмного продукту;
- Тестування програмного продукту;
- Доопрацювання сервісу;
- Перевірка на відповідність сервісу до заданих завдань;
- Супровід програмного продукту.

## **8 Порядок контролю та приймання**

Ціль випробувань – перевірка програмного застосунку на відповідність до поставлених вимог та загальної праці системи.

- Перевірка на наявність елементів які були наголошені у вимогах.
- Перевірка авторизації та аутентифікації;
- Перевірка коректності відображення інтерфейсу;
- Перевірка адаптації інтерфейсу під різні розміри екранів;
- Перевірка коректності праці логіки програми;
- Перевірка системи безпеки програмного продукту.

## ДОДАТОК В (обов'язковий)

### КОД (ЛІСТИНГ) ПРОГРАМИ

#### В.1 Програмний код класу AccountantPageController:

```

package com.degree.cto.controllers;

import com.degree.cto.dtos.TransactionsInfoDTO;
import com.degree.cto.logic.Log.LogService;
import com.degree.cto.repositorys.UsersRepository;
import com.degree.cto.security.SecurityService;
import com.degree.cto.services.AccountantPageService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

import javax.servlet.http.HttpServletRequest;

@Controller
public class AccountantPageController {

    @Autowired
    private LogService logService;

    @Autowired
    private AccountantPageService accountantPageService;

    @Autowired
    private SecurityService securityService;

    @Autowired
    private UsersRepository usersRepository;

    @GetMapping("/accountant")
    public String accountantPage(Model model, HttpServletRequest request) {
        if (securityService.customAccess("/accountant",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор") !=
"redirect:/personal-page") {
            //<header>
            try {
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");}

```

```

model.addAttribute("TeamAccessAcc", "yes");}
    if
    (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
    .contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
    } catch (NullPointerException e){}
    //</header>

    long[] moneyInfo =
accountantPageService.ordersDTOListFindAndTransactionsMoney(accountantPageService.defaultDate());
    model.addAttribute("Date", accountantPageService.defaultDate());
    model.addAttribute("Profit", moneyInfo[0]);
    model.addAttribute("OrderSize", moneyInfo[1]);
    model.addAttribute("Costs", moneyInfo[2]);
    model.addAttribute("NetPrice", moneyInfo[0] - moneyInfo[2]);

    model.addAttribute("Transactions",
accountantPageService.transactionsInfoDTOS(accountantPageService.defaultDate()));
    model.addAttribute("TransactionsOrders",
accountantPageService.ordersDTOList(accountantPageService.defaultDate()));
    return securityService.customAccess("accountant-page",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@PostMapping("/accountant")
public String accountantPage(@ModelAttribute(value = "infoDto")
TransactionsInfoDTO infoDTO, Model model, HttpServletRequest request) {
    if (securityService.customAccess("/accountant",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор") !=
"redirect:/personal-page") {
        //<header>
        try {
            if
            (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
            .contentEquals("Менеджер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
            if
            (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
            .contentEquals("Рекрутер")){model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
            if
            (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
            .contentEquals("Бухгалтер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
            if
            (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
            .contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
            } catch (NullPointerException e){}
            //</header>
            long[] moneyInfo =
accountantPageService.ordersDTOListFindAndTransactionsMoney(infoDTO.logicDate);
            model.addAttribute("Date", infoDTO.logicDate);
            model.addAttribute("Profit", moneyInfo[0]);
            model.addAttribute("OrderSize", moneyInfo[1]);
            model.addAttribute("Costs", moneyInfo[2]);
            model.addAttribute("NetPrice", moneyInfo[0] - moneyInfo[2]);

            model.addAttribute("Transactions",
accountantPageService.transactionsInfoDTOS(infoDTO.logicDate));

```

```

        model.addAttribute("TransactionsOrders",
accountantPageService.ordersDTOList(infoDTO.logicDate));

        return securityService.customAccess("accountant-page",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@PostMapping("/accountant/addCoast")
public String accountantAddCoast(@ModelAttribute(value = "tDTO")
TransactionsInfoDTO tDTO, HttpServletRequest request, Model model) {
    //<header>
    try {
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")){model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
    } catch (NullPointerException e){}
    //</header>
    if (securityService.customAccess("accountant-page",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор") !=
"redirect:/personal-page") {
        if (accountantPageService.addCoast(tDTO,
request.getUserPrincipal().getName())) {
            logService.addLog("log", "Виплата", tDTO.getType(), "Виконано виплату
№" + tDTO.getNumber() + ". Сума:" + tDTO.money);
            return "redirect:/accountant";
        } else {
            model.addAttribute("Error", "Помилка: Потрібно заповнити усі поля!");
            return securityService.customAccess("accountant-page",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор");
        }
    } else {
        return "redirect:/personal-page";
    }
}
}
}

```

## В.2 Програмний код класу DirectorPageController:

```

package com.degree.cto.controllers;

import com.degree.cto.logic.Log.FindLogDTO;
import com.degree.cto.logic.Log.LogService;
import com.degree.cto.repositorys.UsersRepository;

```

```

import com.degree.cto.security.SecurityService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

import javax.servlet.http.HttpServletRequest;

@Controller
public class DirectorPageController {

    @Autowired
    private LogService logService;

    @Autowired
    private SecurityService securityService;

    @Autowired
    private UsersRepository usersRepository;

    @GetMapping("/director")
    public String director(Model model, HttpServletRequest request) {
        if (securityService.customAccess("/director",
request.getUserPrincipal().getName(), "Директор", "Директор") != "redirect:/personal-
page") {
            //<header>
            try {
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
                } catch (NullPointerException e) {}
            //</header>

            model.addAttribute("DateLogStart", "Від першого");
            model.addAttribute("DateLogEnd", "До останнього");
            model.addAttribute("Logs", logService.findFilterList(null));
            return securityService.customAccess("director-page",
request.getUserPrincipal().getName(), "Директор", "Директор");
        } else {
            return "redirect:/personal-page";
        }
    }

    @PostMapping("/director")
    public String director(@ModelAttribute(value = "findLogDTO") FindLogDTO
findLogDTO, Model model, HttpServletRequest request) {
        if (securityService.customAccess("/director",
request.getUserPrincipal().getName(), "Директор", "Директор") != "redirect:/personal-
page") {

```



```

private PartnersRepository partnersRepository;
@Autowired
private UsersRepository usersRepository;
@Autowired
private ServicesListRepository servicesListRepository;
@Autowired
private HomeService homeService;

@GetMapping("/")
public String home(Model model, HttpServletRequest request) {
    model.addAttribute("Services", servicesListRepository.findAll());
    model.addAttribute("Team", homeService.managementList());
    model.addAttribute("Partners", partnersRepository.findAll());
    model.addAttribute("Reviews", homeService.reviewsGetList());
    try {
        request.getUserPrincipal().getName();
        model.addAttribute("Auth", "yes");
    } catch (NullPointerException e) {
        model.addAttribute("Auth", "no");
    }

    return "home";
}

@GetMapping(path = "/logout")
public String logout(HttpServletRequest request) throws ServletException {
    request.logout();
    return "redirect:/";
}
}

```

#### B.4 Программный код класу OrderController:

```

package com.degree.cto.controllers;

import com.degree.cto.dtos.OrdersCheckDTO;
import com.degree.cto.dtos.OrdersDTO;
import com.degree.cto.dtos.OrdersWorkTeamDTO;
import com.degree.cto.dtos.UsersDTO;
import com.degree.cto.logic.Log.LogService;
import com.degree.cto.logic.OrderStatusDAO;
import com.degree.cto.repositorys.*;
import com.degree.cto.security.SecurityService;
import com.degree.cto.services.OrderService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import javax.servlet.http.HttpServletRequest;
import java.util.List;

@Controller
public class OrderController {

    @Autowired
    private UsersRepository usersRepository;

```

```

@Autowired
private OrdersRepository ordersRepository;

@Autowired
private OrdersWorkTeamRepository ordersWorkTeamRepository;

@Autowired
private OrdersCheckRepository ordersCheckRepository;

@Autowired
private OrderStatusRepository orderStatusRepository;

@Autowired
private OrderService orderService;

@Autowired
private OrderStatusDAO autoCreateCollections;

@Autowired
private LogService logService;

@Autowired
private SecurityService securityService;

@GetMapping("/order/{numberOrder}")
public String order(@PathVariable(value = "numberOrder") long numberOrder, Model
model, HttpServletRequest request) {
    //<header>
    try {
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")){model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
    } catch (NullPointerException e){}
    //</header>
    OrdersDTO ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
    List<OrdersCheckDTO> ordersCheckDTOList =
ordersCheckRepository.findAllByNumberOrder(numberOrder);
    List<OrdersWorkTeamDTO> ordersWorkTeamDTOList =
ordersWorkTeamRepository.findAllByNumberOrder(numberOrder);
    UsersDTO usersDTO =
usersRepository.findByPersonalIndent(ordersDTO.getUsername());
    model.addAttribute("NumberOrder", numberOrder);
    model.addAttribute("UserName", usersDTO.getName());
    model.addAttribute("UserIndent", usersDTO.getPersonalIndent());
    model.addAttribute("UserPhoneNumber", usersDTO.getPhone());
    model.addAttribute("UserEmail", usersDTO.getEmail());
    model.addAttribute("BrandCar", ordersDTO.getBrandCar());
    model.addAttribute("ModelCar", ordersDTO.getModelCar());
    model.addAttribute("YearsCar", ordersDTO.getYearsCar());
    model.addAttribute("Service", ordersDTO.getService());
    model.addAttribute("Price", ordersDTO.getPrice());

```

```

        model.addAttribute("DateCreateOrder", ordersDTO.getDateCreateOrder());
        model.addAttribute("DateWork", ordersDTO.getDateWork());
        model.addAttribute("PhoneCall", ordersDTO.getPhoneCall());
        model.addAttribute("Message", ordersDTO.getMessage());
        model.addAttribute("StatusWork", ordersDTO.getStatusWork());

        model.addAttribute("OrdersCheckDTOList", ordersCheckDTOList);
        model.addAttribute("ordersWorkTeamDTOList", ordersWorkTeamDTOList);

        return "order";
    }

    @GetMapping("/order/{numberOrder}/edit")
    public String orderEdit(@PathVariable(value = "numberOrder") long numberOrder,
        Model model, HttpServletRequest request) {
        if (securityService.customAccess("order-edit",
            request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
            page") {

            //<header>
            try {
                if
                (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
                    .contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
                model.addAttribute("TeamAccessCRM", "yes");}
                if
                (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
                    .contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
                ;model.addAttribute("TeamAccessTeam", "yes");}
                if
                (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
                    .contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
                model.addAttribute("TeamAccessAcc", "yes");}
                if
                (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
                    .contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
                model.addAttribute("TeamAccessDirector", "yes");}
            } catch (NullPointerException e){}
            //</header>

            autoCreateCollections.CheckCollectionsOrderStatus();

            OrdersDTO ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
            List<OrdersCheckDTO> ordersCheckDTOList =
            ordersCheckRepository.findAllByNumberOrder(numberOrder);
            List<OrdersWorkTeamDTO> ordersWorkTeamDTOList =
            ordersWorkTeamRepository.findAllByNumberOrder(numberOrder);
            model.addAttribute("editSaveLink", "/order/" + numberOrder + "/edit");
            model.addAttribute("editSaveLinkTeamAdd", "/order/" + numberOrder +
            "/edit/team/add");
            model.addAttribute("editSaveLinkCheckAdd", "/order/" + numberOrder +
            "/edit/check/add");
            model.addAttribute("OrderNumber", numberOrder);
            model.addAttribute("BrandCar", ordersDTO.getBrandCar());
            model.addAttribute("ModelCar", ordersDTO.getModelCar());
            model.addAttribute("YearsCar", ordersDTO.getYearsCar());
            model.addAttribute("Service", ordersDTO.getService());
            model.addAttribute("Price", ordersDTO.getPrice());
            model.addAttribute("DateCreateOrder", ordersDTO.getDateCreateOrder());
            model.addAttribute("DateWork", ordersDTO.getDateWork());
            model.addAttribute("PhoneCall", ordersDTO.getPhoneCall());
            model.addAttribute("Message", ordersDTO.getMessage());
            model.addAttribute("StatusWork", ordersDTO.getStatusWork());
            model.addAttribute("Archive", ordersDTO.getArchive());

```

```

model.addAttribute("NetProfit", ordersDTO.getNetProfit());

model.addAttribute("StatusWorkList", orderStatusRepository.findAll());

model.addAttribute("OrdersCheckDTOList", ordersCheckDTOList);
model.addAttribute("ordersWorkTeamDTOList", ordersWorkTeamDTOList);

return securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
} else {
return "redirect:/personal-page";
}
}

@PostMapping("/order/{numberOrder}/edit")
public String orderEditSave(@PathVariable(value = "numberOrder") long
numberOrder, @ModelAttribute(value = "ordersDTOEdit") OrdersDTO ordersDTOEdit, Model
model, HttpServletRequest request) {
//<header>
try {
if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")){model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
} catch (NullPointerException e){}
//</header>
if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
autoCreateCollections.CheckCollectionsOrderStatus();

OrdersDTO ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
List<OrdersCheckDTO> ordersCheckDTOList =
ordersCheckRepository.findAllByNumberOrder(numberOrder);
List<OrdersWorkTeamDTO> ordersWorkTeamDTOList =
ordersWorkTeamRepository.findAllByNumberOrder(numberOrder);

model.addAttribute("ordersWorkTeamDTOList", ordersWorkTeamDTOList);

if (orderService.orderCreate(ordersDTOEdit,
request.getUserPrincipal().getName()) == null) {
logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + "Успішно змінено " + "працівником:@" +
request.getUserPrincipal().getName());
return securityService.customAccess("redirect:/order/" +
ordersDTO.getOrderNumber() + "/edit", request.getUserPrincipal().getName(),
"Менеджер", "Директор");
} else {
return
securityService.customAccess(String.valueOf(model.getAttribute("editSaveLink")),
request.getUserPrincipal().getName(), "Менеджер", "Директор");
}
}
}

```

```

    } else {
        return "redirect:/personal-page";
    }
}

@PostMapping("/order/{numberOrder}/edit/team/add")
public String orderEditAddTeam(@PathVariable(value = "numberOrder") long
numberOrder, @ModelAttribute(value = "ordersWorkTeamDTO") OrdersWorkTeamDTO
ordersWorkTeamDTO, Model model, HttpServletRequest request) {
    if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
        orderService.orderAddTeam(ordersWorkTeamDTO, numberOrder);
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + "Успішно змінено, додано виконавця замовлення: " +
ordersWorkTeamDTO.getUsername());
        return securityService.customAccess("redirect:/order/" + numberOrder +
"/edit", request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@GetMapping("/order/{numberOrder}/edit/team/del")
public String orderEditDelTeam(@RequestParam String del, @PathVariable(value =
"numberOrder") long numberOrder, Model model, HttpServletRequest request) {
    if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
        orderService.orderDellTeam(del, numberOrder);
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + "Успішно змінено, видалено виконавця замовлення: " +
del);
        return securityService.customAccess("redirect:/order/" + numberOrder +
"/edit", request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@PostMapping("/order/{numberOrder}/edit/check/add")
public String orderEditAddCheck(@PathVariable(value = "numberOrder") long
numberOrder, @ModelAttribute(value = "ordersCheckDTO") OrdersCheckDTO ordersCheckDTO,
Model model, HttpServletRequest request) {
    if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
        orderService.orderAddCheck(ordersCheckDTO, numberOrder);
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + "Успішно змінено, додано елемент в чек-лист: " +
ordersCheckDTO.getName());
        return securityService.customAccess("redirect:/order/" + numberOrder +
"/edit", request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@GetMapping("/order/{numberOrder}/edit/check/del")
public String orderEditDelCheck(@RequestParam String del, @PathVariable(value =
"numberOrder") long numberOrder, Model model, HttpServletRequest request) {
    if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {

```

```

        orderService.orderDellCheck(Long.parseLong(del),
Math.toIntExact(numberOrder));
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + "Успішно змінено, видалено елемент з чек-листу " +
del);
        return securityService.customAccess("redirect:/order/" + numberOrder +
"/edit", request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@GetMapping("/order/create")
public String createOrder(HttpServletRequest request, Model model) {
    //<header>
    try {
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")){model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
    } catch (NullPointerException e){}
    //</header>
    UsersDTO usersDTO =
usersRepository.findByPersonalIndent(request.getUserPrincipal().getName());
    model.addAttribute("UserName", usersDTO.getPersonalIndent());
    model.addAttribute("UserPhone", usersDTO.getPhone());
    model.addAttribute("UserEmail", usersDTO.getEmail());
    return "order-create";
}

@PostMapping("/order/create")
public String createOrder(@ModelAttribute("ordersDTO") OrdersDTO ordersDTO,
HttpServletRequest request, Model model) {
    //<header>
    try {
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")){model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
        if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
    } catch (NullPointerException e){}

```

```

        //</header>
        if (orderService.orderCreate(ordersDTO, request.getUserPrincipal().getName())
== null) {
            logService.addLog("log", "Замовлення", "Створення замовлення", "Створено
нове замовлення користувачем:@" + request.getUserPrincipal().getName());
            return "redirect:/order/" + ordersDTO.getOrderNumber();
        } else {
            return "order-create";
        }
    }

    @GetMapping("/order/management")
    public String management(Model model, HttpServletRequest request) {
        //<header>
        try {
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
        } catch (NullPointerException e){}
        //</header>
        if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
            List<OrdersDTO> allOrders = ordersRepository.findByArchive("Hi");
            List<OrdersDTO> newOrders =
ordersRepository.findByStatusWorkAndArchive("Опрацювання", "Hi");
            List<OrdersDTO> confirmedOrders =
ordersRepository.findByStatusWorkAndArchive("Підтверджено", "Hi");
            List<OrdersDTO> workOrders =
ordersRepository.findByStatusWorkAndArchive("Виконується", "Hi");
            List<OrdersDTO> completeOrders =
ordersRepository.findByStatusWorkAndArchive("Виконано", "Hi");

            model.addAttribute("AllOrderListSize", allOrders.size());
            model.addAttribute("NewOrders", newOrders);
            model.addAttribute("ConfirmedOrders", confirmedOrders);
            model.addAttribute("WorkOrders", workOrders);
            model.addAttribute("CompleteOrders", completeOrders);

            model.addAttribute("NewOrdersSize", newOrders.size());
            model.addAttribute("ConfirmedOrdersSize", confirmedOrders.size());
            model.addAttribute("WorkOrdersSize", workOrders.size());
            model.addAttribute("CompleteOrdersSize", completeOrders.size());

            return securityService.customAccess("order-management",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
        } else {
            return "redirect:/personal-page";
        }
    }
}

```

```

    @PostMapping("/order/management/find")
    public String managementFind(@ModelAttribute("ordersDTO") OrdersDTO ordersDTO,
    Model model, HttpServletRequest request) {
        if (securityService.customAccess("order-edit",
    request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
    page") {
            if (ordersDTO.getOrderNumber() == 0) {
                return "redirect:/order/management";
            } else {
                //<header>
                try {
                    if
    (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
    .contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
    model.addAttribute("TeamAccessCRM", "yes");}
                    if
    (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
    .contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
    ;model.addAttribute("TeamAccessTeam", "yes");}
                    if
    (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
    .contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
    model.addAttribute("TeamAccessAcc", "yes");}
                    if
    (usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
    .contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
    model.addAttribute("TeamAccessDirector", "yes");}
                } catch (NullPointerException e) {}
                //</header>
                List<OrdersDTO> allOrders = ordersRepository.findByArchive("Hi");
                List<OrdersDTO> newOrders =
    ordersRepository.findByOrderNumberAndStatusWorkAndArchive(ordersDTO.orderNumber,
    "Опрацювання", "Hi");
                List<OrdersDTO> confirmedOrders =
    ordersRepository.findByOrderNumberAndStatusWorkAndArchive(ordersDTO.orderNumber,
    "Підтверджено", "Hi");
                List<OrdersDTO> workOrders =
    ordersRepository.findByOrderNumberAndStatusWorkAndArchive(ordersDTO.orderNumber,
    "Виконується", "Hi");
                List<OrdersDTO> completeOrders =
    ordersRepository.findByOrderNumberAndStatusWorkAndArchive(ordersDTO.orderNumber,
    "Виконано", "Hi");

                model.addAttribute("AllOrderListSize", allOrders.size());
                model.addAttribute("NewOrders", newOrders);
                model.addAttribute("ConfirmedOrders", confirmedOrders);
                model.addAttribute("WorkOrders", workOrders);
                model.addAttribute("CompleteOrders", completeOrders);

                model.addAttribute("NewOrdersSize", newOrders.size());
                model.addAttribute("ConfirmedOrdersSize", confirmedOrders.size());
                model.addAttribute("WorkOrdersSize", workOrders.size());
                model.addAttribute("CompleteOrdersSize", completeOrders.size());

                return securityService.customAccess("order-management",
    request.getUserPrincipal().getName(), "Менеджер", "Директор");
            }
        } else {
            return "redirect:/personal-page";
        }
    }
}

```

```

    @GetMapping("/order/management/archive/add")
    public String orderArchiveAdd(@RequestParam long numberOrder, HttpServletRequest
request) {
        if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
            OrdersDTO ordersDTO = new OrdersDTO();
            ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
            ordersDTO.setArchive("Так");
            ordersRepository.save(ordersDTO);
            logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + "Перенесено до архіву");
            return securityService.customAccess("redirect:/order/management",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
        } else {
            return "redirect:/personal-page";
        }
    }

    @GetMapping("/order/management/archive/dell")
    public String orderArchiveDell(@RequestParam long numberOrder, HttpServletRequest
request) {
        if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
            OrdersDTO ordersDTO = new OrdersDTO();
            ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
            ordersDTO.setArchive("Hi");
            ordersRepository.save(ordersDTO);
            logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + "Перенесено з архіву");
            return securityService.customAccess("redirect:/order/management/archive",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
        } else {
            return "redirect:/personal-page";
        }
    }

    @GetMapping("/order/management/order/dell")
    public String orderOrderDell(@RequestParam long numberOrder, HttpServletRequest
request) {
        if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
            OrdersDTO ordersDTO = new OrdersDTO();
            ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
            ordersRepository.delete(ordersDTO);
            logService.addLog("log", "Замовлення", "Видалено замовлення", "Замовлення
№" + numberOrder);
            return securityService.customAccess("redirect:/order/management",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
        } else {
            return "redirect:/personal-page";
        }
    }

    @GetMapping("/order/management/order/status/new")
    public String orderOrderStatusNew(@RequestParam long numberOrder,
HttpServletRequest request) {
        if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
            OrdersDTO ordersDTO = new OrdersDTO();

```

```

        ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
        ordersDTO.setStatusWork("Опрацювання");
        ordersRepository.save(ordersDTO);
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + ". Змінено статус на опрацювання");
        return securityService.customAccess("redirect:/order/management",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@GetMapping("/order/management/order/status/confirmed")
public String orderStatusConfirmed(@RequestParam long numberOrder,
HttpServletRequest request) {
    if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
        OrdersDTO ordersDTO = new OrdersDTO();
        ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
        ordersDTO.setStatusWork("Підтверджено");
        ordersRepository.save(ordersDTO);
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + ". Змінено статус на підтверджено");
        return securityService.customAccess("redirect:/order/management",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@GetMapping("/order/management/order/status/work")
public String orderStatusWork(@RequestParam long numberOrder, HttpServletRequest
request) {
    if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
        OrdersDTO ordersDTO = new OrdersDTO();
        ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
        ordersDTO.setStatusWork("Виконується");
        ordersRepository.save(ordersDTO);
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + ". Змінено статус на виконується");
        return securityService.customAccess("redirect:/order/management",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        return "redirect:/personal-page";
    }
}

@GetMapping("/order/management/order/status/complete")
public String orderStatusComplete(@RequestParam long numberOrder,
HttpServletRequest request) {
    if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
        OrdersDTO ordersDTO = new OrdersDTO();
        ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
        ordersDTO.setStatusWork("Виконано");
        ordersRepository.save(ordersDTO);
        logService.addLog("log", "Замовлення", "Редагування замовлення",
"Замовлення №" + numberOrder + ". Змінено статус на виконано");
        return securityService.customAccess("redirect:/order/management",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
    }
}

```

```

        } else {
            return "redirect:/personal-page";
        }
    }

    @GetMapping("/order/management/archive")
    public String managementArchive(Model model, HttpServletRequest request) {
        if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
            //<header>
            try {
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
            } catch (NullPointerException e) {}
            //</header>
            List<OrdersDTO> ordersDTOS = ordersRepository.findByArchive("Так");
            model.addAttribute("OrdersSize", ordersDTOS.size());
            model.addAttribute("Orders", ordersDTOS);
            return securityService.customAccess("order-archive",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
        } else {
            return "redirect:/personal-page";
        }
    }

    @PostMapping("/order/management/archive/find")
    public String managementArchiveFind(@ModelAttribute("ordersDTO") OrdersDTO
ordersDTO, Model model, HttpServletRequest request) {
        if (securityService.customAccess("order-edit",
request.getUserPrincipal().getName(), "Менеджер", "Директор") != "redirect:/personal-
page") {
            //<header>
            try {
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
            }

```

```

    } catch (NullPointerException e){}
    //</header>
    if (ordersDTO.getOrderNumber() == 0) {
        return
    securityService.customAccess("redirect:/order/management/archive",
    request.getUserPrincipal().getName(), "Менеджер", "Директор");
    } else {
        List<OrdersDTO> ordersDTOS =
ordersRepository.findByArchiveAndOrderNumber("Так", ordersDTO.orderNumber);
        model.addAttribute("OrdersSize", ordersDTOS.size());
        model.addAttribute("Orders", ordersDTOS);
        return securityService.customAccess("order-archive",
request.getUserPrincipal().getName(), "Менеджер", "Директор");
    }
    } else {
        return "redirect:/personal-page";
    }
}
}
}

```

## В.5 Програмный код класу PersonalPageController:

```

package com.degree.cto.controllers;

import com.degree.cto.dtos.ReviewsDTO;
import com.degree.cto.dtos.UsersDTO;
import com.degree.cto.logic.Log.LogService;
import com.degree.cto.repositories.OrdersRepository;
import com.degree.cto.repositories.TransactionsInfoRepository;
import com.degree.cto.repositories.UsersRepository;
import com.degree.cto.security.SecurityService;
import com.degree.cto.services.PersonalPageService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;

import javax.servlet.http.HttpServletRequest;

@Controller
public class PersonalPageController {

    @Autowired
    private SecurityService securityService;

    @Autowired
    private PersonalPageService personalPageService;

    @Autowired
    private UsersRepository usersRepository;

    @Autowired
    private TransactionsInfoRepository transactionsInfoRepository;

    @Autowired
    private OrdersRepository ordersRepository;

```

```

@Autowired
private LogService logService;

@GetMapping("/personal-page")
public String personalsPage(HttpServletRequest request) {
    return securityService.profileCheck("/personal-page", "redirect:/" +
request.getUserPrincipal().getName(), request.getUserPrincipal().getName());
}

@GetMapping("/{username}")
public String personalPage(@PathVariable(value = "username") String username,
Model model, HttpServletRequest request) {
    try {
        //<header>
        try {
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
        } catch (NullPointerException e){}
        //</header>

        model.addAttribute("EditAccess",
personalPageService.editAccess(request.getUserPrincipal().getName(), username));

        UsersDTO usersDTO = usersRepository.findByPersonalIndent(username);
        model.addAttribute("nameUser", usersDTO.getName());
        model.addAttribute("emailUser", usersDTO.getEmail());
        model.addAttribute("usernameUser", usersDTO.getPersonalIndent());
        model.addAttribute("phoneUser", usersDTO.getPhone());
        model.addAttribute("photoUser", usersDTO.getPhoto_url());
        model.addAttribute("statusUser", usersDTO.getStatus());
        model.addAttribute("roleUser", usersDTO.getRole());

        model.addAttribute("TransactionsInfoList",
transactionsInfoRepository.findAllByUsername(username));

        model.addAttribute("OrdersForAccountList",
ordersRepository.findAllByUsername(username));

        return "personal-page";
    } catch (Exception exception) {
        return personalPageService.pageNotFond(username);
    }
}

@PostMapping("/{username}")
public String personalPage(@ModelAttribute("reviewsDTO") ReviewsDTO reviewsDTO,
@PathVariable(value = "username") String username, HttpServletRequest request, Model
model) {

```

```

        model.addAttribute("EditAccess",
personalPageService.editAccess(request.getUserPrincipal().getName(), username));

        //<header>
        try {
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")){model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")){model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
        } catch (NullPointerException e){}
        //</header>

        UsersDTO usersDTO = usersRepository.findByPersonalIndent(username);
        model.addAttribute("nameUser", usersDTO.getName());
        model.addAttribute("emailUser", usersDTO.getEmail());
        model.addAttribute("usernameUser", usersDTO.getPersonalIndent());
        model.addAttribute("phoneUser", usersDTO.getPhone());
        model.addAttribute("photoUser", usersDTO.getPhoto_url());
        model.addAttribute("statusUser", usersDTO.getStatus() + " | " +
usersDTO.getRole());
        model.addAttribute("roleUser", usersDTO.getRole());

        model.addAttribute("TransactionsInfoList",
transactionsInfoRepository.findAllByUsername(username));

        model.addAttribute("OrdersForAccountList",
ordersRepository.findAllByUsername(username));

        if (reviewsDTO.getText().isEmpty()) {
            model.addAttribute("ReviewStatus", "Помилка відгук пустий!");
            return "personal-page";
        } else {
            personalPageService.createReviews(reviewsDTO,
request.getUserPrincipal().getName());
            model.addAttribute("ReviewStatus", "Дякуємо за відгук!");
            logService.addLog("log", "Відгук", "Новий відгук", "Користувач:@" +
username + " Створив новий відгук: " + reviewsDTO.getText());
            return "personal-page";
        }
    }

    @PostMapping("/{username}/edit")
    public String personalPageEdit(@ModelAttribute("usersDTO") UsersDTO usersDTO,
@PathVariable(value = "username") String username, HttpServletRequest request) {
        if (username.contentEquals(request.getUserPrincipal().getName())) {
            UsersDTO dto = usersRepository.findByPersonalIndent(username);
            dto.setName(usersDTO.getName());
            dto.setPhone(usersDTO.getPhone());
            dto.setEmail(usersDTO.getEmail());
            usersRepository.save(dto);
            logService.addLog("log", "Користувачі", "Зміна особистої інформації",
"Користувач:@" + username + " змінив особисту інформацію.");

```

```

        return "redirect:/" + username;
    } else {
        return "redirect:/personal-page";
    }
}
}

```

## В.6 Програмный код класу TeamManagementController:

```

package com.degree.cto.controllers;

import com.degree.cto.dtos.UsersDTO;
import com.degree.cto.logic.Log.LogService;
import com.degree.cto.repositorys.UsersRepository;
import com.degree.cto.security.SecurityService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import javax.servlet.http.HttpServletRequest;
import java.util.List;

@Controller
public class TeamManagementController {

    @Autowired
    private UsersRepository usersRepository;

    @Autowired
    private LogService logService;

    @Autowired
    private SecurityService securityService;

    @GetMapping("/team/management")
    public String teamManagement(Model model, HttpServletRequest request) {
        if (securityService.customAccess("/team/management",
request.getUserPrincipal().getName(), "Рекрутер", "Директор") != "redirect:/personal-
page") {
            //<header>
            try {
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");}
model.addAttribute("TeamAccessCRM", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");}
model.addAttribute("TeamAccessAcc", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()

```

```

        .contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
        } catch (NullPointerException e){}
        //</header>
        List<UsersDTO> usersDTOList = usersRepository.findByStatus("Працівник");
        model.addAttribute("teamListSize", usersDTOList.size());
        model.addAttribute("teamList", usersDTOList);
        return "team-management";
    } else {
        return "redirect:/personal-page";
    }
}

@PostMapping("/team/management/find")
public String teamManagementFind(@ModelAttribute(value = "usersDTO") UsersDTO
usersDTO, Model model, HttpServletRequest request) {
    if (securityService.customAccess("/team/management",
request.getUserPrincipal().getName(), "Рекрутер", "Директор") != "redirect:/personal-
page") {
        //<header>
        try {
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
        } catch (NullPointerException e){}
        //</header>
        UsersDTO usersDTLogic =
usersRepository.findByPersonalIndent(usersDTO.getPersonalIndent());
        try {
            if (usersDTLogic == null && usersDTLogic.getStatus() != "Працівник")
{
                model.addAttribute("Error", "Користувача не знайдено");
            } else {
                List<UsersDTO> usersDTOList =
usersRepository.findByPersonalIndentAndStatus(usersDTO.getPersonalIndent(),
"Працівник");
                model.addAttribute("teamListSize", usersDTOList.size());
                model.addAttribute("teamList", usersDTLogic);
            }
        } catch (Exception e) {
            model.addAttribute("Error", "Користувача не знайдено");
        }
        return "team-management";
    } else {
        return "redirect:/personal-page";
    }
}

@PostMapping("/team/management/add")
public String teamManagementAdd(@ModelAttribute(value = "usersDTO") UsersDTO
usersDTO, Model model, HttpServletRequest request) {

```

```

        if (securityService.customAccess("/team/management",
request.getUserPrincipal().getName(), "Рекрутер", "Директор") != "redirect:/personal-
page") {
            //<header>
            try {
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
                if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
            } catch (NullPointerException e){}
            //</header>
            try {
                UsersDTO usersDTOLogic =
usersRepository.findByPersonalIndent(usersDTO.getPersonalIndent());
                usersDTOLogic.setStatus("Працівник");
                usersDTOLogic.setRole(usersDTO.getRole());
                int randomAvatar = 1 + (int) (Math.random() * 4);
                usersDTOLogic.setPhoto_url("/assets/img/avatars/teams/" +
randomAvatar + ".JPG");
                usersRepository.save(usersDTOLogic);
                logService.addLog("log", "Працівники", "Додавання працівника",
"Користувач:@" + usersDTO.getPersonalIndent() + " тепер працівник");
                return "redirect:/team/management";
            } catch (Exception e) {
                model.addAttribute("Error", "Користувача не знайдено");
                return "team-management";
            }
        } else {
            return "redirect:/personal-page";
        }
    }

    @GetMapping("/team/management/dell")
    public String teamManagementDell(@RequestParam String username,
HttpServletRequest request) {
        if (securityService.customAccess("/team/management",
request.getUserPrincipal().getName(), "Рекрутер", "Директор") != "redirect:/personal-
page") {
            UsersDTO usersDTOLogic = usersRepository.findByPersonalIndent(username);
            usersDTOLogic.setStatus("Клієнт");
            usersDTOLogic.setRole("Клієнт");
            usersRepository.save(usersDTOLogic);
            logService.addLog("log", "Працівники", "Видалення працівника",
"Користувач:@" + username + " більше непрацівник");
            return "redirect:/team/management";
        } else {
            return "redirect:/personal-page";
        }
    }
}

```

## V.7 Програмний код класу TransactionsInfoController:

```

package com.degree.cto.controllers;

import com.degree.cto.dtos.TransactionsInfoDTO;
import com.degree.cto.logic.Log.LogService;
import com.degree.cto.repositorys.TransactionsInfoRepository;
import com.degree.cto.repositorys.UsersRepository;
import com.degree.cto.security.SecurityService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;

import javax.servlet.http.HttpServletRequest;

@Controller
public class TransactionsInfoController {

    @Autowired
    private TransactionsInfoRepository transactionsInfoRepository;

    @Autowired
    private UsersRepository usersRepository;

    @Autowired
    private LogService logService;

    @Autowired
    private SecurityService securityService;

    @GetMapping("/transactions/{numberTransactions}")
    public String transactions(@PathVariable(value = "numberTransactions") long
numberTransactions, Model model, HttpServletRequest request) {
        //<header>
        try {
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Менеджер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessCRM", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Рекрутер")) {model.addAttribute("TeamAccess", "yes")
;model.addAttribute("TeamAccessTeam", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Бухгалтер")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessAcc", "yes");}
            if
(usersRepository.findByPersonalIndent(request.getUserPrincipal().getName()).getRole()
.contentEquals("Директор")) {model.addAttribute("TeamAccess", "yes");
model.addAttribute("TeamAccessDirector", "yes");}
        } catch (NullPointerException e){}
        //</header>
        TransactionsInfoDTO transactionsInfoDTO =
transactionsInfoRepository.findByNumber(numberTransactions);
        model.addAttribute("Number", transactionsInfoDTO.getNumber());
    }
}

```

```

        model.addAttribute("Type", transactionsInfoDTO.getType());
        model.addAttribute("Username", transactionsInfoDTO.getUsername());
        model.addAttribute("ClockWork", transactionsInfoDTO.getClocks_work());
        model.addAttribute("Info", transactionsInfoDTO.getInfo());
        model.addAttribute("FullInfo", transactionsInfoDTO.getFull_info());
        model.addAttribute("Money", transactionsInfoDTO.getMoney());
        model.addAttribute("UsernameCreator",
transactionsInfoDTO.getUsernameCreator());
        model.addAttribute("CreatorRole",
usersRepository.findByPersonalIndent(transactionsInfoDTO.getUsernameCreator()).getRole());
        model.addAttribute("Date", transactionsInfoDTO.getDate());
        if (securityService.customAccess("/accountant",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор") !=
"redirect:/personal-page") {
            model.addAttribute("EditAccess", "yes");
        } else {
            model.addAttribute("EditAccess", "no");
        }
        return "transactions-info";
    }

    @PostMapping("/transactions/{numberTransactions}/edit")
    public String transactionsEdit(@PathVariable(value = "numberTransactions") long
numberTransactions, @ModelAttribute(value = "transactionsInfoDTO")
TransactionsInfoDTO transactionsInfoDTO, HttpServletRequest request) {
        if (securityService.customAccess("/accountant",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор") !=
"redirect:/personal-page") {
transactionsInfoDTO.setId(transactionsInfoRepository.findByNumber(numberTransactions)
.getId());
            transactionsInfoDTO.setNumber(numberTransactions);
            transactionsInfoRepository.save(transactionsInfoDTO);
            logService.addLog("log", "Виплати", "Змінено інформацію про виплату",
"Виплата №" + numberTransactions + " працівником:@" +
transactionsInfoRepository.findByNumber(numberTransactions).getUsernameCreator());
            return "redirect:/transactions/" + numberTransactions;
        } else {
            return "redirect:/personal-page";
        }
    }

    @GetMapping("/transactions/{numberTransactions}/dell")
    public String transactionsDell(@PathVariable(value = "numberTransactions") long
numberTransactions, HttpServletRequest request) {
        if (securityService.customAccess("/accountant",
request.getUserPrincipal().getName(), "Бухгалтер", "Директор") !=
"redirect:/personal-page") {
            logService.addLog("log", "Виплати", "Видалено інформацію про виплату",
"Виплата №" + numberTransactions + " працівником:@" +
transactionsInfoRepository.findByNumber(numberTransactions).getUsernameCreator());
            TransactionsInfoDTO transactionsInfoDTO =
transactionsInfoRepository.findByNumber(numberTransactions);
            transactionsInfoRepository.delete(transactionsInfoDTO);
            return "redirect:/accountant/";
        } else {
            return "redirect:/personal-page";
        }
    }
}

```

## B.8 Програмний код класу SecurityController:

```

package com.degree.cto.security;

import org.keycloak.KeycloakPrincipal;
import org.keycloak.KeycloakSecurityContext;
import org.keycloak.representations.AccessToken;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.annotation.CurrentSecurityContext;
import org.springframework.security.core.context.SecurityContext;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;

import javax.servlet.http.HttpServletRequest;
import java.security.Principal;

@Controller
public class SecurityController {

    @Autowired
    private SecurityService securityService;

    @GetMapping("/authorize/confirm")
    public String authorize(@CurrentSecurityContext SecurityContext securityContext,
        HttpServletRequest request, @RequestParam String url) {
        Principal principal = (Principal)
        securityContext.getAuthentication().getPrincipal();
        KeycloakPrincipal<KeycloakSecurityContext> keycloakPrincipal =
        (KeycloakPrincipal<KeycloakSecurityContext>) principal;
        AccessToken accessToken =
        keycloakPrincipal.getKeycloakSecurityContext().getToken();
        securityService.createProfile(
            accessToken.getGivenName(),
            accessToken.getFamilyName(),
            accessToken.getEmail(),
            String.valueOf(accessToken.getRealmAccess().getRoles()),
            request.getUserPrincipal().getName()
        );
        return "redirect:" + url;
    }
}

```

## B.9 Програмний код класу LogService:

```

package com.degree.cto.logic.Log;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.sql.Timestamp;
import java.util.List;

@Service
public class LogService {

```

```

@Autowired
private LogRepository logRepository;

Timestamp timestamp = new Timestamp(System.currentTimeMillis());

public void addLog(String type, String name, String title, String message) {
    LogDTO log = new LogDTO();
    log.setType(type);
    log.setName(name);
    log.setTitle(title);
    log.setMessage(message);
    log.setDate(timestamp.toLocalDateTime().toString());
    log.setLogicDateDays(createLogicDate());
    System.out.println("-----log-----\n" + title + "\n" + message +
"\n" + log.getDate() + "\n-----");
    logRepository.save(log);
}

private String createLogicDate() {
    String date = timestamp.toLocalDateTime().toString();
    return date.substring(0,10);
}

public List<LogDTO> findFilterList(String date) {
    if (date != null) {
        return findByLogicDateDays(date);
    } else {
        return findAllLogs();
    }
}

private List<LogDTO> findAllLogs() {
    return logRepository.findAll();
}

private List<LogDTO> findByLogicDateDays(String date) {
    return logRepository.findByLogicDateDays(date);
}
}

```

## B.10 Програмный код класу SecurityService:

```

package com.degree.cto.security;

import com.degree.cto.dtos.UsersDTO;
import com.degree.cto.logic.OrderStatusDAO;
import com.degree.cto.logic.Log.LogService;
import com.degree.cto.repositories.UsersRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class SecurityService {

    @Autowired
    private UsersRepository userRepository;

    @Autowired

```

```

private LogService logService;

public String profileCheck(String url, String pageName, String personal_indent) {
    if (userRepository.findByPersonalIndent(personal_indent) == null) {
        return "redirect:/authorize/confirm?url=" + url;
    } else {
        return pageName;
    }
}

public void createProfile(String firstName, String lastName, String email, String
roles, String username) {
    UsersDTO usersDTO = new UsersDTO();
    usersDTO.setName(firstName + " " + lastName);
    usersDTO.setEmail(email);
    usersDTO.setPersonalIndent(username);
    usersDTO.setStatus("Клієнт");
    usersDTO.setRole("");
    int randomAvatar = 1 + (int) (Math.random() * 10);
    usersDTO.setPhoto_url("/assets/img/avatars/clients/" + randomAvatar +
".JPG");
    userRepository.save(usersDTO);
    logService.addLog("log", "Користувачі", "Новий користувач", "Користувач:@" +
username + " приєднався до сервісу");
}

public String customAccess(String page, String username, String role_1, String
role_2) {
    try {
        String userRole =
userRepository.findByPersonalIndent(username).getRole();
        if (userRole.contentEquals(role_1) || userRole.contentEquals(role_2)) {
            return page;
        } else {
            return "redirect:/personal-page";
        }
    } catch (Exception e) {
        return "redirect:/personal-page";
    }
}
}

```

## B.11 Програмний код класу AccountantPageService:

```

package com.degree.cto.services;

import com.degree.cto.dtos.OrdersDTO;
import com.degree.cto.dtos.TransactionsInfoDTO;
import com.degree.cto.repositorys.OrdersRepository;
import com.degree.cto.repositorys.TransactionsInfoRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.sql.Timestamp;
import java.util.List;

@Service
public class AccountantPageService {

```

```

@Autowired
private OrdersRepository ordersRepository;
@Autowired
private TransactionsInfoRepository transactionsInfoRepository;

public String defaultDate() {
    Timestamp timestamp = new Timestamp(System.currentTimeMillis());
    String month = null;
    if (timestamp.toLocalDateTime().getMonthValue() < 10) {
        month = "0" + timestamp.toLocalDateTime().getMonthValue();
    } else {
        month = String.valueOf(timestamp.toLocalDateTime().getMonthValue());
    }
    return timestamp.toLocalDateTime().getYear() + "-" + month;
}

public long[] ordersDTOListFindAndTransactionsMoney(String date) {
    String zeroI = null;
    long[] ordersInfo = new long[3];
    for (int i = 0; i <= 31; i++) {

        if (i < 10) {
            zeroI = "0" + i;
        } else {
            zeroI = String.valueOf(i);
        }

        if (ordersRepository.findByDateWorkAndStatusWork(date + "-" + zeroI,
"Виконано").size() >= 1) {
            List<OrdersDTO> ordersDTO =
ordersRepository.findByDateWorkAndStatusWork(date + "-" + zeroI, "Виконано");
            for (int listOrders = 0; listOrders <= ordersDTO.size() - 1;
listOrders++) {
                ordersInfo[0] = ordersInfo[0] +
ordersDTO.get(listOrders).getNetProfit(); //дохід
                ordersInfo[1]++; //кількість замовлень
            }
        }
        if (transactionsInfoRepository.findByLogicDate(date + "-" + zeroI).size()
>= 1) {
            List<TransactionsInfoDTO> transactionsInfoDTOS =
transactionsInfoRepository.findByLogicDate(date + "-" + zeroI);
            for (int listTransactions = 0; listTransactions <=
transactionsInfoDTOS.size() - 1; listTransactions++) {
                ordersInfo[2] = ordersInfo[2] +
transactionsInfoDTOS.get(listTransactions).getMoney();
            }
        }
    }
    return ordersInfo;
}

public List<TransactionsInfoDTO> transactionsInfoDTOS(String date) {
    List<TransactionsInfoDTO> transactionsInfoDTO = null;
    String transL = null;
    int indexList = 0;

    for (int listener = 0; listener <= 31; listener++) {

        if (listener < 10) {
            transL = "0" + listener;
        } else {
            transL = String.valueOf(listener);
        }
    }
}

```

```

        if (transactionsInfoRepository.findByLogicDate(date + "-" +
transL).size() >= 1) {
            List<TransactionsInfoDTO> transactionsInfoDTOS =
transactionsInfoRepository.findByLogicDate(date + "-" + transL);
            if (transactionsInfoDTO == null) {
                transactionsInfoDTO = transactionsInfoDTOS;
                indexList = indexList + transactionsInfoDTOS.size();
            } else {
                for (int listTransactions = 0; listTransactions <=
transactionsInfoDTOS.size() - 1; listTransactions++) {
                    transactionsInfoDTO.add(indexList,
transactionsInfoDTOS.get(listTransactions));
                    indexList++;
                }
            }
        }
    }
    return transactionsInfoDTO;
}

public List<OrdersDTO> ordersDTOList(String date) {
    List<OrdersDTO> dtoList = null;
    String zeroI = null;
    int indexList = 0;

    for (int i = 0; i <= 31; i++) {

        if (i < 10) {
            zeroI = "0" + i;
        } else {
            zeroI = String.valueOf(i);
        }

        if (ordersRepository.findByDateWorkAndStatusWork(date + "-" + zeroI,
"Виконано").size() >= 1) {
            List<OrdersDTO> ordersDTO =
ordersRepository.findByDateWorkAndStatusWork(date + "-" + zeroI, "Виконано");
            if (dtoList == null) {
                dtoList = ordersDTO;
                indexList = indexList + ordersDTO.size();
            } else {
                for (int listOrders = 0; listOrders <= ordersDTO.size() - 1;
listOrders++) {
                    dtoList.add(indexList, ordersDTO.get(listOrders));
                    indexList++;
                }
            }
        }
    }
    return dtoList;
}

public boolean addCoast(TransactionsInfoDTO transactionsInfoDTO, String
usernameCreator) {
    if (transactionsInfoDTO.getType().equals("Виплата зарплати") ||
transactionsInfoDTO.getType().equals("Виплата премії") ||
transactionsInfoDTO.getType().equals("Виплата авансу") &&
transactionsInfoDTO.getUsername() != "" && transactionsInfoDTO.getDate() != "") {
        transactionsInfoDTO.setNumber(transactionsNumberCheck());
        transactionsInfoDTO.setInfo(transactionsInfoDTO.getType());
        transactionsInfoDTO.setFull_info(transactionsInfoDTO.getType() + " " +
transactionsInfoDTO.getUsername());
        transactionsInfoDTO.setUsernameCreator(usernameCreator);
    }
}

```

```

transactionsInfoDTO.setLogicDate(transactionsInfoDTO.date.substring(0,10));
    transactionsInfoDTO.setLink("window.location='/transactions/" +
transactionsInfoDTO.getNumber() + "';");
    transactionsInfoDTO.setType("Виплата");
    transactionsInfoRepository.save(transactionsInfoDTO);
    return true;
} else if (transactionsInfoDTO.getType() != null &&
transactionsInfoDTO.getDate() != null) {
    transactionsInfoDTO.setNumber(transactionsNumberCheck());
    transactionsInfoDTO.setUsernameCreator(usernameCreator);

transactionsInfoDTO.setLogicDate((transactionsInfoDTO.date.substring(0,10)));
    transactionsInfoDTO.setLink("window.location='/transactions/" +
transactionsInfoDTO.getNumber() + "';");
    transactionsInfoRepository.save(transactionsInfoDTO);
    return true;
} else { return false; }
}

private long transactionsNumberCheck() {
    long i;
    for (i = 1; transactionsInfoRepository.findByNumber(i) != null; i++) {
    }
    return i;
}
}

```

## B.12 Программный код класу HomeService:

```

package com.degree.cto.services;

import com.degree.cto.dtos.ReviewsDTO;
import com.degree.cto.dtos.UsersDTO;
import com.degree.cto.repositories.ReviewsRepository;
import com.degree.cto.repositories.UsersRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class HomeService {

    @Autowired
    private ReviewsRepository reviewsRepository;
    @Autowired
    private UsersRepository usersRepository;

    int[] logicId = new int[9];

    public List<UsersDTO> managementList() {
        List<UsersDTO> list = usersRepository.findByRole("Директор");
        list.addAll(usersRepository.findByRole("Менеджер"));
        list.addAll(usersRepository.findByRole("Бухгалтер"));
        list.addAll(usersRepository.findByRole("Рекрутер"));
        return list;
    }
}

```

```

public List<ReviewsDTO> reviewsGetList() {
    List<ReviewsDTO> reviewsList = reviewsRepository.findAll();
    reviewsGetLogicId(reviewsList);
    reviewsList.clear();
    for (int i = 0; i < 9; i++) {
        if (logicId[i] != 0) {
            reviewsList.add(reviewsRepository.findByLogicId(logicId[i]));
        }
    }
    return reviewsList;
}

private int[] reviewsGetLogicId(List<ReviewsDTO> list) {
    long sizeList = list.size();
    for (int i = 0; i < 9; i++) {
        int randomReviews = 1 + (int) (Math.random() * sizeList);
        if (randomReviews != logicId[0] && randomReviews != logicId[1] &&
randomReviews != logicId[2] && randomReviews != logicId[3] &&
            randomReviews != logicId[4] && randomReviews != logicId[5] &&
randomReviews != logicId[6] && randomReviews != logicId[7] &&
            randomReviews != logicId[8]) {
            logicId[i] = randomReviews;
        } else {
            if (sizeList >= 9) {
                i--;
            } else {
                logicId[i] = 0;
            }
        }
    }
    return logicId;
}
}

```

### V.13 Програмный код класса OrderService:

```

package com.degree.cto.services;

import com.degree.cto.dtos.OrdersCheckDTO;
import com.degree.cto.dtos.OrdersDTO;
import com.degree.cto.dtos.OrdersWorkTeamDTO;
import com.degree.cto.dtos.UsersDTO;
import com.degree.cto.repositorys.OrdersCheckRepository;
import com.degree.cto.repositorys.OrdersRepository;
import com.degree.cto.repositorys.OrdersWorkTeamRepository;
import com.degree.cto.repositorys.UsersRepository;
import org.bouncycastle.cms.PasswordRecipientId;
import org.bson.types.ObjectId;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.sql.Timestamp;

@Service
public class OrderService {

    @Autowired
    private OrdersRepository ordersRepository;
}

```

```

@Autowired
private UsersRepository usersRepository;

@Autowired
private OrdersWorkTeamRepository ordersWorkTeamRepository;

@Autowired
private OrdersCheckRepository ordersCheckRepository;

Timestamp timestamp = new Timestamp(System.currentTimeMillis());

public String orderCreate(OrdersDTO ordersDTO, String username) {

    if(orderValid(ordersDTO) == null) {
        orderCreateSave(ordersDTO);
        return null;
    } else {
        return orderValid(ordersDTO);
    }
}

private String orderValid(OrdersDTO ordersDTO) {
    return null;
}

private void orderCreateSave(OrdersDTO ordersDTO) {
    OrdersDTO ordersDTOLogic = new OrdersDTO();
    if (ordersDTO.getUsername() == null) {
        ordersDTOLogic =
ordersRepository.findByOrderNumber(ordersDTO.orderNumber);
        ordersDTO.setUsername(ordersDTOLogic.username);
    }
    if (ordersDTO.getOrderNumber() == 0) {
        ordersDTO.setOrderNumber(orderCreateNumber());
    } else {

ordersDTO.setId(ordersRepository.findByOrderNumber(ordersDTO.getOrderNumber()).id);
    }
    if (ordersDTO.getLink() == null) {
        ordersDTO.setLink("window.location='/order/'+ordersDTO.getOrderNumber() +
";");
    }
    if (ordersDTOLogic.getStatusWork() == null) {
        ordersDTO.setStatusWork("Опрацювання");
    } else {
        ordersDTO.setStatusWork(ordersDTOLogic.getStatusWork());
    }
    if (ordersDTO.getDateCreateOrder() == null) {
        ordersDTO.setDateCreateOrder(String.valueOf(timestamp));
    }
    if (ordersDTO.getPhoneCall() == null) {
        ordersDTO.setPhoneCall("Так");
    }
    if (ordersDTO.getArchive() == null) {
        ordersDTO.setArchive("Hi");
    }
    ordersRepository.save(ordersDTO);
}

private long orderCreateNumber() {
    long i;
    for(i = 1; ordersRepository.findByOrderNumber(i) != null; i++){ }
    return i;
}

```

```

    public void orderAddTeam(OrdersWorkTeamDTO ordersWorkTeamDTO, long numberOrder) {
        ordersWorkTeamDTO.setNumberOrder(numberOrder);
        UsersDTO usersDTO =
usersRepository.findByPersonalIndent(ordersWorkTeamDTO.username);
        ordersWorkTeamDTO.setStatusUser(usersDTO.getStatus());
        ordersWorkTeamDTO.setName(usersDTO.getName());
        ordersWorkTeamDTO.setLink("window.location='/@" +
usersDTO.getPersonalIndent() + "'");
        ordersWorkTeamRepository.save(ordersWorkTeamDTO);
    }

    public void orderDellTeam(String delDto, long numberOrder) {
        OrdersWorkTeamDTO ordersWorkTeamDTO =
ordersWorkTeamRepository.findByNumberOrderAndUsername(numberOrder, delDto);
        ordersWorkTeamRepository.delete(ordersWorkTeamDTO);
    }

    public void orderAddCheck(OrdersCheckDTO ordersCheckDTO, long numberOrder) {
        ordersCheckDTO.setNumberOrder(numberOrder);
        ordersCheckDTO.setNumberCheck(orderCreateNumberCheck());
        ordersCheckRepository.save(ordersCheckDTO);
        OrdersDTO ordersDTO = ordersRepository.findByOrderNumber(numberOrder);
        ordersDTO.setPrice(orderPrice(numberOrder));
        ordersRepository.save(ordersDTO);
    }

    private int orderPrice(long numberOrder) {
        long i;
        int price = 0;
        for(i = 1; ordersCheckRepository.findByNumberOrderAndNumberCheck(numberOrder,
i) != null; i++){
            OrdersCheckDTO ordersCheckDTO =
ordersCheckRepository.findByNumberCheck(i);
            price = price + ordersCheckDTO.getPrice();
        }

        return price;
    }

    private long orderCreateNumberCheck() {
        long i;
        for(i = 1; ordersCheckRepository.findByNumberCheck(i) != null; i++){ }
        return i;
    }

    public void orderDellCheck(long delDto, long orderNumber) {
        OrdersCheckDTO ordersCheckDTO =
ordersCheckRepository.findByNumberCheck(delDto);
        ordersCheckRepository.delete(ordersCheckDTO);
        OrdersDTO ordersDTO = ordersRepository.findByOrderNumber(orderNumber);
        ordersDTO.setPrice(orderPrice(orderNumber));
        ordersRepository.save(ordersDTO);
    }
}

```

## B.14 Программный код класу PersonalPageService:

```

package com.degree.cto.services;

import com.degree.cto.dtos.ReviewsDTO;
import com.degree.cto.repositorys.ReviewsRepository;
import com.degree.cto.repositorys.UsersRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.sql.Timestamp;

@Service
public class PersonalPageService {

    @Autowired
    private UsersRepository usersRepository;
    @Autowired
    private ReviewsRepository reviewsRepository;

    public String pageNotFond(String username) {
        if (usersRepository.findByPersonalIndent(username) == null) {
            return "redirect:/personal-page";
        } else {
            return "personal-page";
        }
    }

    public void createReviews(ReviewsDTO reviewsDTO, String username) {
        reviewsDTO.setTimestamp(String.valueOf(new
Timestamp(System.currentTimeMillis()).substring(0,10)));
        reviewsDTO.setUsername(username);

reviewsDTO.setUser_image(usersRepository.findByPersonalIndent(reviewsDTO.username).ge
tPhoto_url());
        reviewsDTO.setLogicId(createLogicId());
        reviewsRepository.save(reviewsDTO);
    }

    private long createLogicId() {
        long i;
        for(i = 1; reviewsRepository.findByLogicId(i) != null; i++){ }
        return i;
    }

    public String editAccess(String username, String urlPage) {
        if (username.equals(urlPage)) {
            return "true";
        } else {
            return "false";
        }
    }
}

```

## B.15 Программный код класу OrdersCheckDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;

```

```

import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter
@Setter
@Document(collection = "OrdersCheck")
public class OrdersCheckDTO {

    @Id
    public ObjectId id;
    public long numberOrder;
    public String name;
    public long numberCheck;
    public int price;
}

```

## B.16 Програмный код класу OrdersDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter
@Setter
@Document(collection = "Orders")
public class OrdersDTO {

    @Id
    public ObjectId id;
    public long orderNumber;
    public String username;
    public String brandCar;
    public String modelCar;
    public String yearsCar;
    public String service;
    public int price;
    public String dateCreateOrder;
    public String dateWork;
    public String phoneCall;
    public String message;
    public String statusWork;
    public String archive;
    public String link;
    public long netProfit;
}

```

## B.17 Програмный код класу OrderStatusDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter
@Setter
@Document(collection = "OrderStatus")
public class OrderStatusDTO {

    @Id
    public ObjectId id;
    public int numberStatus;
    public String status;
}

```

### В.18 Програмный код класу OrdersWorkTeamDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter
@Setter
@Document(collection = "OrdersWorkTeam")
public class OrdersWorkTeamDTO {

    @Id
    public ObjectId id;
    public long numberOrder;
    public String username;
    public String Name;
    public String statusUser;
    public String link;
}

```

### В.19 Програмный код класу PartnersDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter

```

```

@Setter
@Document(collection = "Partners")
public class PartnersDTO {
    @Id
    public ObjectId id;
    public String image_url;
    public String name;
    public String text;
}

```

## B.20 Программный код класу ReviewsDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

import java.sql.Timestamp;

@Getter
@Setter
@Document(collection = "Reviews")
public class ReviewsDTO {

    @Id
    public ObjectId id;
    public long logicId;
    public String text;
    public String timestamp;
    public String username;
    public String user_image;
}

```

## B.21 Программный код класу ServicesListDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter
@Setter
@Document(collection = "ServicesList")
public class ServicesListDTO {

    @Id
    public ObjectId id;
    public String name;
}

```

```

    public String text;
    public String price;
    public String image_url;
}

```

## V.22 Програмный код класу TransactionsInfoDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

import java.sql.Timestamp;

@Getter
@Setter
@Document(collection = "Transactions")
public class TransactionsInfoDTO {

    @Id
    public ObjectId id;
    public long number;
    public String username;
    public String type;
    public int clocks_work;
    public String info;
    public String full_info;
    public int money;
    public String usernameCreator;
    public String date;
    public String logicDate;
    public String link;

}

```

## V.23 Програмный код класу UsersDTO:

```

package com.degree.cto.dtos;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter
@Setter
@Document(collection = "Users")
public class UsersDTO {
    @Id
    public ObjectId id;
}

```

```

    public String name;
    public String personalIndent;
    public String phone;
    public String email;
    public String status;
    public String role;
    public String photo_url;
}

```

## B.24 Программный код класу FindLogDTO:

```

package com.degree.cto.logic.Log;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
public class FindLogDTO {
    public String date;
}

```

## B.25 Программный код класу LogDTO:

```

package com.degree.cto.logic.Log;

import lombok.Getter;
import lombok.Setter;
import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Getter
@Setter
@Document(collection = "Logs")
public class LogDTO {

    @Id
    public ObjectId objectId;
    public String type;
    public String name;
    public String title;
    public String message;
    public String date;
    public String logicDateDays;
}

```

## В.26 Програмный код класу OrderStatusDAO:

```

package com.degree.cto.logic;

import com.degree.cto.dtos.OrderStatusDTO;
import com.degree.cto.repositories.OrderStatusRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class OrderStatusDAO {
    @Autowired
    private OrderStatusRepository orderStatusRepository;

    public void CheckCollectionsOrderStatus() {

        if (orderStatusRepository.findAllByNumberStatus(1) == null) {
            OrderStatusDTO orderStatusDTO = new OrderStatusDTO();
            orderStatusDTO.setNumberStatus(1);
            orderStatusDTO.setStatus("Опрацювання");
            orderStatusRepository.save(orderStatusDTO);
        }
        if (orderStatusRepository.findAllByNumberStatus(2) == null) {
            OrderStatusDTO orderStatusDTO = new OrderStatusDTO();
            orderStatusDTO.setNumberStatus(2);
            orderStatusDTO.setStatus("Підтверджено");
            orderStatusRepository.save(orderStatusDTO);
        }
        if (orderStatusRepository.findAllByNumberStatus(3) == null) {
            OrderStatusDTO orderStatusDTO = new OrderStatusDTO();
            orderStatusDTO.setNumberStatus(3);
            orderStatusDTO.setStatus("Виконується");
            orderStatusRepository.save(orderStatusDTO);
        }
        if (orderStatusRepository.findAllByNumberStatus(4) == null) {
            OrderStatusDTO orderStatusDTO = new OrderStatusDTO();
            orderStatusDTO.setNumberStatus(4);
            orderStatusDTO.setStatus("Виконано");
            orderStatusRepository.save(orderStatusDTO);
        }
    }
}

```

## В.27 Програмный код класу BeanKeycloakConfiguration:

```

package com.degree.cto.security;

import org.keycloak.adapters.springboot.KeycloakSpringBootConfigResolver;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class BeanKeycloakConfiguration {
    @Bean
    public KeycloakSpringBootConfigResolver KeycloakConfigResolver() {

```

```

        return new KeycloakSpringBootConfigResolver();
    }
}

```

## B.28 Программный код класса SecurityConfig:

```

package com.degree.cto.security;

import org.keycloak.adapters.springsecurity.KeycloakSecurityComponents;
import
org.keycloak.adapters.springsecurity.authentication.KeycloakAuthenticationProvider;
import
org.keycloak.adapters.springsecurity.config.KeycloakWebSecurityConfigurerAdapter;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Authentication
ManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.core.authority.mapping.SimpleAuthorityMapper;
import org.springframework.security.core.session.SessionRegistryImpl;
import
org.springframework.security.web.authentication.session.RegisterSessionAuthentication
Strategy;
import
org.springframework.security.web.authentication.session.SessionAuthenticationStrategy
;

@Configuration
@EnableWebSecurity
@ComponentScan(basePackageClasses = KeycloakSecurityComponents.class)
public class SecurityConfig extends KeycloakWebSecurityConfigurerAdapter {

    @Autowired
    public void configureGlobal(
        AuthenticationManagerBuilder auth) throws Exception {

        KeycloakAuthenticationProvider keycloakAuthenticationProvider
            = keycloakAuthenticationProvider();
        keycloakAuthenticationProvider.setGrantedAuthoritiesMapper(
            new SimpleAuthorityMapper());
        auth.authenticationProvider(keycloakAuthenticationProvider);
    }

    @Bean
    @Override
    protected SessionAuthenticationStrategy sessionAuthenticationStrategy() {
        return new RegisterSessionAuthenticationStrategy(
            new SessionRegistryImpl());
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        super.configure(http);
        http

```

```

        .authorizeRequests()
        .antMatchers("/resources/**", "/static/**", "/assets/**").permitAll()
        .antMatchers("/").permitAll()
        .anyRequest().authenticated()
        .and()
        .logout()
        .permitAll();
    }
}

```

### V.29 Программний код класу CtoApplication:

```

package com.degree.cto;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class CtoApplication {
    public static void main(String[] args) {
        SpringApplication.run(CtoApplication.class, args);
    }
}

```

### V.30 Программний код інтерфейсу LogRepository:

```

package com.degree.cto.logic.Log;

import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface LogRepository extends MongoRepository<LogDTO, ObjectId> {
    List<LogDTO> findByLogicDateDays(String date);
}

```

### V.31 Программний код інтерфейсу OrdersCheckRepository:

```

package com.degree.cto.repositories;

import com.degree.cto.dtos.OrdersCheckDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;

```

```

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface OrdersCheckRepository extends MongoRepository<OrdersCheckDTO,
ObjectId> {
    List<OrdersCheckDTO> findAllByNumberOrder(long number);
    OrdersCheckDTO findByNumberCheck(long number);
    OrdersCheckDTO findByNumberOrder(long number);
    OrdersCheckDTO findByNumberOrderAndNumberCheck(long numberOrder, long
numberCheck);
}

```

### В.32 Програмний код інтерфейсу OrdersRepository:

```

package com.degree.cto.repositories;

import com.degree.cto.dtos.OrdersDTO;
import com.degree.cto.dtos.TransactionsInfoDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface OrdersRepository extends MongoRepository<OrdersDTO, ObjectId> {
    List<OrdersDTO> findAllByUsername(String username);
    OrdersDTO findByOrderNumber(long number);
    List<OrdersDTO> findByStatusWorkAndArchive(String status, String archive);
    List<OrdersDTO> findByOrderNumberAndStatusWorkAndArchive(long number, String
status, String archive);
    List<OrdersDTO> findByArchive(String archive);
    List<OrdersDTO> findByArchiveAndOrderNumber(String archive, long number);
    List<OrdersDTO> findByDateWorkAndStatusWork(String date, String status);
}

```

### В.33 Програмний код інтерфейсу OrderStatusRepository:

```

package com.degree.cto.repositories;

import com.degree.cto.dtos.OrderStatusDTO;
import com.degree.cto.services.OrderService;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface OrderStatusRepository extends MongoRepository<OrderStatusDTO,
ObjectId> {

```

```

    OrderStatusDTO findAllByNumberStatus(int number);
}

```

### **В.34 Програмний код інтерфейсу OrdersWorkTeamRepository:**

```

package com.degree.cto.repositories;

import com.degree.cto.dtos.OrdersWorkTeamDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface OrdersWorkTeamRepository extends MongoRepository<OrdersWorkTeamDTO,
ObjectId> {
    List<OrdersWorkTeamDTO> findAllByNumberOrder(long number);
    OrdersWorkTeamDTO findByNumberOrderAndUsername(long number, String username);
}

```

### **В.35 Програмний код інтерфейсу PartnersRepository:**

```

package com.degree.cto.repositories;

import com.degree.cto.dtos.PartnersDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface PartnersRepository extends MongoRepository<PartnersDTO, ObjectId> {
}

```

### **В.36 Програмний код інтерфейсу ReviewsRepository:**

```

package com.degree.cto.repositories;

import com.degree.cto.dtos.ReviewsDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface ReviewsRepository extends MongoRepository<ReviewsDTO, ObjectId> {
    ReviewsDTO findByLogicId(long LogicId);
}

```

### В.37 Програмный код интерфейсу ServicesListRepository:

```
package com.degree.cto.repositories;

import com.degree.cto.dtos.ServicesListDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;

public interface ServicesListRepository extends MongoRepository<ServicesListDTO,
ObjectId> {
}
```

### В.38 Програмный код интерфейсу TransactionsInfoRepository:

```
package com.degree.cto.repositories;

import com.degree.cto.dtos.TransactionsInfoDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;

import java.util.List;

public interface TransactionsInfoRepository extends
MongoRepository<TransactionsInfoDTO, ObjectId> {
    TransactionsInfoDTO findByNumber(long number);
    List<TransactionsInfoDTO> findByLogicDate(String date);
    List<TransactionsInfoDTO> findAllByUsername(String username);
}
```

### В.39 Програмный код интерфейсу UsersRepository:

```
package com.degree.cto.repositories;

import com.degree.cto.dtos.UsersDTO;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface UsersRepository extends MongoRepository<UsersDTO, ObjectId> {
    UsersDTO findByPersonalIndent(String personal_indent);
    List<UsersDTO> findByRole(String role);
    List<UsersDTO> findByStatus(String status);
    List<UsersDTO> findByPersonalIndentAndStatus(String personal_indent, String
status);
}
```

## V.40 Програмний код інтерфейсу CtoApplicationTests:

```

package com.degree.cto;

import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration;
import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.web.servlet.MockMvc;

import static
org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
import static
org.springframework.test.web.servlet.request.MockMvcRequestBuilders.post;
import static
org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

@SpringBootTest
@AutoConfigureMockMvc
class CtoApplicationTests {

    int[] statusCode = new int[]{200, 302, 403};

    @Autowired
    private MockMvc mockMvc;

    private void basicIntegrationsGetTest(String url, int statusPage) throws
Exception {
        mockMvc.perform(get(url)).andExpect(status().is(statusPage));
    }

    private void basicIntegrationsPostTest(String url, int statusPage) throws
Exception {
        mockMvc.perform(post(url)).andExpect(status().is(statusPage));
    }

    @Test
    void contextLoads() {
    }

    @Test
    void homePage() throws Exception {
        basicIntegrationsGetTest("/", statusCode[0]);
        basicIntegrationsGetTest("/logout", statusCode[1]);
    }

    @Test
    void accountantPage() throws Exception {
        basicIntegrationsGetTest("/accountant", statusCode[1]);
        basicIntegrationsPostTest("/accountant", statusCode[2]);
        basicIntegrationsPostTest("/accountant/addCoast", statusCode[2]);
    }

    @Test

```

```

void directorPage() throws Exception {
    basicIntegrationsGetTest("/director", statusCode[1]);
    basicIntegrationsPostTest("/director", statusCode[2]);
}

@Test
void orderPages() throws Exception {
    basicIntegrationsGetTest("/order/1", statusCode[1]);
    basicIntegrationsGetTest("/order/1/edit", statusCode[1]);
    basicIntegrationsPostTest("/order/1/edit", statusCode[2]);
    basicIntegrationsPostTest("/order/1/edit/team/add", statusCode[2]);
    basicIntegrationsPostTest("/order/1/edit/team/dell", statusCode[2]);
    basicIntegrationsPostTest("/order/1/edit/check/add", statusCode[2]);
    basicIntegrationsPostTest("/order/1/edit/check/dell", statusCode[2]);
    basicIntegrationsGetTest("/order/create", statusCode[1]);
    basicIntegrationsPostTest("/order/create", statusCode[2]);
    basicIntegrationsGetTest("/order/management", statusCode[1]);
    basicIntegrationsPostTest("/order/management/find", statusCode[2]);
    basicIntegrationsGetTest("/order/management/archive/add", statusCode[1]);
    basicIntegrationsGetTest("/order/management/archive/dell", statusCode[1]);
    basicIntegrationsGetTest("/order/management/order/dell", statusCode[1]);
    basicIntegrationsGetTest("/order/management/order/status/new",
statusCode[1]);
    basicIntegrationsGetTest("/order/management/order/status/confirmed",
statusCode[1]);
    basicIntegrationsGetTest("/order/management/order/status/work",
statusCode[1]);
    basicIntegrationsGetTest("/order/management/order/status/complete",
statusCode[1]);
    basicIntegrationsGetTest("/order/management/archive", statusCode[1]);
    basicIntegrationsGetTest("/order/management/archive", statusCode[1]);
    basicIntegrationsPostTest("/order/management/archive/find", statusCode[2]);
}

@Test
void personalPage() throws Exception {
    basicIntegrationsGetTest("/@_okhotnik_", statusCode[1]);
    basicIntegrationsPostTest("/@_okhotnik_/edit", statusCode[2]);
}

@Test
void teamManagementPages() throws Exception {
    basicIntegrationsGetTest("/team/management", statusCode[1]);
    basicIntegrationsPostTest("/team/management/find", statusCode[2]);
    basicIntegrationsPostTest("/team/management/add", statusCode[2]);
    basicIntegrationsGetTest("/team/management/dell", statusCode[1]);
}

@Test
void transactionsPage() throws Exception {
    basicIntegrationsGetTest("/transactions/1", statusCode[1]);
    basicIntegrationsGetTest("/transactions/1/dell", statusCode[1]);
    basicIntegrationsPostTest("/transactions/1/edit", statusCode[2]);
}
}

```

## V.41 Код розмітки HTML Accountant-page:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
  <title>service station</title>
  <meta name="description" content="Дипломний проект Охота В.В.">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
  <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
  <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
  <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
  <link rel="stylesheet" href="/assets/css/accoutant-page.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
  <link rel="stylesheet" href="/assets/css/Brands.css">
  <link rel="stylesheet" href="/assets/css/dialog-page.css">
  <link rel="stylesheet" href="/assets/css/director-page.css">
  <link rel="stylesheet" href="/assets/css/Features-Blue.css">
  <link rel="stylesheet" href="/assets/css/Features-Boxed.css">
  <link rel="stylesheet" href="/assets/css/Features-Clean.css">
  <link rel="stylesheet" href="/assets/css/Footer-Dark.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.css">
  <link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
  <link rel="stylesheet" href="/assets/css/Map-Clean.css">
  <link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
  <link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
  <link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
  <link rel="stylesheet" href="/assets/css/Projects-Clean.css">
  <link rel="stylesheet" href="/assets/css/styles.css">
  <link rel="stylesheet" href="/assets/css/Team-Clean.css">
  <link rel="stylesheet" href="/assets/css/Team-Grid.css">
  <link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button" style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color: #000000;"><i class="fas fa-car" style="font-size: 24px;color: rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse" id="navcol-1">

```

```

        <ul class="navbar-nav me-auto">
            <li class="nav-item"></li>
            <li class="nav-item"><a class="nav-link"
href="/order/create">Створити замовлення</a></li>
            <li class="nav-item dropdown" th:if="{TeamAccess == 'yes'}"><a
class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown"
href="#">Функції працівника&nbsp;</a>
                <div class="dropdown-menu">
                    <a class="dropdown-item" href="/order/management"
th:if="{TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
                    <a class="dropdown-item" href="/team/management"
th:if="{TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
                    <a class="dropdown-item" href="/accountant"
th:if="{TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
                    <a class="dropdown-item" href="/director"
th:if="{TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
                </li>
            </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
        </div>
    </div>
</nav>
    <section>
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <section class="highlight-clean">
                        <div class="container">
                            <div class="intro">
                                <h2 class="text-center" style="height:
32px;">Бухгалтерська</h2>
                            </div>
                        </div>
                    </section>
                    <p style="text-align: left;font-size:
18px;"><span>Статистика:&nbsp;</span><br><span th:text="{Date}"></span></p>
                    <div class="accordion" role="tablist" id="accordion-3">
                        <div class="accordion-item">
                            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-3 .item-1" aria-expanded="false" aria-controls="accordion-3 .item-
1">Змінити дату для статистики</button></h2>
                            <div class="accordion-collapse collapse item-1"
role="tabpanel" data-bs-parent="#accordion-3">
                                <div class="accordion-body">
                                    <form th:method="POST" th:action="@{/accountant}"
th:object="{infoDTO}">
                                        <p><span>Виберіть місяць:</span><input
class="form-control" type="month" th:value="{Date}" th:name="logicDate"></p>
                                        <p style="text-align: center;"><button
class="btn btn-primary" type="submit" style="background:
rgb(255,204,30);">Застосувати</button></p>
                                    </form>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <hr>
    </div>
<div class="container">

```

```

<div class="row">
  <div class="col-md-3">
    <div class="card shadow border-start-primary py-2">
      <div class="card-body">
        <div class="row align-items-center no-gutters">
          <div class="col me-2">
            <div class="text-uppercase text-primary fw-bold
text-xs mb-1" style="color: rgb(94,154,244);"><span style="color:
rgb(94,154,244);">Чистий прибуток</span></div>
            <div class="text-dark fw-bold h5 mb-0"><span
style="color: rgb(109,109,109);" th:text="{NetPrice}"></span></div>
          </div>
          <div class="col-auto"><i class="fas fa-money-check-
alt fa-2x text-gray-300" style="color: rgb(83,186,136);"></i></div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-md-3">
    <div class="card shadow border-start-primary py-2">
      <div class="card-body">
        <div class="row align-items-center no-gutters">
          <div class="col me-2">
            <div class="text-uppercase text-primary fw-bold
text-xs mb-1"><span style="color: rgb(94,154,244);">прибуток</span></div>
            <div class="text-dark fw-bold h5 mb-0"><span
style="color: rgb(109,109,109);" th:text="{Profit}"></span></div>
          </div>
          <div class="col-auto"><i class="fas fa-money-bill-alt
fa-2x text-gray-300" style="color: rgb(81,151,232);"></i></div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-md-3">
    <div class="card shadow border-start-primary py-2">
      <div class="card-body">
        <div class="row align-items-center no-gutters">
          <div class="col me-2">
            <div class="text-uppercase text-primary fw-bold
text-xs mb-1"><span style="color: rgb(94,154,244);">Витрати</span></div>
            <div class="text-dark fw-bold h5 mb-0"><span
style="color: rgb(109,109,109);" th:text="{Costs}"></span></div>
          </div>
          <div class="col-auto"><i class="fas fa-shopping-cart
fa-2x text-gray-300" style="color: rgb(119,108,241);"></i></div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-md-3">
    <div class="card shadow border-start-primary py-2">
      <div class="card-body">
        <div class="row align-items-center no-gutters">
          <div class="col me-2">
            <div class="text-uppercase text-primary fw-bold
text-xs mb-1"><span style="color: rgb(94,154,244);">Виконано замовлень</span></div>
            <div class="text-dark fw-bold h5 mb-0"><span
style="color: rgb(109,109,109);" th:text="{OrderSize}"></span></div>
          </div>
          <div class="col-auto"><i class="fas fa-book fa-2x
text-gray-300" style="color: rgb(52,205,196);"></i></div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
<hr>
</div>
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="accordion" role="tablist" id="accordion-1">
                <div class="accordion-item">
                    <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-1 .item-1" aria-expanded="false" aria-controls="accordion-1 .item-
1">Подати звіт за зарплату</button></h2>
                    <div class="accordion-collapse collapse item-1"
role="tabpanel" data-bs-parent="#accordion-1">
                        <div class="accordion-body">
                            <div>
                                <ul class="nav nav-tabs" role="tablist"
style="font-size: 17px;">
                                    <li class="nav-item"
role="presentation"><a class="nav-link active" role="tab" data-bs-toggle="tab"
href="#tab-1">Зарплата</a></li>
                                    <li class="nav-item"
role="presentation"><a class="nav-link" role="tab" data-bs-toggle="tab" href="#tab-
2">Премія</a></li>
                                    <li class="nav-item"
role="presentation"><a class="nav-link" role="tab" data-bs-toggle="tab" href="#tab-
3">Аванс</a></li>
                                </ul>
                                <div class="tab-content">
                                    <div class="tab-pane active"
role="tabpanel" id="tab-1">
                                        <form th:method="POST"
th:action="@{/accountant/addCoast}" th:object="{tDTO}">
                                            <input class="form-control"
type="hidden" th:name="type" th:value="{ 'Виплата зарплати' }">
                                            <p><span>Ідентифікатор
працівника:</span><input class="form-control" type="text" th:name="username"></p>
                                            <p><span>Кількість годин
праці:</span><input class="form-control" type="number" th:name="clocks_work"
th:value="0"></p>
                                            <p><span>Сума:</span><input
class="form-control" type="number" th:name="money" th:value="0"></p>
                                            <p><span>Дата:</span><input
class="form-control" type="datetime-local" th:name="date"></p>
                                            <p style="text-align:
center;"><button class="btn btn-primary" type="submit" style="background:
rgb(255,204,30);">Подати</button></p>
                                        </form>
                                    </div>
                                    <div class="tab-pane" role="tabpanel"
id="tab-2">
                                        <form th:method="POST"
th:action="@{/accountant/addCoast}" th:object="{tDTO}">
                                            <input class="form-control"
type="hidden" th:name="type" th:value="{ 'Виплата премії' }">
                                            <p><span>Ідентифікатор
працівника:</span><input class="form-control" type="text" th:name="username"></p>
                                            <p><span>Кількість годин
праці:</span><input class="form-control" type="number" th:name="clocks_work"
th:value="0"></p>
                                            <p><span>Сума:</span><input
class="form-control" type="number" th:name="money" th:value="0"></p>

```

```

                <p><span>Дата:</span><input
class="form-control" type="datetime-local" th:name="date"></p>
                <p style="text-align:
center;"><button class="btn btn-primary" type="submit" style="background:
rgb(255,204,30);">Подати</button></p>
                </form>
            </div>
            <div class="tab-pane" role="tabpanel"
id="tab-3">
                <form th:method="POST"
th:action="@{/accountant/addCoast}" th:object="{tDTO}">
                    <input class="form-control"
type="hidden" th:name="type" th:value="{ 'Виплата авансу' }">
                    <p><span>Ідентифікатор
працівника:</span><input class="form-control" type="text" th:name="username"></p>
                    <p><span>Кількість годин
праці:</span><input class="form-control" type="number" th:name="clocks_work"
th:value="0"></p>
                    <p><span>Сума:</span><input
class="form-control" type="number" th:name="money" th:value="0"></p>
                    <p><span>Дата:</span><input
class="form-control" type="datetime-local" th:name="date"></p>
                    <p style="text-align:
center;"><button class="btn btn-primary" type="submit" style="background:
rgb(255,204,30);">Подати</button></p>
                </form>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="col-md-6">
    <div class="accordion" role="tablist" id="accordion-2">
        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-2 .item-1" aria-expanded="false" aria-controls="accordion-2 .item-
1">Подати звіт за витрату</button></h2>
            <div class="accordion-collapse collapse item-1"
role="tabpanel" data-bs-parent="#accordion-2">
                <div class="accordion-body">
                    <form th:method="POST"
th:action="@{/accountant/addCoast}" th:object="{tDTO}">
                        <p><span>Призначення:</span><input
class="form-control" type="text" th:name="type"></p>
                        <fieldset></fieldset>
                        <p><span>Короткі деталі:</span><input
class="form-control" type="text" th:name="info"></p>
                        <p><span>Повна інформація:</span><input
class="form-control" type="text" th:name="full_info"></p>
                        <p><span>Сума:</span><input class="form-
control" type="number" th:name="money" th:value="0"></p>
                        <p><span>Дата:</span><input class="form-
control" type="datetime-local" th:name="date"></p>
                        <p style="text-align: center;"><button
class="btn btn-primary" type="submit" style="background:
rgb(255,204,30);">Подати</button></p>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>
    <span style="color: var(--bs-red);" th:text="{Error}" th:if="{Error !=
null}"></span></span></p>
    <hr>
</div>
</section>
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <section>
                <div class="card">
                    <div class="card-body">
                        <div>
                            <h4>Обіг коштів прибуток</h4>
                        </div>
                        <div>
                            <div class="table-responsive" id="accountant-page-
table">
                                <table class="table">
                                    <thead>
                                        <tr>
                                            <th style="width: 1%;">Призначення</th>
                                            <th style="width: 65%;">Деталі</th>
                                            <th style="width: 20%;">Сума</th>
                                            <th style="width: 20%;">Дата</th>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr th:each="elTListOrd:
${TransactionsOrders}" class="table-row-link" style="cursor: pointer; cursor: hand;"
th:attr="onclick=${elTListOrd.link}">
                                            <td th:text="{ 'Надання послуг' }"></td>
                                            <td th:text="{ 'Ремонт/діагностика №' +
elTListOrd.orderNumber }"><br></td>
                                            <td th:text="{ '+' +
elTListOrd.netProfit }"></td>
                                            <td
th:text="{ {elTListOrd.dateWork} }"></td>
                                        </tr>
                                    </tbody>
                                </table>
                            </div>
                        </div>
                    </div>
                </div>
            </section>
        </div>
        <div class="col-md-6">
            <section>
                <div class="card">
                    <div class="card-body">
                        <div>
                            <h4>Обіг коштів витрати</h4>
                        </div>
                        <div>
                            <div class="table-responsive" id="accountant-page-
table">
                                <table class="table">
                                    <thead>
                                        <tr>
                                            <th style="width:
1%;">Призначення</th>

```



```
</html>
```

## B.42 Код розмітки HTML Director-page:

```
<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
  <title>service station</title>
  <meta name="description" content="Дипломний проект Охота В.В.">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
  <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
  <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
  <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
  <link rel="stylesheet" href="/assets/css/accoutant-page.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
  <link rel="stylesheet" href="/assets/css/Brands.css">
  <link rel="stylesheet" href="/assets/css/dialog-page.css">
  <link rel="stylesheet" href="/assets/css/director-page.css">
  <link rel="stylesheet" href="/assets/css/Features-Blue.css">
  <link rel="stylesheet" href="/assets/css/Features-Boxed.css">
  <link rel="stylesheet" href="/assets/css/Features-Clean.css">
  <link rel="stylesheet" href="/assets/css/Footer-Dark.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.css">
  <link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
  <link rel="stylesheet" href="/assets/css/Map-Clean.css">
  <link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
  <link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
  <link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
  <link rel="stylesheet" href="/assets/css/Projects-Clean.css">
  <link rel="stylesheet" href="/assets/css/styles.css">
  <link rel="stylesheet" href="/assets/css/Team-Clean.css">
  <link rel="stylesheet" href="/assets/css/Team-Grid.css">
  <link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button" style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color: #000000;"><i class="fas fa-car" style="font-size: 24px;color:
```



```

class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-3 .item-1" aria-expanded="false" aria-controls="accordion-3 .item-
1">Фільтр</button></h2>
    <div class="accordion-collapse collapse item-1"
role="tabpanel" data-bs-parent="#accordion-3">
        <div class="accordion-body">
            <form th:method="POST"
th:action="@{/director}" th:object="${findLogDTO}">
                <p><span>День:</span><input
class="form-control" type="date" th:name="date"></p>
                <p style="text-align:
center;"><button class="btn btn-primary" type="submit" style="background:
rgb(255,204,30);">Застосувати</button></p>
            </form>
        </div>
    </div>
</div>
</div>
<div class="table-responsive director-page-system-
messages">
    <table class="table">
        <thead>
            <tr>
                <th style="width: 20%;">Заголовок</th>
                <th style="width: 65%;">Вміст</th>
                <th style="width: 20%;">Дата</th>
            </tr>
        </thead>
        <tbody>
            <tr th:each="elLogs : ${Logs}">
                <td th:text="${elLogs.title}"></td>
                <td th:text="${elLogs.message}"></td>
                <td th:text="${elLogs.date}"></td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</div>
</div>
</div>
<hr>
</div>
</section>
<footer class="footer-dark">
    <div class="container">
        <div class="row">
            <div class="col-sm-6 col-md-3 item">
                <h3>Навігація</h3>
                <ul>
                    <li><a href="/#Info-section">Переваги</a></li>
                    <li><a href="/#Service-section">Послуги</a></li>
                    <li><a href="/#Contacts-section">Контакти</a></li>
                    <li><a href="/#Team-section">Керівники</a></li>
                    <li><a href="/#Partners-section">Партнери</a></li>
                    <li><a href="/#Reviews-section">Відгуки</a></li>
                </ul>
            </div>
            <div class="col-md-6 item text">
                <h3>СТО</h3>
                <p th:utext="${'Такі машини, як ця, можуть бігати вічно. <br>При
належному догляді вони безсмертні. <br>— Домінік Торетто'}"></p>
            </div>
            <div class="col item social"><a

```

```

href="https://www.facebook.com/vadim.ochota" style="color: rgb(255,204,30);"><i
class="icon ion-social-facebook" style="color: rgb(255,204,30);"></i></a><a
href="https://www.instagram.com/_okhotnik_" style="color: rgb(255,204,30);"><i
class="icon ion-social-instagram" style="color: rgb(255,204,30);"></i></a></div>
</div>
<p class="copyright">CTO © 2022</p>
</div>
</footer>
<script src="/assets/js/jquery.min.js"></script>
<script src="/assets/bootstrap/js/bootstrap.min.js"></script>
<script src="/assets/js/bs-init.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.js"
></script>
<script src="/assets/js/Lightbox-Gallery.js"></script>
</body>

</html>

```

### В.43 Код розмітки HTML Home:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-
fit=no">
  <title>service station</title>
  <meta name="description" content="Дипломний проект Охота В.В.">
  <link rel="icon" type="image/png" sizes="512x512" href="assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="assets/img/icon.png">
  <link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="assets/fonts/fontawesome-all.min.css">
  <link rel="stylesheet" href="assets/fonts/font-awesome.min.css">
  <link rel="stylesheet" href="assets/fonts/ionicons.min.css">
  <link rel="stylesheet" href="assets/fonts/material-icons.min.css">
  <link rel="stylesheet" href="assets/fonts/fontawesome5-overrides.min.css">
  <link rel="stylesheet" href="assets/css/accountant-page.css">
  <link rel="stylesheet" href="assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">
  <link rel="stylesheet" href="assets/css/Bootstrap-Calendar.css">
  <link rel="stylesheet" href="assets/css/Brands.css">
  <link rel="stylesheet" href="assets/css/dialog-page.css">
  <link rel="stylesheet" href="assets/css/director-page.css">
  <link rel="stylesheet" href="assets/css/Features-Blue.css">
  <link rel="stylesheet" href="assets/css/Features-Boxed.css">
  <link rel="stylesheet" href="assets/css/Features-Clean.css">
  <link rel="stylesheet" href="assets/css/Footer-Dark.css">
  <link rel="stylesheet" href="assets/css/Highlight-Blue.css">
  <link rel="stylesheet" href="assets/css/Highlight-Clean.css">
  <link rel="stylesheet" href="assets/css/Highlight-Phone.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.cs
s">
  <link rel="stylesheet" href="assets/css/Lightbox-Gallery.css">
  <link rel="stylesheet" href="assets/css/Map-Clean.css">

```

```

<link rel="stylesheet" href="assets/css/Modern-Contact-Form.css">
<link rel="stylesheet" href="assets/css/Navigation-with-Button.css">
<link rel="stylesheet" href="assets/css/Newsletter-Subscription-Form.css">
<link rel="stylesheet" href="assets/css/Projects-Clean.css">
<link rel="stylesheet" href="assets/css/styles.css">
<link rel="stylesheet" href="assets/css/Team-Clean.css">
<link rel="stylesheet" href="assets/css/Team-Grid.css">
<link rel="stylesheet" href="assets/css/Testimonials.css">
</head>

<body style="background: var(--bs-dark);">
  <nav class="navbar navbar-light navbar-expand-lg navigation-clean-button"
  style="background: rgb(238,244,247);">
    <div class="container"><i class="fas fa-car" style="font-size: 42px;color:
  rgb(255,204,30);"></i><a class="navbar-brand" href="/" style="color:
  #000000;">СТО</a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-
  target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span
  class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse" id="navcol-1">
      <ul class="navbar-nav me-auto">
        <li class="nav-item"><a class="nav-link" href="/#Info-section"
  style="border-color: var(--bs-body-color);filter: blur(0px);">Переваги</a></li>
        <li class="nav-item"><a class="nav-link" href="/#Service-section"
  style="border-color: var(--bs-body-color);filter: blur(0px);">Послуги</a></li>
        <li class="nav-item"><a class="nav-link" href="/#Contacts-
  section" style="border-color: var(--bs-body-color);filter:
  blur(0px);">Контакти</a></li>
        <li class="nav-item"><a class="nav-link" href="/#Team-section"
  style="border-color: var(--bs-body-color);filter: blur(0px);">Керівники</a></li>
        <li class="nav-item"><a class="nav-link" href="/#Partners-
  section" style="border-color: var(--bs-body-color);filter:
  blur(0px);">Партнери</a></li>
        <li class="nav-item"><a class="nav-link" href="/#Reviews-section"
  style="border-color: var(--bs-body-color);filter: blur(0px);">Відгуки</a></li>
      </ul><span class="navbar-text actions" style="color: var(--bs-blue);"
  th:if="{Auth == 'no'}"> <a id="navbar-register" class="login" href="#"></a><a
  class="btn btn-light action-button" role="button" id="navbar-spun-buton"
  href="/personal-page">Реєстрація/Вхід</a></span>
      <span class="navbar-text actions" style="color: var(--bs-blue);"
  th:if="{Auth == 'yes'}"> <a class="btn btn-light action-button" role="button"
  id="navbar-spun-buton" href="/personal-page">Особистий кабінет</a></span>
    </div>
  </div>
  <div class="carousel slide" data-bs-ride="carousel" id="carousel-1"
  style="height: 700px;">
    <div class="carousel-inner" style="height: 700px;">
      <div class="carousel-item active" style="filter: blur(0px) sepia(0%);">
        <h1 class="text-center d-block" id="header-slider-text"
  style="color: rgb(255,255,255);"></h1>
      </div>
    </div>
  </div>
  <section style="margin-top: 12px;">
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <div class="accordion" role="tablist" id="accordion-1"></div>
        </div>
      </div>
    </div>
  </section>
  <section id="Info-section" class="features-blue" style="background: var(--bs-

```

```

dark);">
    <div class="container">
        <div class="intro">
            <h2 class="text-center">Переваги</h2>
            <p class="text-center">Робимо на совість!</p>
        </div>
        <div class="row features">
            <div class="col-sm-6 col-md-4 item"><i class="fa fa-map-marker
icon"></i>
                <h3 class="name">Розташування</h3>
                <p class="description">Сервіс знаходиться у центральній частині
міста та має комфортні заїзди на територію та парковки.</p>
            </div>
            <div class="col-sm-6 col-md-4 item"><i class="fa fa-clock-o
icon"></i>
                <h3 class="name">Швидко</h3>
                <p class="description">Виконуємо роботу без зайвого. Коротко і по
суті.</p>
            </div>
            <div class="col-sm-6 col-md-4 item"><i class="fa fa-list-alt icon"
style="color: rgb(255, 204, 30);"></i>
                <h3 class="name">Зручне бронювання&nbsp;</h3>
                <p class="description">Запис можливий на сайті, по телефону чи
месенджерах.</p>
            </div>
            <div class="col-sm-6 col-md-4 item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em" fill="currentColor"
viewBox="0 0 16 16" class="bi bi-people-fill icon">
                <path d="M7 14s-1 0-1-1 1-4 5-4 5 3 5 4-1 1-1 1H7zm4-6a3 3 0
1 0 0-6 3 3 0 0 0 0 6z"></path>
                <path fill-rule="evenodd" d="M5.216 14A2.238 2.238 0 0 1 5
13c0-1.355.68-2.75 1.936-3.72A6.325 6.325 0 0 0 5 9c-4 0-5 3-5 4s1 1 1
1h4.216z"></path>
                <path d="M4.5 8a2.5 2.5 0 1 0 0-5 2.5 2.5 0 0 0 0 5z"></path>
            </svg>
                <h3 class="name">Індивідуальний підхід</h3>
                <p class="description">Кожен клієнт має бути задоволений від
послуг і це основне.</p>
            </div>
            <div class="col-sm-6 col-md-4 item"><i class="fa fa-dollar icon"></i>
                <h3 class="name">Конкурентні ціни</h3>
                <p class="description">Також доступні спеціальні знижки для наших
клієнтів ;)</p>
            </div>
            <div class="col-sm-6 col-md-4 item"><i class="material-icons
icon">settings</i>
                <h3 class="name">Якісні запчастини</h3>
                <p class="description">Використовуємо тільки оригінальні
запчастини або тільки надійних виробників, які себе зарекомендували</p>
            </div>
        </div>
    </div>
</section>
<section class="photo-gallery" style="background: rgb(238,244,247);">
    <div class="container">
        <div class="intro">
            <h2 class="text-center" id="Service-section">Послуги</h2>
            <p class="text-center">Якщо не знайшли потрібного, чекаємо вашого
звернення у формі для зв'язку.</p>
        </div>
    </div>
    <div class="container">
        <div class="row">
            <div class="col-md-3" th:each="elServ : ${Services}">

```

```

        <div class="card" style="margin-bottom: 10px;">
            <div class="card-body card-product-home" style="cursor:
pointer; cursor: hand;" onclick="window.location='/order/create';">
                <h1 style="font-size: 22px;color: rgb(255,204,30);"
th:text="{elServ.name}"></h1>
                <p th:text="{elServ.text}"></p>
                <p style="font-size: 14px;text-align: right;"
th:text="{elServ.price}"></p>
            </div>
        </div>
    </div>
</div>
</div>
<div class="col">
    <section id="Contacts-section" class="highlight-clean" style="padding:
7px 0px;background: var(--bs-dark);">
        <div class="container">
            <div class="intro">
                <h2 class="text-center" style="color: var(--bs-body-
bg);">Контакти</h2>
            </div>
        </div>
    </section>
</div>
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <section class="highlight-clean">
                <div class="container" style="background: var(--bs-dark);">
                    <div class="intro">
                        <h2 class="text-center" style="height: 41px;font-size:
21px;color: var(--bs-body-bg);">Зв'язок</h2>
                        <p class="text-start"><i class="fa fa-phone"
style="color: rgb(255,204,30);"></i>&nbsp; +380 68 379 30 60</p>
                        <p class="text-start"><i class="icon ion-email"
style="color: rgb(255,204,30);"></i>&nbsp; vadim2000ohota@gmail.com</p>
                        <p class="text-start"><i class="fa fa-map" style="color:
rgb(255,204,30);"></i>&nbsp; Хмельницький вул. Інституцька</p>
                    </div>
                    <div class="buttons"><a class="btn btn-primary" role="button"
href="/order/create" style="background: rgb(255,204,30);">Залишити заяву
онлайн<br></a><a class="btn btn-primary" role="button" href="/personal-page"
style="background: rgb(255,204,30);">Ввійти в особистий кабінет<br></a></div>
                </div>
            </section>
        </div>
        <div class="col-md-6">
            <section class="map-clean" style="background: var(--bs-
dark);"><iframe allowfullscreen="" frameborder="0"
src="https://www.google.com/maps/embed/v1/place?key=AIzaSyDPuPTVyF7pG7N4btlvI3e96w7r9
6mkQ7U&amp;q=%D0%A5%D0%BC%D0%B5%D0%BB%D1%8C%D0%BD%D0%B8%D1%86%D1%8C%D0%BA%D0%B8%D0%B9
+%D0%9D%D0%B0%D1%86%D1%96%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9+%D0%A3%D0%B
D%D1%96%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%82%D0%B5%D1%82&amp;zoom=12" width="100%"
height="450" style="background: var(--bs-dark);"></iframe></section>
            </div>
        </div>
    </div>
</div>
<section class="team-clean" style="background: rgb(255, 255, 255);">
    <div class="container">
        <div id="Team-section" class="intro">
            <h2 class="text-center">Керівники</h2>
            <p class="text-center">Для нас робота - це хобі<br><br></p>

```



```

        <li><a href="/#Partners-section">Партнери</a></li>
        <li><a href="/#Reviews-section">Відгуки</a></li>
    </ul>
</div>
<div class="col-md-6 item text">
    <h3>СТО</h3>
    <p th:utext="{ 'Такі машини, як ця, можуть бігати вічно. <br>При належному догляді вони безсмертні. <br> – Домінік Торетто'}"></p>
</div>
<div class="col item social"><a
href="https://www.facebook.com/vadim.ochota" style="color: rgb(255,204,30);"><i
class="icon ion-social-facebook" style="color: rgb(255,204,30);"></i></a><a
href="https://www.instagram.com/_okhotnik_" style="color: rgb(255,204,30);"><i
class="icon ion-social-instagram" style="color: rgb(255,204,30);"></i></a></div>
</div>
<p class="copyright">СТО © 2022</p>
</div>
</footer>
<script src="assets/js/jquery.min.js"></script>
<script src="assets/bootstrap/js/bootstrap.min.js"></script>
<script src="assets/js/bs-init.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.js"
></script>
<script src="assets/js/Lightbox-Gallery.js"></script>
</body>
</html>

```

## B.44 Код розмітки HTML Order:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>service station</title>
    <meta name="description" content="Дипломний проект Охота В.В.">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
    <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
    <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
    <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
    <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
    <link rel="stylesheet" href="/assets/css/accoutant-page.css">
    <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">
    <link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
    <link rel="stylesheet" href="/assets/css/Brands.css">
    <link rel="stylesheet" href="/assets/css/dialog-page.css">
    <link rel="stylesheet" href="/assets/css/director-page.css">
    <link rel="stylesheet" href="/assets/css/Features-Blue.css">

```

```

<link rel="stylesheet" href="/assets/css/Features-Boxed.css">
<link rel="stylesheet" href="/assets/css/Features-Clean.css">
<link rel="stylesheet" href="/assets/css/Footer-Dark.css">
<link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
<link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
<link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.cs
s">
<link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
<link rel="stylesheet" href="/assets/css/Map-Clean.css">
<link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
<link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
<link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
<link rel="stylesheet" href="/assets/css/Projects-Clean.css">
<link rel="stylesheet" href="/assets/css/styles.css">
<link rel="stylesheet" href="/assets/css/Team-Clean.css">
<link rel="stylesheet" href="/assets/css/Team-Grid.css">
<link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button"
style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color:
#000000;"><i class="fas fa-car" style="font-size: 24px;color:
rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse"
class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-
hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
  <div class="collapse navbar-collapse" id="navcol-1">
    <ul class="navbar-nav me-auto">
      <li class="nav-item"></li>
      <li class="nav-item"><a class="nav-link"
href="/order/create">Створити замовлення</a></li>
      <li class="nav-item dropdown" th:if="${TeamAccess == 'yes'}"><a
class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown"
href="#">Функції працівника&nbsp;</a>
        <div class="dropdown-menu">
          <a class="dropdown-item" href="/order/management"
th:if="${TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
          <a class="dropdown-item" href="/team/management"
th:if="${TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
          <a class="dropdown-item" href="/accountant"
th:if="${TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
          <a class="dropdown-item" href="/director"
th:if="${TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
        </li>
      </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
    </div>
  </div>
</nav>
<section>
  <div class="container">
    <div class="row">
      <div class="col-md-12" style="text-align: center;">
        <hr>
        <p><span style="font-size: 23px;">Замовлення №&nbsp;</span><span
style="font-size: 23px;" th:text="${NumberOrder}"></span></p>
      </div>
    </div>
  </div>
</section>

```



```

        <thead>
            <tr>
                <th style="width: 80%;">Запчастина,
послуга</th>
                <th style="width: 20%;">Ціна</th>
            </tr>
        </thead>
        <tbody>
            <tr th:each="elOrdersCheckDTOList :
${OrdersCheckDTOList}">
                <td
th:text="{elOrdersCheckDTOList.name}"></td>
                <td
th:text="{elOrdersCheckDTOList.price}"></td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</div>
</div>
</div>
</div>
<hr>
</section>
<footer class="footer-dark">
    <div class="container">
        <div class="row">
            <div class="col-sm-6 col-md-3 item">
                <h3>Навігація</h3>
                <ul>
                    <li><a href="#Info-section">Переваги</a></li>
                    <li><a href="#Service-section">Послуги</a></li>
                    <li><a href="#Contacts-section">Контакти</a></li>
                    <li><a href="#Team-section">Керівники</a></li>
                    <li><a href="#Partners-section">Партнери</a></li>
                    <li><a href="#Reviews-section">Відгуки</a></li>
                </ul>
            </div>
            <div class="col-md-6 item text">
                <h3>СТО</h3>
                <p th:utext="{ '{Такі машини, як ця, можуть бігати вічно. <br>При
належному догляді вони безсмертні. <br> – Домінік Торетто'}'"></p>
            </div>
            <div class="col item social"><a
href="https://www.facebook.com/vadim.ochota" style="color: rgb(255,204,30);"><i
class="icon ion-social-facebook" style="color: rgb(255,204,30);"></i></a><a
href="https://www.instagram.com/_okhotnik_" style="color: rgb(255,204,30);"><i
class="icon ion-social-instagram" style="color: rgb(255,204,30);"></i></a></div>
        </div>
        <p class="copyright">СТО © 2022</p>
    </div>
</footer>
<script src="/assets/js/jquery.min.js"></script>
<script src="/assets/bootstrap/js/bootstrap.min.js"></script>
<script src="/assets/js/bs-init.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.js"
></script>
<script src="/assets/js/Lightbox-Gallery.js"></script>
</body>
</html>

```

## B.45 Код розмітки HTML Oder-archive:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
  <title>service station</title>
  <meta name="description" content="Дипломний проект Охота В.В.">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
  <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
  <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
  <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
  <link rel="stylesheet" href="/assets/css/accoutant-page.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
  <link rel="stylesheet" href="/assets/css/Brands.css">
  <link rel="stylesheet" href="/assets/css/dialog-page.css">
  <link rel="stylesheet" href="/assets/css/director-page.css">
  <link rel="stylesheet" href="/assets/css/Features-Blue.css">
  <link rel="stylesheet" href="/assets/css/Features-Boxed.css">
  <link rel="stylesheet" href="/assets/css/Features-Clean.css">
  <link rel="stylesheet" href="/assets/css/Footer-Dark.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.css">
  <link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
  <link rel="stylesheet" href="/assets/css/Map-Clean.css">
  <link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
  <link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
  <link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
  <link rel="stylesheet" href="/assets/css/Projects-Clean.css">
  <link rel="stylesheet" href="/assets/css/styles.css">
  <link rel="stylesheet" href="/assets/css/Team-Clean.css">
  <link rel="stylesheet" href="/assets/css/Team-Grid.css">
  <link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button" style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color: #000000;"><i class="fas fa-car" style="font-size: 24px;color: rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse" id="navcol-1">

```

```

        <ul class="navbar-nav me-auto">
            <li class="nav-item"></li>
            <li class="nav-item"><a class="nav-link"
href="/order/create">Створити замовлення</a></li>
            <li class="nav-item dropdown" th:if="{TeamAccess == 'yes'}"><a
class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown"
href="#">Функції працівника&nbsp;</a>
                <div class="dropdown-menu">
                    <a class="dropdown-item" href="/order/management"
th:if="{TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
                    <a class="dropdown-item" href="/team/management"
th:if="{TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
                    <a class="dropdown-item" href="/accountant"
th:if="{TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
                    <a class="dropdown-item" href="/director"
th:if="{TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
                </li>
        </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
    </div>
</div>
</nav>
<section>
    <div class="container">
        <div class="row">
            <div class="col-md-12" style="text-align: center;">
                <hr>
                <p><span style="font-size: 23px;">Замовлення №&nbsp;</span><span
style="font-size: 23px;" th:text="{NumberOrder}"></span></p>
            </div>
            <div class="row">
                <div class="col-md-6">
                    <h1 style="color: rgb(0,0,0);font-size: 19px;">Інформація</h1>
                    <p><span>Замовник: &nbsp;</span><span
th:text="{UserName}"></span></p>
                    <p><span>Ідентифікатор: &nbsp;</span><span
th:text="{UserIndent}"></span></p>
                    <p><span>Номер телефону: &nbsp;</span><span
th:text="{UserPhoneNumber}"></span></p>
                    <p><span>Електрона пошта: &nbsp;</span><span
th:text="{UserEmail}"></span></p>
                    <p><span>Марка авто: &nbsp;</span><span
th:text="{BrandCar}"></span></p>
                    <p><span>Модель авто: &nbsp;</span><span
th:text="{ModelCar}"></span></p>
                    <p><span>Рік авто: &nbsp;</span><span
th:text="{YearsCar}"></span></p>
                    <p><span>Тип роботи: &nbsp;</span><span
th:text="{Service}"></span></p>
                    <p><span>Ціна: &nbsp;</span><span th:text="{Price}"></span></p>
                    <p><span>Дата створення заяви: &nbsp;</span><span
th:text="{DateCreateOrder}"></span></p>
                    <p><span>Дата виконання: &nbsp;</span><span
th:text="{DateWork}"></span></p>
                    <p><span>Телефонувати: &nbsp;</span><span
th:text="{PhoneCall}"></span></p>
                    <p><span>Коментар: &nbsp;</span><span
th:text="{Message}"></span></p>
                    <p><span>Статус: &nbsp;</span><span style="color: #1aa665;"
th:text="{StatusWork}"></span></p>
                </div>

```



```

        <ul>
            <li><a href="/#Info-section">Переваги</a></li>
            <li><a href="/#Service-section">Послуги</a></li>
            <li><a href="/#Contacts-section">Контакти</a></li>
            <li><a href="/#Team-section">Керівники</a></li>
            <li><a href="/#Partners-section">Партнери</a></li>
            <li><a href="/#Reviews-section">Відгуки</a></li>
        </ul>
    </div>
    <div class="col-md-6 item text">
        <h3>СТО</h3>
        <p th:utext="{ '${'Такі машини, як ця, можуть бігати вічно. <br>При належному догляді вони безсмертні. <br> – Домінік Торетто'}" ></p>
    </div>
    <div class="col item social"><a
href="https://www.facebook.com/vadim.ochota" style="color: rgb(255,204,30);" ><i
class="icon ion-social-facebook" style="color: rgb(255,204,30);" ></i></a><a
href="https://www.instagram.com/_okhotnik_" style="color: rgb(255,204,30);" ><i
class="icon ion-social-instagram" style="color: rgb(255,204,30);" ></i></a></div>
    </div>
    <p class="copyright">СТО © 2022</p>
</div>
</footer>
<script src="/assets/js/jquery.min.js"></script>
<script src="/assets/bootstrap/js/bootstrap.min.js"></script>
<script src="/assets/js/bs-init.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.js"
></script>
    <script src="/assets/js/Lightbox-Gallery.js"></script>
</body>
</html>

```

## В.46 Код розмітки HTML Oder-create:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>service station</title>
    <meta name="description" content="Дипломний проект Охота В.В.">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
    <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
    <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
    <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
    <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
    <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
    <link rel="stylesheet" href="/assets/css/accoutant-page.css">
    <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">

```

```

<link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
<link rel="stylesheet" href="/assets/css/Brands.css">
<link rel="stylesheet" href="/assets/css/dialog-page.css">
<link rel="stylesheet" href="/assets/css/director-page.css">
<link rel="stylesheet" href="/assets/css/Features-Blue.css">
<link rel="stylesheet" href="/assets/css/Features-Boxed.css">
<link rel="stylesheet" href="/assets/css/Features-Clean.css">
<link rel="stylesheet" href="/assets/css/Footer-Dark.css">
<link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
<link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
<link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.css">
<link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
<link rel="stylesheet" href="/assets/css/Map-Clean.css">
<link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
<link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
<link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
<link rel="stylesheet" href="/assets/css/Projects-Clean.css">
<link rel="stylesheet" href="/assets/css/styles.css">
<link rel="stylesheet" href="/assets/css/Team-Clean.css">
<link rel="stylesheet" href="/assets/css/Team-Grid.css">
<link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button" style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color: #000000;"><i class="fas fa-car" style="font-size: 24px;color: rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse" id="navcol-1">
      <ul class="navbar-nav me-auto">
        <li class="nav-item"></li>
        <li class="nav-item"><a class="nav-link" href="/order/create">Створити замовлення</a></li>
        <li class="nav-item dropdown" th:if="${TeamAccess == 'yes'}"><a class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown" href="#">Функції працівника&nbsp;</a>
          <div class="dropdown-menu">
            <a class="dropdown-item" href="/order/management" th:if="${TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
            <a class="dropdown-item" href="/team/management" th:if="${TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
            <a class="dropdown-item" href="/accountant" th:if="${TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
            <a class="dropdown-item" href="/director" th:if="${TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
        </li>
      </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий кабінет</a></span>
    </div>
  </div>
</nav>
<section>
  <hr>
  <form th:method="POST" th:action="@{/order/create}" th:object="${ordersDTO}">
    <h1 style="color: #000000;font-size: 28px;text-align: center;">Створення замовлення</h1>

```



```

    </div>
</footer>
<script src="/assets/js/jquery.min.js"></script>
<script src="/assets/bootstrap/js/bootstrap.min.js"></script>
<script src="/assets/js/bs-init.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.js"
></script>
<script src="/assets/js/Lightbox-Gallery.js"></script>
</body>

</html>

```

## B.47 Код розмітки HTML Oder-edit:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-
fit=no">
  <title>service station</title>
  <meta name="description" content="Дипломний проект Охота В.В.">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
  <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
  <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
  <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
  <link rel="stylesheet" href="/assets/css/accoutant-page.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-
Code.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
  <link rel="stylesheet" href="/assets/css/Brands.css">
  <link rel="stylesheet" href="/assets/css/dialog-page.css">
  <link rel="stylesheet" href="/assets/css/director-page.css">
  <link rel="stylesheet" href="/assets/css/Features-Blue.css">
  <link rel="stylesheet" href="/assets/css/Features-Boxed.css">
  <link rel="stylesheet" href="/assets/css/Features-Clean.css">
  <link rel="stylesheet" href="/assets/css/Footer-Dark.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.cs
s">
  <link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
  <link rel="stylesheet" href="/assets/css/Map-Clean.css">
  <link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
  <link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
  <link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
  <link rel="stylesheet" href="/assets/css/Projects-Clean.css">
  <link rel="stylesheet" href="/assets/css/styles.css">

```

```

    <link rel="stylesheet" href="/assets/css/Team-Clean.css">
    <link rel="stylesheet" href="/assets/css/Team-Grid.css">
    <link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button"
style="background: rgb(238,244,247);">
    <div class="container"><a class="navbar-brand" href="/" style="color:
#000000;"><i class="fas fa-car" style="font-size: 24px;color:
rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse"
class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-
hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
        <div class="collapse navbar-collapse" id="navcol-1">
            <ul class="navbar-nav me-auto">
                <li class="nav-item"></li>
                <li class="nav-item"><a class="nav-link"
href="/order/create">Створити замовлення</a></li>
                <li class="nav-item dropdown" th:if="${TeamAccess == 'yes'}"><a
class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown"
href="#">Функції працівника&nbsp;</a>
                    <div class="dropdown-menu">
                        <a class="dropdown-item" href="/order/management"
th:if="${TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
                        <a class="dropdown-item" href="/team/management"
th:if="${TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
                        <a class="dropdown-item" href="/accountant"
th:if="${TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
                        <a class="dropdown-item" href="/director"
th:if="${TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
                    </li>
                </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
            </div>
        </div>
</nav>
    <div class="container">
        <hr>
        <div class="row">
            <div class="col-md-12" style="text-align: center;">
                <p id="order-edit-page"><span style="font-size:
23px;">Редагування</span></p>
                <p><span style="font-size: 23px;">Замовлення №&nbsp;</span><span
style="font-size: 23px;" th:text="${OrderNumber}"></span></p>
            </div>
        </div>
        <div class="row">
            <div class="col-md-6">
                <form th:method="POST" th:action="${editSaveLink}"
th:object="${ordersDTOEdit}">
                    <h1 style="color: rgb(0,0,0);font-size: 19px;">Інформація</h1>
                    <input class="form-control" type="hidden" name="orderNumber"
th:value="${OrderNumber}">
                    <p><span>Марка:&nbsp;</span><input class="form-control" type="text"
th:value="${BrandCar}" th:name="brandCar"></span></p>
                    <p><span>Модель:&nbsp;</span><input th:value="${ModelCar}" class="form-
control" type="text" th:name="modelCar"></span></p>
                    <p><span>Рік авто:&nbsp;</span><input th:value="${YearsCar}"
class="form-control" type="month" th:name="yearsCar"></span></p>
                    <p><span>Тип роботи:&nbsp;</span><input th:value="${Service}"
class="form-control" type="text" th:name="service"></span></p>
                    <p><span>Ціна:&nbsp;</span><input th:value="${Price}" class="form-

```

```

control" type="number" th:name="price"></span></p>
    <p><span>Дата виконання:&nbsp;<input th:value="{DateWork}"
class="form-control" type="date" th:name="dateWork"></span></p>
    <p><span>Телефонувати:&nbsp;</span><span>&nbsp;<input
type="checkbox" th:name="phoneCall" value="Hi" th:checked="{PhoneCall ==
'Hi'}"></span><span>&nbsp;Hi&nbsp;</span></p>
    <p><span>Коментар:&nbsp;<input th:value="{Message}" class="form-
control" type="text" th:name="message"></textarea></span></p>
    <p><span>Архів:&nbsp;</span><span>&nbsp;<input type="checkbox"
th:checked="{Archive == 'Так'}" th:value="Так"
th:name="archive"></span><span>&nbsp;Архівувати</span></p>
    <p><span>Чистий прибуток:&nbsp;<input th:value="{NetProfit}"
class="form-control" type="number" th:name="netProfit"></span></p>
    <p style="text-align: center;"><button class="btn btn-primary"
type="submit" style="background: rgb(255,204,30);">Зберегти</button></p>
    </form>
  </div>
  <div class="col-md-6">
    <div>
      <h1 style="color: rgb(0,0,0);font-size: 19px;">Працівники які
виконували замовлення<br></h1>
      <div class="table-responsive">
        <table class="table">
          <thead>
            <tr>
              <th style="width: 60%;"><strong>Ім'я та
призвище</strong><br></th>
              <th style="width:
20%;"><strong>Посада</strong><br></th>
              <th style="width: 20%;">Дія</th>
            </tr>
          </thead>
          <tbody>
            <tr th:each="elOrdersWorkTeamDTOList :
${ordersWorkTeamDTOList}">
              <td
th:text="{elOrdersWorkTeamDTOList.username}"></td>
              <td
th:text="{elOrdersWorkTeamDTOList.statusUser}"></td>
              <td><a th:href="{ '/' + OrderNumber +
'/edit/team/del?del=' + elOrdersWorkTeamDTOList.username}" style="color: var(--bs-
red);">Видалити</a><br></td>
            </tr>
          </tbody>
        </table>
      </div>
      <form th:method="POST" th:action="{editSaveLinkTeamAdd}"
th:object="{ordersWorkTeamDTO}">
        <h1 style="color: rgb(0,0,0);font-size: 19px;">Додати
працівника</h1>
        <p>Ідентифікатор працівника:&nbsp;<input class="form-control"
type="text" th:name="username"></p>
        <p style="text-align: center;"><button class="btn btn-
primary" type="submit" style="background: rgb(255,204,30);">Зберегти</button></p>
      </form>
      <hr>
    </div>
  </div>
  <div>
    <h1 style="color: rgb(0,0,0);font-size: 19px;">Чек-лист</h1>
    <div class="table-responsive">
      <table class="table">
        <thead>
          <tr>
            <th style="width: 60%;">Запчастина, послуга</th>

```

```

                <th style="width: 20%;">Ціна</th>
                <th style="width: 20%;">Дія</th>
            </tr>
        </thead>
        <tbody>
            <tr th:each="elOrdersCheckDTOList :
${OrdersCheckDTOList}">
                <td th:text="${elOrdersCheckDTOList.name}"></td>
                <td th:text="${elOrdersCheckDTOList.price}"></td>
                <td><a th:href="'${'/order/' + OrderNumber +
'/edit/check/del?del=' + elOrdersCheckDTOList.numberCheck}'" style="color: var(--bs-
red);">Видалити</a>&nbsp;</td>
            </tr>
        </tbody>
    </table>
</div>
    <form th:method="POST" th:action="${editSaveLinkCheckAdd}"
th:object="${ordersCheckDTO}">
        <h1 style="color: rgb(0,0,0);font-size: 19px;">Чек-лист
додати елемент</h1>
        <p>Запчастина, послуга:&nbsp;<input class="form-control"
type="text" th:name="name"></p>
        <p>Ціна:&nbsp;<input class="form-control" type="number"
th:name="price"></p>
        <p style="text-align: center;"><button class="btn btn-
primary" type="submit" style="background: rgb(255,204,30);">Зберегти</button></p>
    </form>
</div>
</div>
<hr>
</div>
<footer class="footer-dark">
    <div class="container">
        <div class="row">
            <div class="col-sm-6 col-md-3 item">
                <h3>Навігація</h3>
                <ul>
                    <li><a href="/#Info-section">Переваги</a></li>
                    <li><a href="/#Service-section">Послуги</a></li>
                    <li><a href="/#Contacts-section">Контакти</a></li>
                    <li><a href="/#Team-section">Керівники</a></li>
                    <li><a href="/#Partners-section">Партнери</a></li>
                    <li><a href="/#Reviews-section">Відгуки</a></li>
                </ul>
            </div>
            <div class="col-md-6 item text">
                <h3>СТО</h3>
                <p th:utext="'${'Такі машини, як ця, можуть бігати вічно. <br>При
належному догляді вони безсмертні. <br> – Домінік Торетто}'"></p>
                <div class="col item social"><a
href="https://www.facebook.com/vadim.ochota" style="color: rgb(255,204,30);"><i
class="icon ion-social-facebook" style="color: rgb(255,204,30);"></i></a><a
href="https://www.instagram.com/_okhotnik_" style="color: rgb(255,204,30);"><i
class="icon ion-social-instagram" style="color: rgb(255,204,30);"></i></a></div>
            </div>
            <p class="copyright">СТО © 2022</p>
        </div>
</footer>
<script src="/assets/js/jquery.min.js"></script>
<script src="/assets/bootstrap/js/bootstrap.min.js"></script>
<script src="/assets/js/bs-init.js"></script>
<script>

```

```

src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.js"
></script>
  <script src="/assets/js/Lightbox-Gallery.js"></script>
</body>

</html>

```

## В.48 Код розмітки HTML Oder-management:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-
fit=no">
  <title>service station</title>
  <meta name="description" content="Дипломний проект Охота В.В.">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
  <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
  <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
  <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
  <link rel="stylesheet" href="/assets/css/accoutant-page.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-
Code.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
  <link rel="stylesheet" href="/assets/css/Brands.css">
  <link rel="stylesheet" href="/assets/css/dialog-page.css">
  <link rel="stylesheet" href="/assets/css/director-page.css">
  <link rel="stylesheet" href="/assets/css/Features-Blue.css">
  <link rel="stylesheet" href="/assets/css/Features-Boxed.css">
  <link rel="stylesheet" href="/assets/css/Features-Clean.css">
  <link rel="stylesheet" href="/assets/css/Footer-Dark.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.cs
s">
  <link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
  <link rel="stylesheet" href="/assets/css/Map-Clean.css">
  <link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
  <link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
  <link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
  <link rel="stylesheet" href="/assets/css/Projects-Clean.css">
  <link rel="stylesheet" href="/assets/css/styles.css">
  <link rel="stylesheet" href="/assets/css/Team-Clean.css">
  <link rel="stylesheet" href="/assets/css/Team-Grid.css">
  <link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>

```

```

<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button"
style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color:
#000000;"><i class="fas fa-car" style="font-size: 24px;color:
rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse"
class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-
hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
  <div class="collapse navbar-collapse" id="navcol-1">
    <ul class="navbar-nav me-auto">
      <li class="nav-item"></li>
      <li class="nav-item"><a class="nav-link"
href="/order/create">Створити замовлення</a></li>
      <li class="nav-item dropdown" th:if="{TeamAccess == 'yes'}"><a
class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown"
href="#">Функції працівника&nbsp;</a>
        <div class="dropdown-menu">
          <a class="dropdown-item" href="/order/management"
th:if="{TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
          <a class="dropdown-item" href="/team/management"
th:if="{TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
          <a class="dropdown-item" href="/accountant"
th:if="{TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
          <a class="dropdown-item" href="/director"
th:if="{TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
        </li>
      </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
    </div>
  </div>
</nav>
<section>
  <div class="container">
    <div class="row">
      <div class="col-md-12">
        <section class="highlight-clean">
          <div class="container">
            <div class="intro">
              <h2 class="text-center">Керування замовленнями</h2>
              <p class="text-center"><span>Кількість усіх
замовлень:&nbsp;</span><span th:text="{AllOrderListSize}"></span></p>
            </div>
            <div class="buttons"><a class="btn btn-primary director-
page-navigations" type="button" th:href="{/order/management/archive}">Перейти в
архів</a></div>
          </div>
        </section>
      </div>
    </div>
  </div>
</section>
<section>
  <div class="container">
    <hr>
    <div class="row">
      <div class="col-md-3">
        <div class="accordion" role="tablist" id="accordion-8">
          <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-8 .item-1" aria-expanded="false" aria-controls="accordion-8 .item-
1">Пошук</button></h2>
            <div class="accordion-collapse collapse item-1"

```

```

role="tabpanel" data-bs-parent="#accordion-8">
    <div class="accordion-body">
        <form th:method="POST"
th:action="@{/order/management/find}" th:object="{ordersDTO}">
            <p class="mb-0"><input class="form-control"
type="number" value="0" placeholder="Номер замовлення" name="orderNumber"></p>
            <p class="mb-0" style="text-align:
center;"><button class="btn btn-primary" type="submit" style="background:
rgb(255,204,30);margin-top: 12px;margin-bottom: 17px;">Застосувати</button></p>
            </form>
        </div>
    </div>
</div>
</div>
</div>
<div class="col-md-3">
</div>
</div>
<hr>
</div>
<div class="container">
    <div class="row">
        <div class="col-md-3">
            <h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Опрацювання<br></h1>
            <p><span>Кількість замовлень:&nbsp;  </span><span
th:text="{NewOrdersSize}"></span></p>
            <hr>
            <div style="border-bottom-left-radius: 15px;">
                <div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
${NewOrders}">
                    <p style="text-align: center;"><span>Замовлення
№&nbsp;  </span><span th:text="{elNewOrder.orderNumber}"></span></p>
                    <div class="accordion" role="tablist"
th:id="{ 'accordion-' + elNewOrder.orderNumber + '0' }">
                        <div class="accordion-item">
                            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-1' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01' }">Інформація</button></h2>
                            <div th:class="{ 'accordion-collapse collapse
item-1' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
                                <div class="accordion-body">
                                    <p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;  </span><span
th:text="{elNewOrder.username}">&nbsp;  </span></p>
                                    <p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;  </span><span th:text="{elNewOrder.brandCar} + ' ' +
${elNewOrder.modelCar}">&nbsp;  </span></p>
                                    <p><span style="font-size: 16px;">Тип
роботи:&nbsp;  &nbsp; </span><span th:text="{elNewOrder.service}">&nbsp;  </span></p>
                                    <p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;  </span><span
th:text="{elNewOrder.dateCreateOrder}">&nbsp;  </span></p>
                                    <p><span style="font-size: 16px;">Дата
бажаного початку роботи:&nbsp;  </span><span
th:text="{elNewOrder.dateWork}">&nbsp;  </span></p>
                                <p class="mb-0"></p>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{#accordion-' + elNewOrder.orderNumber + '0' + ' .item-2'}" aria-
expanded="false" th:aria-controls="{#accordion-' + elNewOrder.orderNumber + '0' + '
.item-2'}">Керування інформацією</button></h2>
            <div th:class="{#accordion-collapse collapse collapse
item-2}" role="tabpanel" th:data-bs-parent="{#accordion-' + elNewOrder + '0'}">
                <div class="accordion-body">
                    <p><a th:href="{#order/' +
elNewOrder.orderNumber}">Детальніше</a></p>
                    <p><a th:href="{#order/' +
elNewOrder.orderNumber + '/edit'}">Редагувати</a></p>
                    <p><a
th:href="{#order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber}">Архівувати</a></p>
                    <p><a
th:href="{#order/management/order/dell?numberOrder=' + elNewOrder.orderNumber}"
style="color: var(--bs-red);">Видалити</a></p>
                </div>
            </div>
        </div>
        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{#accordion-' + elNewOrder.orderNumber + '0' + ' .item-3'}" aria-
expanded="false" th:aria-controls="{#accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}">Змінити статус</button></h2>
            <div class="accordion-collapse collapse item-3"
role="tabpanel" th:data-bs-parent="{#accordion-' + elNewOrder + '0'}">
                <div class="accordion-body">
                    <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{#order/management/order/status/confirmed?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Підтверджено</a></p>
                    <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{#order/management/order/status/work?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконується</a></p>
                    <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{#order/management/order/status/complete?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконано</a></p>
                </div>
            </div>
        </div>
    </div>
    </div>
    </div>
    </div>
    </div>
    </div>
    <div class="col-md-3">
        <h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Підтверджені</h1>
        <p><span>Кількість замовлень:&nbsp;</span><span>
th:text="{ConfirmedOrdersSize}"></span></p>
        <hr>
        <div style="border-bottom-left-radius: 15px;">
            <div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
{ConfirmedOrders}">

```

```

        <p style="text-align: center;"><span>Замовлення
№&nbsp;</span><span th:text="{elNewOrder.orderNumber}"></span></p>
        <div class="accordion" role="tablist"
th:id="{ 'accordion-' + elNewOrder.orderNumber + '0' }">
            <div class="accordion-item">
                <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-1' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01' }">Інформація</button></h2>
                <div th:class="{ 'accordion-collapse collapse
item-1' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
                    <div class="accordion-body">
                        <p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;</span><span
th:text="{elNewOrder.username}">&nbsp;</span></p>
                        <p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;</span><span th:text="{elNewOrder.brandCar} + ' ' +
{elNewOrder.modelCar}">&nbsp;</span></p>
                        <p><span style="font-size: 16px;">Тип
роботи:&nbsp;&nbsp;</span><span th:text="{elNewOrder.service}">&nbsp;</span></p>
                        <p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;</span><span
th:text="{elNewOrder.dateCreateOrder}">&nbsp;</span></p>
                        <p><span style="font-size: 16px;">Дата
початку роботи:&nbsp;</span><span th:text="{elNewOrder.dateWork}">&nbsp;</span></p>
                    <p class="mb-0"></p>
                </div>
            </div>
        </div>
        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-2' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-2' }">Керування інформацією</button></h2>
            <div th:class="{ 'accordion-collapse collapse
item-2' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
                <div class="accordion-body">
                    <p><a th:href="{ '/order/' +
elNewOrder.orderNumber }">Детальніше</a></p>
                    <p><a th:href="{ '/order/' +
elNewOrder.orderNumber + '/edit' }">Редагувати</a></p>
                    <p><a
th:href="{ '/order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber }">Архівувати</a></p>
                    <p><a
th:href="{ '/order/management/order/dell?numberOrder=' + elNewOrder.orderNumber }"
style="color: var(--bs-red);">Видалити</a></p>
                </div>
            </div>
        </div>
        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-3' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-3' }" >Змінити статус</button></h2>
            <div class="accordion-collapse collapse item-3"
role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
                <div class="accordion-body">
                    <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/new?numberOrder=' + elNewOrder.orderNumber

```

```

} " style="background: rgb(51,164,213);width: 100%;">Опрацювання</a></p>
<p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="'${'/order/management/order/status/work?numberOrder=' +
elNewOrder.orderNumber }'" style="background: rgb(51,164,213);width:
100%;">Виконується</a></p>
<p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="'${'/order/management/order/status/complete?numberOrder=' +
elNewOrder.orderNumber }'" style="background: rgb(51,164,213);width:
100%;">Виконано</a></p>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="col-md-3">
<h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Виконання</h1>
<p><span>Кількість замовлень:&nbsp;&nbsp;&nbsp;</span><span
th:text="'${WorkOrdersSize}'">&nbsp;&nbsp;&nbsp;</span></p>
<hr>
<div style="border-bottom-left-radius: 15px;">
<div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
${WorkOrders}">
<p style="text-align: center;"><span>Замовлення
№&nbsp;&nbsp;&nbsp;</span><span th:text="'${elNewOrder.orderNumber}'"></span></p>
<div class="accordion" role="tablist"
th:id="'${'accordion-' + elNewOrder.orderNumber + '0'}'">
<div class="accordion-item">
<h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="'${'#accordion-' + elNewOrder.orderNumber + '0' + ' .item-1'}'" aria-
expanded="false" th:aria-controls="'${'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01'}'">Інформація</button></h2>
<div th:class="'${'accordion-collapse collapse
item-1'}'" role="tabpanel" th:data-bs-parent="'${'#accordion-' + elNewOrder + '0'}'">
<div class="accordion-body">
<p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;&nbsp;&nbsp;</span><span
th:text="'${elNewOrder.username}'">&nbsp;&nbsp;&nbsp;</span></p>
<p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;&nbsp;&nbsp;</span><span th:text="'${elNewOrder.brandCar} + ' ' +
${elNewOrder.modelCar}'">&nbsp;&nbsp;&nbsp;</span></p>
<p><span style="font-size: 16px;">Тип
роботи:&nbsp;&nbsp;&nbsp;</span><span th:text="'${elNewOrder.service}'">&nbsp;&nbsp;&nbsp;</span></p>
<p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;&nbsp;&nbsp;</span><span
th:text="'${elNewOrder.dateCreateOrder}'">&nbsp;&nbsp;&nbsp;</span></p>
<p><span style="font-size: 16px;">Дата
виконання роботи:&nbsp;&nbsp;&nbsp;</span><span
th:text="'${elNewOrder.dateWork}'">&nbsp;&nbsp;&nbsp;</span></p>
<p class="mb-0"></p>
</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="'${'#accordion-' + elNewOrder.orderNumber + '0' + ' .item-2'}'" aria-

```

```

expanded="false" th:aria-controls="\${'accordion-' + elNewOrder.orderNumber + '0' + '
.item-2'}">Керування інформацією</button></h2>
    <div th:class="\${'accordion-collapse collapse collapse
item-2'}" role="tabpanel" th:data-bs-parent="\${'#accordion-' + elNewOrder + '0'}">
        <div class="accordion-body">
            <p><a th:href="\${'/order/' +
elNewOrder.orderNumber}">Детальніше</a></p>
            <p><a th:href="\${'/order/' +
elNewOrder.orderNumber + '/edit'}">Редагувати</a></p>
            <p><a
th:href="\${'/order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber}">Архівувати</a></p>
            <p><a
th:href="\${'/order/management/order/dell?numberOrder=' + elNewOrder.orderNumber}"
style="color: var(--bs-red);">Видалити</a></p>
        </div>
    </div>
</div>
<div class="accordion-item">
    <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="\${'#accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}" aria-
expanded="false" th:aria-controls="\${'accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}" >Змінити статус</button></h2>
    <div class="accordion-collapse collapse collapse item-3"
role="tabpanel" th:data-bs-parent="\${'#accordion-' + elNewOrder + '0'}">
        <div class="accordion-body">
            <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="\${'/order/management/order/status/new?numberOrder=' + elNewOrder.orderNumber
}" style="background: rgb(51,164,213);width: 100%;">Опрацювання</a></p>
            <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="\${'/order/management/order/status/confirmed?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Підтверджено</a></p>
            <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="\${'/order/management/order/status/complete?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконано</a></p>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="col-md-3">
    <h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Виконано</h1>
    <p><span>Кількість замовлень:&nbsp;&nbsp;&nbsp;</span><span>
th:text="\${CompleteOrdersSize}"></span></p>
    <hr>
    <div style="border-bottom-left-radius: 15px;">
        <div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
\${CompleteOrders}">
            <p style="text-align: center;"><span>Замовлення
№&nbsp;&nbsp;&nbsp;</span><span th:text="\${elNewOrder.orderNumber}"></span><br><span>Чистий
дохід: </span><span th:text="\${elNewOrder.netProfit}"></span></p>
        </div>
    </div>
th:id="\${'accordion-' + elNewOrder.orderNumber + '0'}">

```

```

        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-1'}" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01'}">Інформація</button></h2>
            <div th:class="{ 'accordion-collapse collapse
item-1'}" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0'}">
                <div class="accordion-body">
                    <p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;</span><span
th:text="{elNewOrder.username}">&nbsp;</span></p>
                    <p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;</span><span th:text="{elNewOrder.brandCar} + ' ' +
${elNewOrder.modelCar}">&nbsp;</span></p>
                    <p><span style="font-size: 16px;">Тип
роботи:&nbsp;&nbsp;</span><span th:text="{elNewOrder.service}">&nbsp;</span></p>
                    <p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;</span><span
th:text="{elNewOrder.dateCreateOrder}">&nbsp;</span></p>
                    <p><span style="font-size: 16px;">Дата
завершення роботи:&nbsp;</span><span
th:text="{elNewOrder.dateWork}">&nbsp;</span></p>
                <p class="mb-0"></p>
            </div>
        </div>
        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-2'}" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-2'}">Керування інформацією</button></h2>
            <div th:class="{ 'accordion-collapse collapse
item-2'}" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0'}">
                <div class="accordion-body">
                    <p><a th:href="{ '/order/' +
elNewOrder.orderNumber}">Детальніше</a></p>
                    <p><a th:href="{ '/order/' +
elNewOrder.orderNumber + '/edit'}">Редагувати</a></p>
                    <p><a
th:href="{ '/order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber}">Архівувати</a></p>
                    <p><a
th:href="{ '/order/management/order/dell?numberOrder=' + elNewOrder.orderNumber}"
style="color: var(--bs-red);">Видалити</a></p>
                </div>
            </div>
        </div>
        <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-3'}" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}" >ЗМІНИТИ СТАТУС</button></h2>
            <div class="accordion-collapse collapse item-3"
role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0'}">
                <div class="accordion-body">
                    <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/new?numberOrder=' + elNewOrder.orderNumber
}" style="background: rgb(51,164,213);width: 100%;">Опрацювання</a></p>
                    <p><a class="btn btn-primary director-
page-navigations" role="button"

```



## В.49 Код розмітки HTML Personal-page:

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
  <title>service station</title>
  <meta name="description" content="Дипломний проект Охота В.В.">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
  <link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
  <link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
  <link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
  <link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
  <link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
  <link rel="stylesheet" href="/assets/css/accoutant-page.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">
  <link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
  <link rel="stylesheet" href="/assets/css/Brands.css">
  <link rel="stylesheet" href="/assets/css/dialog-page.css">
  <link rel="stylesheet" href="/assets/css/director-page.css">
  <link rel="stylesheet" href="/assets/css/Features-Blue.css">
  <link rel="stylesheet" href="/assets/css/Features-Boxed.css">
  <link rel="stylesheet" href="/assets/css/Features-Clean.css">
  <link rel="stylesheet" href="/assets/css/Footer-Dark.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
  <link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.css">
  <link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
  <link rel="stylesheet" href="/assets/css/Map-Clean.css">
  <link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
  <link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
  <link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
  <link rel="stylesheet" href="/assets/css/Projects-Clean.css">
  <link rel="stylesheet" href="/assets/css/styles.css">
  <link rel="stylesheet" href="/assets/css/Team-Clean.css">
  <link rel="stylesheet" href="/assets/css/Team-Grid.css">
  <link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button" style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color: #000000;"><i class="fas fa-car" style="font-size: 24px;color: rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse" id="navcol-1">

```

```

        <ul class="navbar-nav me-auto">
            <li class="nav-item"></li>
            <li class="nav-item"><a class="nav-link"
href="/order/create">Створити замовлення</a></li>
            <li class="nav-item dropdown" th:if="{TeamAccess == 'yes'}"><a
class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown"
href="#">Функції працівника&nbsp;</a>
                <div class="dropdown-menu">
                    <a class="dropdown-item" href="/order/management"
th:if="{TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
                    <a class="dropdown-item" href="/team/management"
th:if="{TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
                    <a class="dropdown-item" href="/accountant"
th:if="{TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
                    <a class="dropdown-item" href="/director"
th:if="{TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
                </li>
        </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
    </div>
</div>
</nav>
<section>
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <section class="highlight-clean">
                    <div class="container">
                        <div class="intro">
                            <h2 class="text-center">Керування замовленнями</h2>
                            <p class="text-center"><span>Кількість усіх
замовлень: &nbsp;</span><span th:text="{AllOrderListSize}"></span></p>
                        </div>
                        <div class="buttons"><a class="btn btn-primary director-
page-navigations" type="button" th:href="{@{/order/management/archive]}">Перейти в
архів</a></div>
                    </div>
                </section>
            </div>
        </div>
    </div>
</section>
<section>
    <div class="container">
        <hr>
        <div class="row">
            <div class="col-md-3">
                <div class="accordion" role="tablist" id="accordion-8">
                    <div class="accordion-item">
                        <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-8 .item-1" aria-expanded="false" aria-controls="accordion-8 .item-
1">Пошук</button></h2>
                        <div class="accordion-collapse collapse item-1"
role="tabpanel" data-bs-parent="#accordion-8">
                            <div class="accordion-body">
                                <form th:method="POST"
th:action="{@{/order/management/find}}" th:object="{ordersDTO}">
                                    <p class="mb-0"><input class="form-control"
type="number" value="0" placeholder="Номер замовлення" name="orderNumber"></p>
                                    <p class="mb-0" style="text-align:
center;"><button class="btn btn-primary" type="submit" style="background:

```

```

rgb(255,204,30);margin-top: 12px;margin-bottom: 17px;"/>Застосувати</button></p>
      </form>
    </div>
  </div>
</div>
</div>
</div>
</div>
<div class="col-md-3">
</div>
</div>
<div class="container">
  <div class="row">
    <div class="col-md-3">
      <h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Опрацювання<br></h1>
      <p><span>Кількість замовлень:&nbsp;</span><span
th:text="{NewOrdersSize}"></span></p>
      <hr>
      <div style="border-bottom-left-radius: 15px;">
        <div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
${NewOrders}">
          <p style="text-align: center;"><span>Замовлення
№&nbsp;</span><span th:text="{elNewOrder.orderNumber}"></span></p>
          <div class="accordion" role="tablist"
th:id="{ 'accordion-' + elNewOrder.orderNumber + '0' }">
            <div class="accordion-item">
              <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + '
.item-1' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01' }">Інформація</button></h2>
              <div th:class="{ 'accordion-collapse collapse
item-1' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
                <div class="accordion-body">
                  <p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;</span><span
th:text="{elNewOrder.username}">&nbsp;</span></p>
                  <p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;</span><span th:text="{elNewOrder.brandCar} + ' ' +
${elNewOrder.modelCar}">&nbsp;</span></p>
                  <p><span style="font-size: 16px;">Тип
роботи:&nbsp;&nbsp;</span><span th:text="{elNewOrder.service}">&nbsp;</span></p>
                  <p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;</span><span
th:text="{elNewOrder.dateCreateOrder}">&nbsp;</span></p>
                  <p><span style="font-size: 16px;">Дата
бажаного початку роботи:&nbsp;</span><span
th:text="{elNewOrder.dateWork}">&nbsp;</span></p>
                  <p class="mb-0"></p>
                </div>
              </div>
            </div>
          </div>
        <div class="accordion-item">
          <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + '
.item-2' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-2' }">Керування інформацією</button></h2>
          <div th:class="{ 'accordion-collapse collapse
item-2' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">

```

```

                <div class="accordion-body">
                    <p><a th:href="\${'/order/' +
elNewOrder.orderNumber}">Детальніше</a></p>
                    <p><a th:href="\${'/order/' +
elNewOrder.orderNumber + '/edit'}">Редагувати</a></p>
                    <p><a
th:href="\${'/order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber}">Архівувати</a></p>
                    <p><a
th:href="\${'/order/management/order/dell?numberOrder=' + elNewOrder.orderNumber}"
style="color: var(--bs-red);">Видалити</a></p>
                </div>
            </div>
            <div class="accordion-item">
                <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="\${'#accordion-' + elNewOrder.orderNumber + '0' + ' .item-3'}" aria-
expanded="false" th:aria-controls="\${'accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}" >Змінити статус</button></h2>
                <div class="accordion-collapse collapse item-3"
role="tabpanel" th:data-bs-parent="\${'#accordion-' + elNewOrder + '0'}">
                    <div class="accordion-body">
                        <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="\${'/order/management/order/status/confirmed?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Підтверджено</a></p>
                        <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="\${'/order/management/order/status/work?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконується</a></p>
                        <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="\${'/order/management/order/status/complete?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконано</a></p>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-md-3">
            <h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Підтверджені</h1>
            <p><span>Кількість замовлень:&nbsp;</span><span
th:text="\${ConfirmedOrdersSize}"></span></p>
            <hr>
            <div style="border-bottom-left-radius: 15px;">
                <div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
${ConfirmedOrders}">
                    <p style="text-align: center;"><span>Замовлення
№&nbsp;</span><span th:text="\${elNewOrder.orderNumber}"></span></p>
                    <div class="accordion" role="tablist"
th:id="\${'accordion-' + elNewOrder.orderNumber + '0'}">
                        <div class="accordion-item">
                            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="\${'#accordion-' + elNewOrder.orderNumber + '0' + ' .item-1'}" aria-

```

```

expanded="false" th:aria-controls="${'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01'}">Інформація</button></h2>
    <div th:class="${'accordion-collapse collapse collapse
item-1'}" role="tabpanel" th:data-bs-parent="${'#accordion-' + elNewOrder + '0'}">
        <div class="accordion-body">
            <p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;</span><span
th:text="${elNewOrder.username}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;</span><span th:text="${elNewOrder.brandCar} + ' ' +
${elNewOrder.modelCar}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Тип
роботи:&nbsp;&nbsp;</span><span th:text="${elNewOrder.service}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;</span><span
th:text="${elNewOrder.dateCreateOrder}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Дата
початку роботи:&nbsp;</span><span th:text="${elNewOrder.dateWork}">&nbsp;</span></p>
            <p class="mb-0"></p>
        </div>
    </div>
</div>
<div class="accordion-item">
    <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="${'#accordion-' + elNewOrder.orderNumber + '0' + '
.item-2'}" aria-
expanded="false" th:aria-controls="${'accordion-' + elNewOrder.orderNumber + '0' + '
.item-2'}">Керування інформацією</button></h2>
    <div th:class="${'accordion-collapse collapse collapse
item-2'}" role="tabpanel" th:data-bs-parent="${'#accordion-' + elNewOrder + '0'}">
        <div class="accordion-body">
            <p><a th:href="${'/order/' +
elNewOrder.orderNumber}">Детальніше</a></p>
            <p><a th:href="${'/order/' +
elNewOrder.orderNumber + '/edit'}">Редагувати</a></p>
            <p><a
th:href="${'/order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber}">Архівувати</a></p>
            <p><a
th:href="${'/order/management/order/dell?numberOrder=' + elNewOrder.orderNumber}"
style="color: var(--bs-red);">Видалити</a></p>
        </div>
    </div>
</div>
<div class="accordion-item">
    <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="${'#accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}" aria-
expanded="false" th:aria-controls="${'accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}" >Змінити статус</button></h2>
    <div class="accordion-collapse collapse collapse item-3"
role="tabpanel" th:data-bs-parent="${'#accordion-' + elNewOrder + '0'}">
        <div class="accordion-body">
            <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="${'/order/management/order/status/new?numberOrder=' + elNewOrder.orderNumber
}" style="background: rgb(51,164,213);width: 100%;">Опрацювання</a></p>
            <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="${'/order/management/order/status/work?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконується</a></p>
            <p><a class="btn btn-primary director-
page-navigations" role="button"

```

```

th:href="{ '/order/management/order/status/complete?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконано</a></p>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="col-md-3">
<h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Виконання</h1>
<p><span>Кількість замовлень:&nbsp;</span><span
th:text="{WorkOrdersSize}">&nbsp;</span></p>
<hr>
<div style="border-bottom-left-radius: 15px;">
<div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
${WorkOrders}">
<p style="text-align: center;"><span>Замовлення
№&nbsp;</span><span th:text="{elNewOrder.orderNumber}"></span></p>
<div class="accordion" role="tablist"
th:id="{ 'accordion-' + elNewOrder.orderNumber + '0' }">
<div class="accordion-item">
<h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + '
.item-1' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01' }">Інформація</button></h2>
<div th:class="{ 'accordion-collapse collapse
item-1' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
<div class="accordion-body">
<p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;</span><span
th:text="{elNewOrder.username}">&nbsp;</span></p>
<p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;</span><span th:text="{elNewOrder.brandCar} + ' ' +
${elNewOrder.modelCar}">&nbsp;</span></p>
<p><span style="font-size: 16px;">Тип
роботи:&nbsp;&nbsp;</span><span th:text="{elNewOrder.service}">&nbsp;</span></p>
<p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;</span><span
th:text="{elNewOrder.dateCreateOrder}">&nbsp;</span></p>
<p><span style="font-size: 16px;">Дата
виконання роботи:&nbsp;</span><span
th:text="{elNewOrder.dateWork}">&nbsp;</span></p>
<p class="mb-0"></p>
</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + '
.item-2' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-2' }">Керування інформацією</button></h2>
<div th:class="{ 'accordion-collapse collapse
item-2' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
<div class="accordion-body">
<p><a th:href="{ '/order/' +
elNewOrder.orderNumber }">Детальніше</a></p>
<p><a th:href="{ '/order/' +

```

```

elNewOrder.orderNumber + '/edit'}">Редагувати</a></p>
    <p><a
th:href="{ '/order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber }">Архівувати</a></p>
    <p><a
th:href="{ '/order/management/order/dell?numberOrder=' + elNewOrder.orderNumber }"
style="color: var(--bs-red);">Видалити</a></p>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-3'}" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-3'}" >Змінити статус</button></h2>
    <div class="accordion-collapse collapse item-3"
role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0'}">
      <div class="accordion-body">
        <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/new?numberOrder=' + elNewOrder.orderNumber
}" style="background: rgb(51,164,213);width: 100%;">Опрацювання</a></p>
        <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/confirmed?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Підтверджено</a></p>
        <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/complete?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконано</a></p>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
<div class="col-md-3">
  <h1 style="color: rgb(0,0,0);font-size: 20px;text-align:
center;">Виконано</h1>
  <p><span>Кількість замовлень:&nbsp;</span><span>
th:text="{ {CompleteOrdersSize} }"></span></p>
  <hr>
  <div style="border-bottom-left-radius: 15px;">
    <div class="col" style="background: rgb(255,255,255);border-
top-left-radius: 15px;border-top-right-radius: 15px;border-bottom-right-radius:
15px;border-bottom-left-radius: 15px;border-style: outset;" th:each="elNewOrder :
${CompleteOrders}">
      <p style="text-align: center;"><span>Замовлення
№&nbsp;</span><span th:text="{ {elNewOrder.orderNumber} }"></span><br><span>Чистий
дохід: </span><span th:text="{ {elNewOrder.netProfit} }"></span></p>
    <div class="accordion" role="tablist"
th:id="{ 'accordion-' + elNewOrder.orderNumber + '0'}">
      <div class="accordion-item">
        <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-1'}" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-01'}" >Інформація</button></h2>
        <div th:class="{ 'accordion-collapse collapse
item-1'}" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0'}">

```

```

        <div class="accordion-body">
            <p><span style="font-size:
16px;">Індифікатор клієнта:&nbsp;</span><span
th:text="{elNewOrder.username}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Марка
та модель авто:&nbsp;</span><span th:text="{elNewOrder.brandCar} + ' ' +
{elNewOrder.modelCar}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Тип
роботи:&nbsp;&nbsp;</span><span th:text="{elNewOrder.service}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Дата
створення заявки:&nbsp;</span><span
th:text="{elNewOrder.dateCreateOrder}">&nbsp;</span></p>
            <p><span style="font-size: 16px;">Дата
завершення роботи:&nbsp;</span><span
th:text="{elNewOrder.dateWork}">&nbsp;</span></p>
            <p class="mb-0"></p>
        </div>
    </div>
    <div class="accordion-item">
        <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-2' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-2' }">Керування інформацією</button></h2>
        <div th:class="{ 'accordion-collapse collapse
item-2' }" role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
            <div class="accordion-body">
                <p><a th:href="{ '/order/' +
elNewOrder.orderNumber }">Детальніше</a></p>
                <p><a th:href="{ '/order/' +
elNewOrder.orderNumber + '/edit' }">Редагувати</a></p>
                <p><a
th:href="{ '/order/management/archive/add?numberOrder=' +
elNewOrder.orderNumber }">Архівувати</a></p>
                <p><a
th:href="{ '/order/management/order/dell?numberOrder=' + elNewOrder.orderNumber }"
style="color: var(--bs-red);">Видалити</a></p>
            </div>
        </div>
    </div>
    <div class="accordion-item">
        <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" th:data-bs-
target="{ '#accordion-' + elNewOrder.orderNumber + '0' + ' .item-3' }" aria-
expanded="false" th:aria-controls="{ 'accordion-' + elNewOrder.orderNumber + '0' + '
.item-3' }">Змінити статус</button></h2>
        <div class="accordion-collapse collapse item-3"
role="tabpanel" th:data-bs-parent="{ '#accordion-' + elNewOrder + '0' }">
            <div class="accordion-body">
                <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/new?numberOrder=' + elNewOrder.orderNumber
}" style="background: rgb(51,164,213);width: 100%;">Опрацювання</a></p>
                <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/confirmed?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Підтверджено</a></p>
                <p><a class="btn btn-primary director-
page-navigations" role="button"
th:href="{ '/order/management/order/status/work?numberOrder=' +
elNewOrder.orderNumber }" style="background: rgb(51,164,213);width:
100%;">Виконується</a></p>
            </div>
        </div>
    </div>

```



```

<title>service station</title>
<meta name="description" content="Дипломний проект Охота В.В.">
<link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
<link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
<link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
<link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
<link rel="icon" type="image/png" sizes="512x512" href="/assets/img/icon.png">
<link rel="stylesheet" href="/assets/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="/assets/fonts/fontawesome-all.min.css">
<link rel="stylesheet" href="/assets/fonts/font-awesome.min.css">
<link rel="stylesheet" href="/assets/fonts/ionicons.min.css">
<link rel="stylesheet" href="/assets/fonts/material-icons.min.css">
<link rel="stylesheet" href="/assets/fonts/fontawesome5-overrides.min.css">
<link rel="stylesheet" href="/assets/css/accountant-page.css">
<link rel="stylesheet" href="/assets/css/Bootstrap-4-Calendar-No-Custom-Code.css">
<link rel="stylesheet" href="/assets/css/Bootstrap-Calendar.css">
<link rel="stylesheet" href="/assets/css/Brands.css">
<link rel="stylesheet" href="/assets/css/dialog-page.css">
<link rel="stylesheet" href="/assets/css/director-page.css">
<link rel="stylesheet" href="/assets/css/Features-Blue.css">
<link rel="stylesheet" href="/assets/css/Features-Boxed.css">
<link rel="stylesheet" href="/assets/css/Features-Clean.css">
<link rel="stylesheet" href="/assets/css/Footer-Dark.css">
<link rel="stylesheet" href="/assets/css/Highlight-Blue.css">
<link rel="stylesheet" href="/assets/css/Highlight-Clean.css">
<link rel="stylesheet" href="/assets/css/Highlight-Phone.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/baguettebox.js/1.10.0/baguetteBox.min.css">
<link rel="stylesheet" href="/assets/css/Lightbox-Gallery.css">
<link rel="stylesheet" href="/assets/css/Map-Clean.css">
<link rel="stylesheet" href="/assets/css/Modern-Contact-Form.css">
<link rel="stylesheet" href="/assets/css/Navigation-with-Button.css">
<link rel="stylesheet" href="/assets/css/Newsletter-Subscription-Form.css">
<link rel="stylesheet" href="/assets/css/Projects-Clean.css">
<link rel="stylesheet" href="/assets/css/styles.css">
<link rel="stylesheet" href="/assets/css/Team-Clean.css">
<link rel="stylesheet" href="/assets/css/Team-Grid.css">
<link rel="stylesheet" href="/assets/css/Testimonials.css">
</head>

<body>
<nav class="navbar navbar-light navbar-expand-lg navigation-clean-button" style="background: rgb(238,244,247);">
  <div class="container"><a class="navbar-brand" href="/" style="color: #000000;"><i class="fas fa-car" style="font-size: 24px;color: rgb(255,204,30);"></i><span>СТО</span></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse" id="navcol-1">
      <ul class="navbar-nav me-auto">
        <li class="nav-item"></li>
        <li class="nav-item"><a class="nav-link" href="/order/create">СТВОРИТИ ЗАМОВЛЕННЯ</a></li>
        <li class="nav-item dropdown" th:if="${TeamAccess == 'yes'}"><a class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown" href="#">Функції працівника&nbsp;</a>
          <div class="dropdown-menu">
            <a class="dropdown-item" href="/order/management" th:if="${TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
            <a class="dropdown-item" href="/team/management" th:if="${TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
            <a class="dropdown-item" href="/accountant">

```

```

th:if="{TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
      <a class="dropdown-item" href="/director"
th:if="{TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
    </li>
  </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
  </div>
</div>
</nav>
<section>
  <div class="container">
    <div class="row">
      <div class="col-md-12">
        <section class="highlight-clean">
          <div class="container">
            <div class="intro">
              <br>
              <h2 class="text-center">Керування персоналом</h2>
              <p class="text-center"><span>Кількість
Працівників:&nbsp;</span><span th:text="{teamListSize}"></span></p>
            </div>
            <div class="buttons"></div>
          </div>
        </section>
      </div>
    </div>
  </div>
  <br>
  <div class="container">
    <div class="row">
      <div class="col-md-4">
        <div class="accordion" role="tablist" id="accordion-4">
          <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-4 .item-1" aria-expanded="false" aria-controls="accordion-4 .item-
1">Знайти працівника</button></h2>
            <div class="accordion-collapse collapse item-1"
role="tabpanel" data-bs-parent="#accordion-4">
              <div class="accordion-body">
                <form th:method="POST"
th:action="@{/team/management/find}" th:object="{usersDTO}">
                  <p class="mb-0"><span>Ідентифікатор
користувача:</span><br><input class="form-control" type="text" style="width: 100%;"
th:name="personalIdent"></p><br>
                  <p class="mb-0" style="text-align:
center;"><button class="btn btn-light action-button" id="navbar-spun-buton"
type="submit">Пошук</button><br></p>
                </form>
              </div>
            </div>
          </div>
          <div class="accordion-item">
            <h2 class="accordion-header" role="tab"><button
class="accordion-button collapsed" data-bs-toggle="collapse" data-bs-
target="#accordion-4 .item-2" aria-expanded="false" aria-controls="accordion-4 .item-
2">Додати/редагувати працівника</button></h2>
            <div class="accordion-collapse collapse item-2"
role="tabpanel" data-bs-parent="#accordion-4">
              <div class="accordion-body">
                <form th:method="POST"
th:action="@{/team/management/add}" th:object="{usersDTO}">

```







```

th:if="{TeamAccessCRM == 'yes' || TeamAccessDirector == 'yes'}">CRM</a>
    <a class="dropdown-item" href="/team/management"
th:if="{TeamAccessTeam == 'yes' || TeamAccessDirector == 'yes'}">Відділ кадрів</a>
    <a class="dropdown-item" href="/accountant"
th:if="{TeamAccessAcc == 'yes' || TeamAccessDirector == 'yes'}">Бухгалтерська</a>
    <a class="dropdown-item" href="/director"
th:if="{TeamAccessDirector == 'yes'}">Кабінет директора</a></div>
    </li>
    </ul><span class="navbar-text actions" style="color: var(--bs-blue);"> <a
id="navbar-register" class="login" href="/logout">Вихід</a><a class="btn btn-light
action-button" role="button" id="navbar-spun-buton" href="/personal-page">Особистий
кабінет</a></span>
    </div>
</div>
</nav>
<section>
    <div class="container">
        <hr>
        <div class="row">
            <div class="col-md-12">
                <div class="card">
                    <div class="card-body">
                        <p class="card-text" style="font-size:
21px;"><span>Транзакція Id:&nbsp;</span><span th:text="{Number}"></span></p>
                        <hr>
                        <p class="card-text"><span>Призначення:&nbsp;</span><span
th:text="{Type}"></span></p>
                        <p class="card-text"><span>Ідентифікатор
користувача:&nbsp;</span><span th:text="{Username}"></span></p>
                        <p class="card-text"><span>Пропрацьовано
годин:&nbsp;</span><span th:text="{ClockWork}"></span></p>
                        <p class="card-text"><span>Короткі
деталі:&nbsp;</span><span th:text="{Info}"></span></p>
                        <p class="card-text"><span>Повна
інформація:&nbsp;</span><span th:text="{FullInfo}"></span></p>
                        <p class="card-text"><span>Сума:&nbsp;</span><span
th:text="{Money}"></span></p>
                        <p class="card-text" style="text-align:
right;"><span>Видано:</span>&nbsp;<span th:text="{CreatorRole}"></span>&nbsp;<a
th:href="'/@' + {UsernameCreator}"><span
th:text="{UsernameCreator}"></span></a></p>
                        <p class="card-text" style="text-align:
right;"><span>Дата:&nbsp;</span><span th:text="{Date}"></span></p>
                        <div th:if="{EditAccess} == 'yes'">
                            <hr>
                            <div class="accordion" role="tablist" id="accordion-
1">
                                <div class="accordion-item">
                                    <h2 class="accordion-header"
role="tab"><button class="accordion-button collapsed" data-bs-toggle="collapse" data-
bs-target="#accordion-1 .item-1" aria-expanded="false" aria-controls="accordion-1
.item-1">Редагувати<br></button></h2>
                                    <div class="accordion-collapse collapse item-
1" role="tabpanel" data-bs-parent="#accordion-1">
                                        <div class="accordion-body">
                                            <form th:method="POST"
th:action="'/transactions/' + {Number} + '/edit'"
th:object="{transactionsInfoDTO}">
<p><span>Призначення:&nbsp;</span><input class="form-control" type="text"
th:value="{Type}" th:name="type"></p>
                                                <p><span>Ідентифікатор
користувача:&nbsp;</span><input class="form-control" type="text"
th:value="{Username}" th:name="username"></p>

```



```

    <hr>
  </div>
</section>
<footer class="footer-dark">
  <div class="container">
    <div class="row">
      <div class="col-sm-6 col-md-3 item">
        <h3>Навігація</h3>
        <ul>
          <li><a href="#Info-section">Переваги</a></li>
          <li><a href="#Service-section">Послуги</a></li>
          <li><a href="#Contacts-section">Контакти</a></li>
          <li><a href="#Team-section">Керівники</a></li>
          <li><a href="#Partners-section">Партнери</a></li>
          <li><a href="#Reviews-section">Відгуки</a></li>
        </ul>
      </div>
      <div class="col-md-6 item text">
        <h3>СТО</h3>
        <p th:utext="{ 'Такі машини, як ця, можуть бігати вічно. <br>При належному догляді вони безсмертні. <br>— Домінік Торетто' }"></p>
      </div>
      <div class="col item social"><a
href="https://www.facebook.com/vadim.ochota" style="color: rgb(255,204,30);"><i
class="icon ion-social-facebook" style="color: rgb(255,204,30);"></i></a><a
href="https://www.instagram.com/_okhotnik_/" style="color: rgb(255,204,30);"><i
class="icon ion-social-instagram" style="color: rgb(255,204,30);"></i></a></div>
      <p class="copyright">СТО © 2022</p>
    </div>
  </footer>
  <script src="/assets/js/jquery.min.js"></script>
  <script src="/assets/bootstrap/js/bootstrap.min.js"></script>
  <script src="/assets/js/bs-init.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.10.0/baguetteBox.min.js"
></script>
  <script src="/assets/js/Lightbox-Gallery.js"></script>
</body>

</html>

```

ДОДАТОК Г  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

Хмельницький Національний Університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

Дипломний проект, на тему:

«Інтернет-платформа із CRM-системою для станцій технічного  
обслуговування автомобілів»

Студент: Охота Вадим Володимирович

Керівник: Бедратюк Леонід Петрович, д-р фіз.-мат. наук професор

### Актуальність дипломного проекту

Актуальність теми полягає в тому, що на сьогодні розроблюване програмне забезпечення є надзвичайно потрібним, більшість рішень які присутні на ринку не надають можливостей реалізації єдиного програмного продукту для підприємства сервісного обслуговування автомобілів. Для кращої взаємодії з клієнтами та процесом виконання замовлення використовуються CRM системи, спеціалізованих систем для даного типу підприємства недостатньо на ринку програмного забезпечення, що створює необхідність для створення та впровадження власних спеціалізованих CRM систем. Наявні на ринку системи здебільшого не надають можливості створення веб-сайту підприємства який взаємодіє із CRM, що у свою чергу зменшує ефективність взаємодії з клієнтами. Окрім того, більшість програмних продуктів потребують щомісячної ліцензії для можливості їх використання, щоб забезпечити всі процеси на підприємстві необхідно хоча б декілька програмних продуктів, що у свою чергу приносить збільшення витрат.

## Мета проекту

Мета проекту – розробка інтернет-платформи для станції сервісного обслуговування автомобілів, яка буде слугувати як сайт візитівка із особистим кабінетом для користувачів, із можливістю відстеження виконання замовлення та інформації про виконання. Також програмне забезпечення має включати у себе робочі сторінки для працівників СТО:

- CRM;
- керування працівниками;
- бухгалтерського обліку;
- відображення взаємодії із програмним забезпеченням.

## Завдання проекту

### Завдання для реалізації проекту:

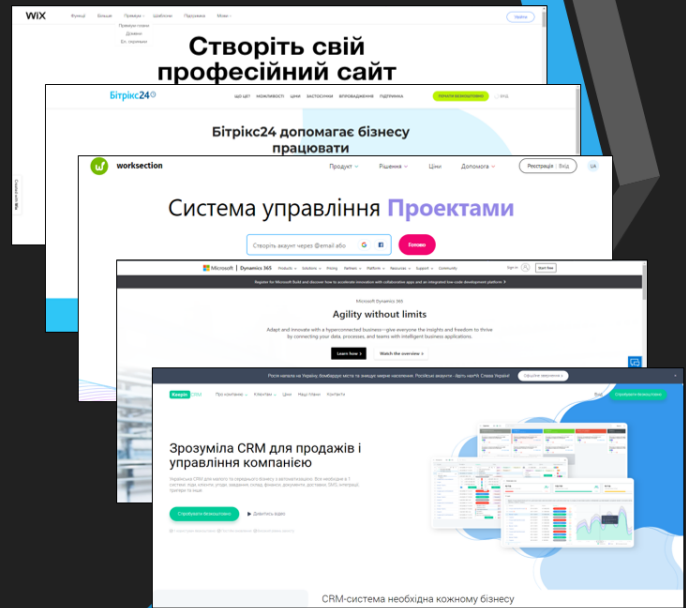
1. провести аналіз предметної області;
2. проаналізувати наявне програмне забезпечення;
3. сформулювати технічне завдання;
4. спроектувати програмний продукт;
5. виконати програмну реалізацію програмного продукту;
6. провести тестові випробування застосунку;
7. контейнеризувати програмний продукт у програмному забезпеченні Docker

## Аналіз існуючих рішень

Переважна більшість продуктів, які доступні на ринку мають направлений функціонал для створення веб-сайтів або CRM систем. Що означає нестачу можливостей для створення комплексних рішень. Слід зауважити що для використання та побудови персоналізованих систем потрібно придбати ліцензію та володіти вміннями використання сервісу.

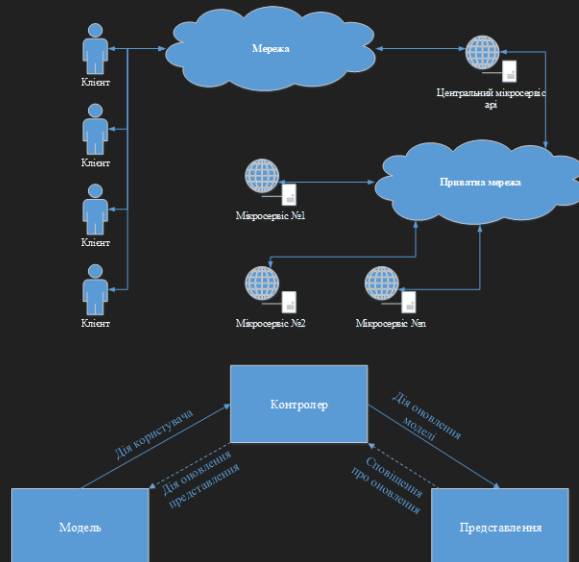
Існуючі рішення:

- WIX
- Bitrix24
- Worksection
- Dynamics 365
- КерерinCRM

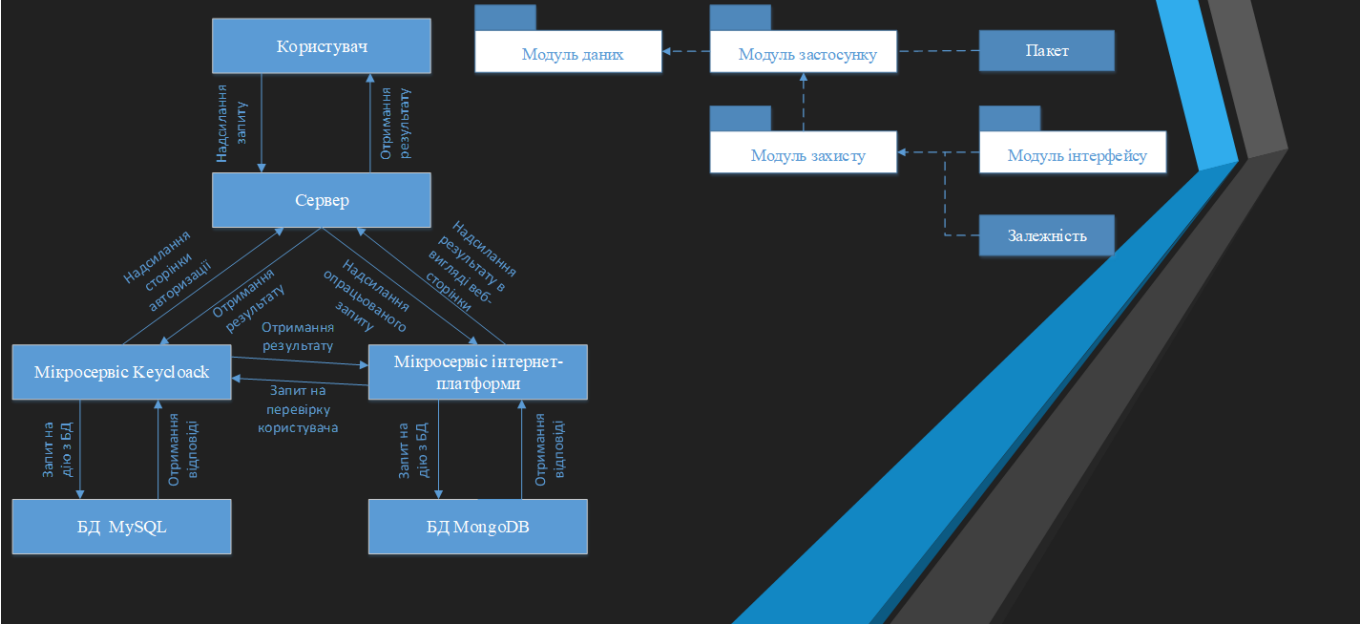


## Архітектура інтернет-платформи

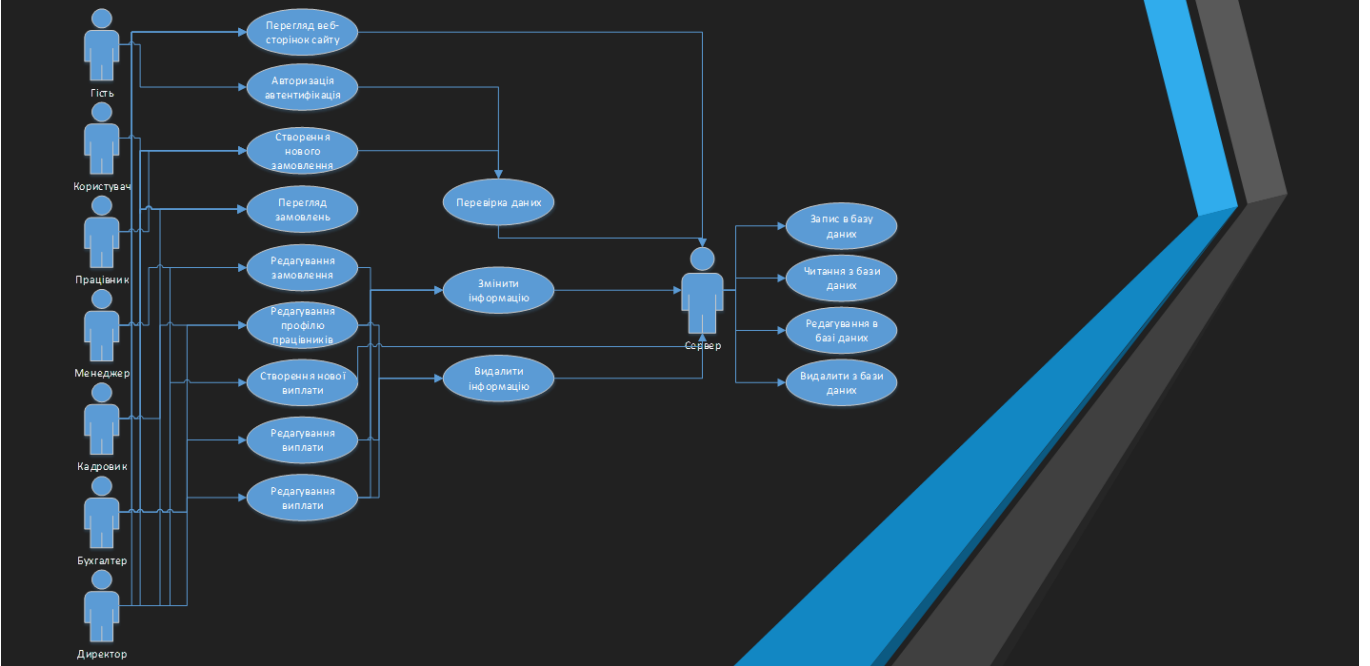
Для реалізації програмного продукту вибрано мікросервісну архітектуру та шаблон MVC



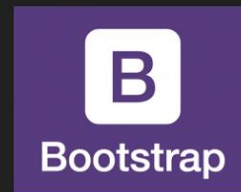
## Модулі інтернет-платформи та потоки даних



## Діаграма варіантів використання

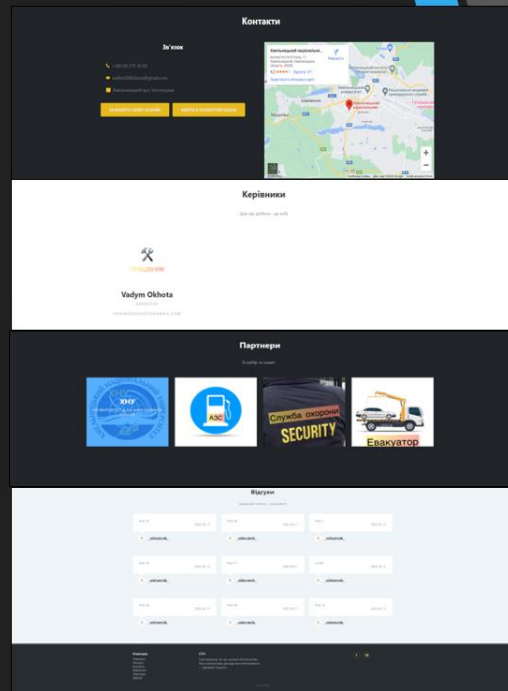
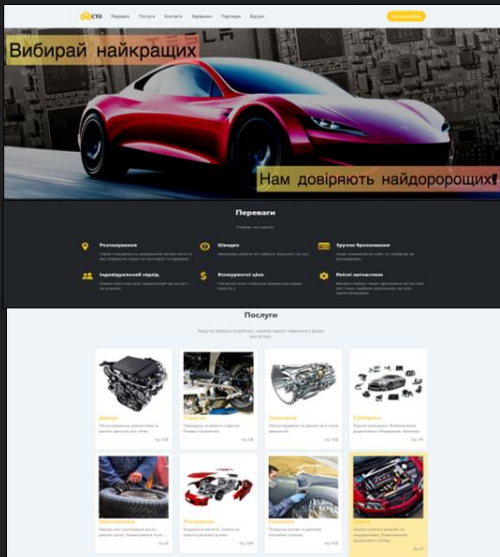


### Інструменти та технології



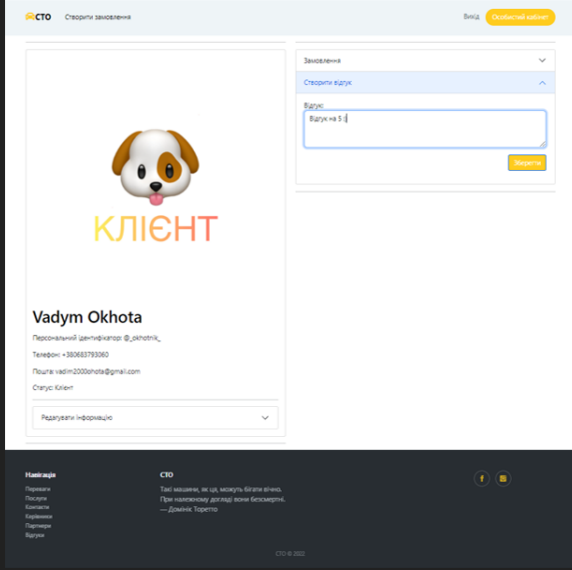
### Результат розробки

#### Головна сторінка

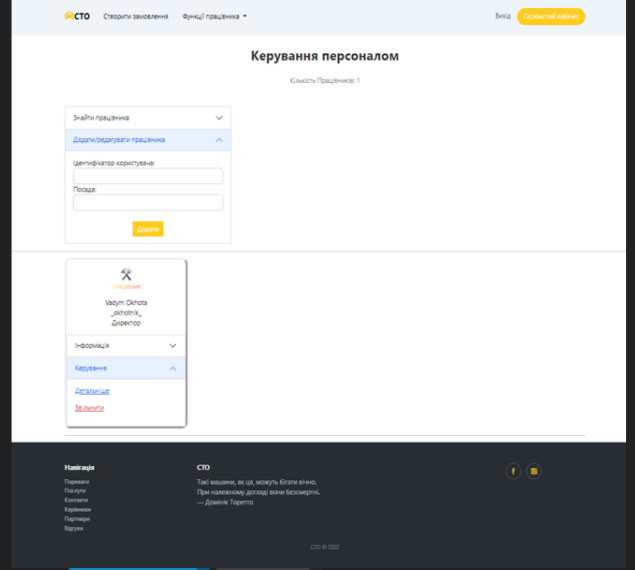


# Результат розробки

## Особиста сторінка

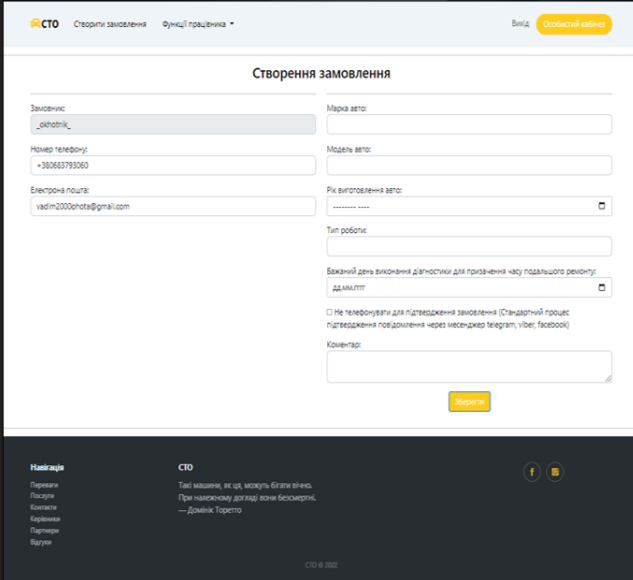


## Сторінка керування працівниками

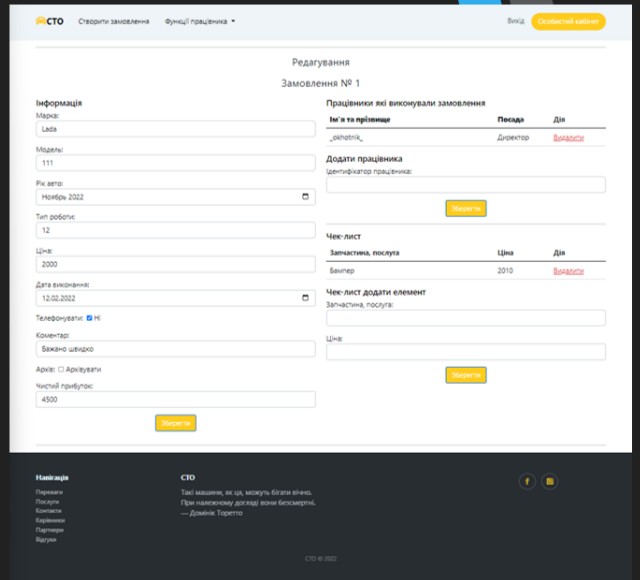


# Результат розробки

## Сторінка створення замовлення

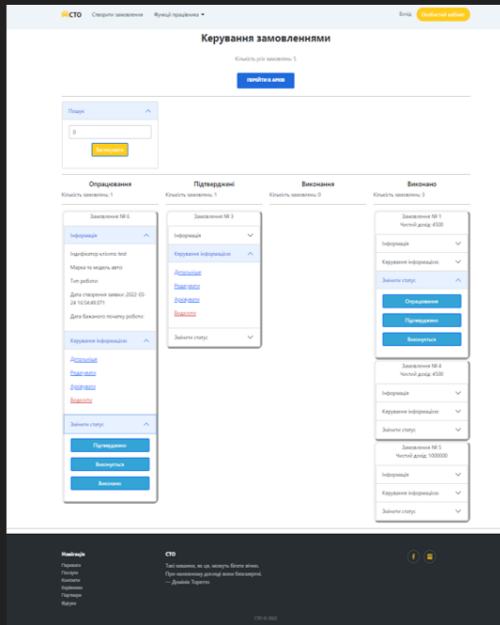


## Сторінка редагування замовлення

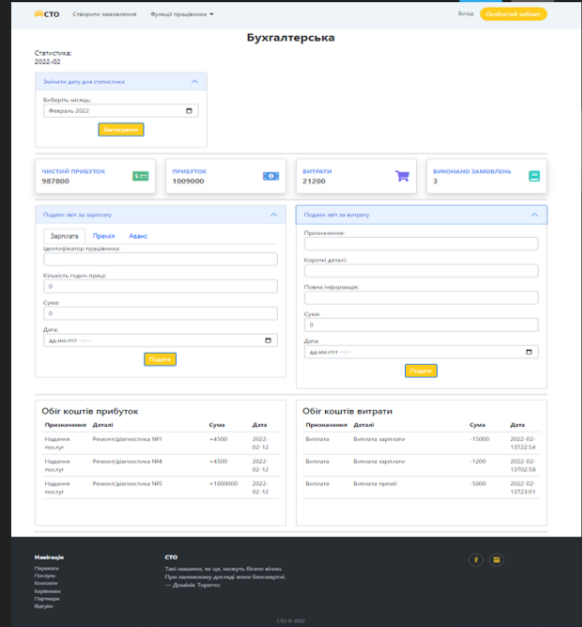


# Результат розробки

## Сторінка CRM

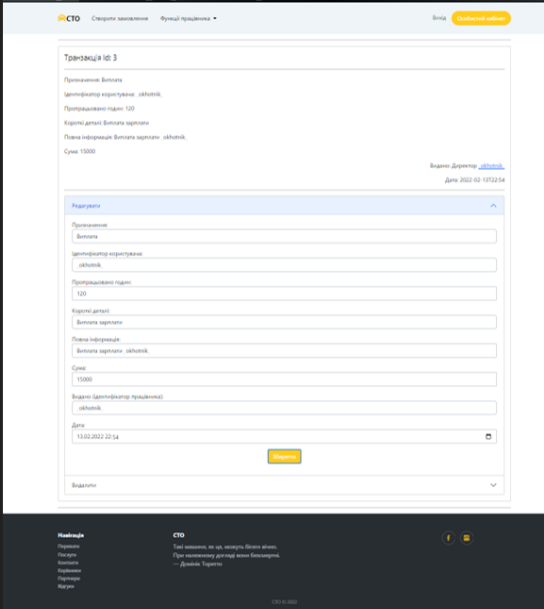


## Сторінка бухгалтерського обліку

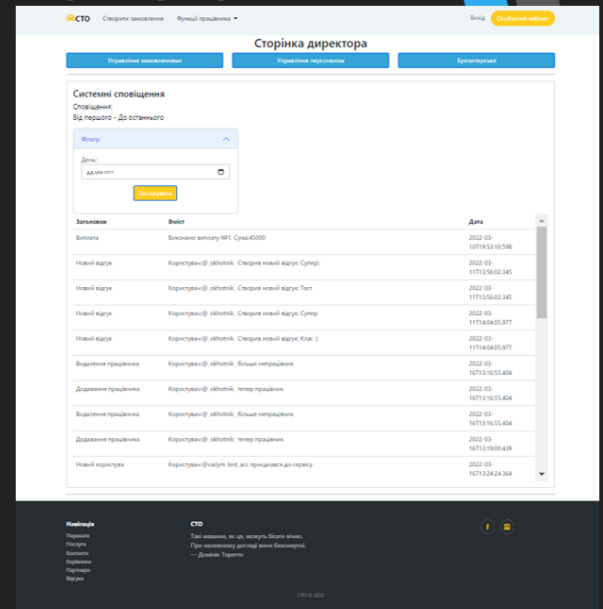


# Результат розробки

## Сторінка інформації транзакції



## Сторінка директора



## Висновки

При виконанні дипломного проекту проведено аналіз предметної області. Аналіз підтвердив актуальність розроблювального програмного забезпечення. Проведено аналіз наявного програмного забезпечення, визначено переваги та недоліки програмних продуктів, що допоможе краще спроектувати розроблюване програмне забезпечення. Згідно з виконаного аналізу розроблено технічне завдання, яке чітко встановлює функціональні вимоги для програмного продукту. Проведено проектування програмного забезпечення, визначено архітектурні підходи та методи реалізації для забезпечення високої продуктивності, оптимізації програмного продукту, спроектовано ергономічний користувацький інтерфейс застосунку. На основі проектування виконано програмну реалізацію програмного продукту. Програмний продукт протестовано та визначено що всі функції застосунку коректно працюють. Готовий програмний продукт впроваджено у системі контейнеризації Docker, та повністю готовий до використання.

Завідувачу кафедри інженерії програмного  
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Охота В. В.

Прізвище, ініціали

факультет ІТ, 4 курс, група ПІЗ-18-1

### ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

25.05.2022  
дата

  
підпис

## Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 3.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 12%

ID: 104064 Назва: Інтернет-платформа із CRM-системою для станцій технічного обслуговування автомобілів Додано в БД: 2022-05-26 Автора: В. В. Охота Керівники: Л.П.Бедратюк Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	72832	1124	4342 (6%)	63 (6%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:  
Кафедра ІПЗ

ID перевірки:  
1011345634

Дата перевірки:  
26.05.2022 19:42:06 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
26.05.2022 19:44:14 EEST

ID користувача:  
100005589

Назва документа: Дипломний\_проект\_записка\_Охота-1-97

Кількість сторінок: 99 Кількість слів: 13205 Кількість символів: 110231 Розмір файлу: 4.05 MB ID файлу: 1011231415

## 8.21% Схожість

Найбільша схожість: 3.93% з джерелом з Бібліотеки (ID файлу: 1008265198)

3.62% Джерела з Інтернету 224 ..... Сторінка 101

5.62% Джерела з Бібліотеки 102 ..... Сторінка 103

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 1

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ  
освітнього ступеня «Бакалавр»

Дипломник Охота Вадим Володимирович

Тема Інтернет-платформа із CRM-системою для станцій технічного обслуговування автомобілів

Спеціальність 121 – Інженерія програмного забезпечення

**Обсяг дипломного проекту:**

Кількість листів креслень \_\_\_\_\_; кількість сторінок записки \_\_\_\_\_

1. Короткий зміст пояснювальної записки та прийнятих рішень У дипломній роботі  
Проведено аналіз предметної області, проведено аналіз схожого програмного  
забезпечення визначено їх переваги та недоліки. За результатами аналізу розроблено  
архітектуру програмного забезпечення, визначені функціональні вимоги та  
та розроблено дизайн інтерфейсу і модулі програмного забезпечення. Виконана  
програмна реалізація та тестування програмного забезпечення та його впровадження  
у засобі контейнеризації Docker

2. Висновок про відповідність проекту поставленому завданню Дипломна робота  
освітнього ступеня «бакалавр» у повній мірі відповідає поставленому завданню як у  
теоретичній, так і в практичній її частині.

3. Характеристика виконання кожного розділу проекту, ступінь використання  
останніх досягнень науки і техніки та передових методів роботи У вступі описується  
актуальність дипломного проекту. У першому розділі аналізується предметна область  
та подібне програмне забезпечення, описуються вимоги для програмного  
забезпечення. У другому розділі аналізується та вибирається архітектура для  
програмного забезпечення. Вибираються підходи для створення безпеки програмного  
забезпечення, описуються модулі програми та створюється дизайн інтерфейсу  
користувача. У третьому розділі проведені етапи реалізації програмної системи. У  
четвертому розділі виконано інтеграційне тестування застосунку, та підтверджено  
його коректу роботу.

4. Позитивні сторони проекту Дипломний проект містить детальний аналіз роботи.  
Розроблений застосунок має практичний та зручний інтерфейс користувача. Для  
реалізації програмного забезпечення застосовано новітні технології та практики  
програмування. Програмне забезпечення містить усі потрібні функції щоб повністю  
задовільнити тему роботи.

---

---

---

---

---

---

---

---

---

---

5. Негативні сторони проекту Для надсилання форм звернення було б доцільніше використовувати динамічні методи надсилання запиту та отримання відповіді, що у свою чергу дозволило б не оновляти цілком веб-сторінку.

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконано відповідно до теми дипломного проекту. В пояснювальній записці дотримано усіх вимог та стандартів.

7. Відгук про дипломний проект в цілому Загалом дипломний проект повноцінно описує вирішення завдання. Матеріал дипломної роботи чіткий та структурований, у повній мірі розкриває тему, всі розділи логічні та повноцінно описані, що дозволяє зрозуміти весь проект. Графічний матеріал дає можливість наочно зобразити доцільність та ефективність рішень для вирішення задачі.

8. Інші зауваження \_\_\_\_\_

9. Оцінка дипломного проекту Розглянувши позитивні та негативні сторони дипломного проекту можна зробити висновок, що дипломний проект виконано в повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «відмінно».

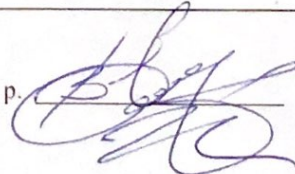
РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) \_\_\_\_\_

Чешун Віктор Миколайович, кандидат технічних наук, доцент кафедри КБ, Хмельницький національний університет

.. 30 ..  
(підпис)

*травня*

2022 р.



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Інтернет-платформа із CRM системою для станцій технічного обслуговування автомобілів»

Автор: Охота Вадим Володимирович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Бедратюк Леонід Петрович, д-р фіз.-мат. наук, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідальність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить лавинисті текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

**Підтвердження:**

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) у тексті дипломного проекту системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання на проектування, відомість документів, у структурі змісту, написах в рамках, назвах розділів/підрозділів тощо) та в назвах переліку джерел посилання;
- 2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 3) усі запозичення є фрагментарними або мають належним чином оформлені посилання;
- 4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлених збігів ідентичності схожості, складає 8.21% і адресується до 224 джерел, що з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломного проекту

Керівник



Л. П. Бедратюк

Гарант ОП



Л. П. Бедратюк

Завідувач кафедри



Л. П. Бедратюк