

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

**КВАЛІФІКАЦІЙНА РОБОТА**

Трохимчук Ольги Вікторівни

на здобуття ступеня вищої освіти Бакалавра (бакалаврський)

Сайт портфоліо вебстудії

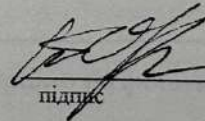
Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

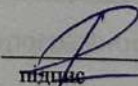
Шифр КвРПЗ 210104.01.05 ПЗ

Виконав студент III курсу, група ПЗс-21-1

  
підпис

О.В. Трохимчук  
Ініціали, прізвище

Керівник канд. пед. наук, доцент

  
підпис

О.Г. Онишко  
Ініціали, прізвище

Нормоконтролер канд. пед. наук, доцент

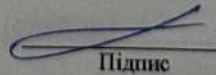
  
підпис

Н.І. Праворська  
Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії

програмного забезпечення

  
Підпис

Л. П. Бедратюк  
Ініціали, прізвище

7 червня 2024 р.

Хмельницький 2024

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  
Завідувач кафедри 173  
Л. П. Бедратюк  
06 01 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Трохимчук Ользі Вікторівні

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Сайт портфоліо вебстудії

Керівник проєкту (роботи) Онишко Оксана Григорівна, канд. пед. наук, доцент

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

Затверджена наказом ректора університету від 08.01.2024 р. № 6-КП

2. Строк подання студентом проєкту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проєкту (роботи) Матеріали переддипломної практики, Методичні вказівки до виконання кваліфікаційної роботи



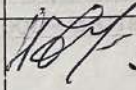
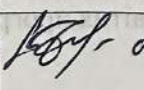
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі, проєктування програмного забезпечення, програмна реалізація системи, тестування програмної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проекту (роботи)

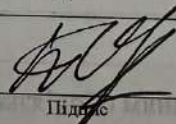
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Праворська Н.І., канд. пед. наук, доцент		
Антиплагіат	Форкун Ю.В., доцент кафедри ІІЗ	 30.05.24	 06.06.24

7. Дата видачі завдання « 05 » лютого 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1 Ознайомлення з тематикою дипломного проектування (ДП), визначення та узгодження індивідуальних там ДП	05.02 – 14.02.2024	
2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання	15.02 – 28.02.2024	
3 Проектування програмного забезпечення	01.03 – 30.03.2024	
4 Програмна реалізація	01.04 – 20.04.2024	
5 Тестування програмного забезпечення	21.04 – 30.04.2024	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2024	
7 Попередній захист ДП	Травень 2024 (згідно графіка)	
8 Перевірка ДП на плагіат, нормоконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки	26.05 – 30.05.2024	
9 Підготовка до захисту та захист ДП	з 01.06.2024	

Студент

  
Підпис

О.В. Трохимчук  
Ініціали, прізвище

Керівник проекту (роботи)

  
Підпис

Олександр С. Г.  
Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Сайт портфоліо вебстудії»

Автор роботи: Трохимчук Ольга Вікторівна

Керівник роботи: Онишко Оксана Григорівна

Пояснювальна записка: 56 с., 13 рис., 3 табл., 7 дод., 52 джерела

Графічна частина: 18 презентаційних слайдів

Ключові слова: AF, BR, CSS, SQL, MYSQL, MSSQL, DBMS, HTML, JS, MVC, QS, QA, SATC, TC, TCS, TCP, TP, TD

Мета кваліфікаційної роботи: розробка архітектури, компонентів сайту програмного застосунку для подання портфоліо про роботи та інформації про співробітників веб-студії.

Практичне значення нашої роботи полягає у автоматизації роботи офісу веб дизайнерів, завдання якої полягає у наповненні та обробці в інформаційній системі даних співробітників та їх роботи. Система дозволяє максимально спростити виконання основних операцій над такими об'єктами як: редагування, видалення, оновлення та створення нових елементів.

Основна перевага даної роботи полягає у гнучкості її використання та налаштування, що надає можливість досить ефективно та максимально якісно її використовувати на будь-яких типах пристроїв.

## ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.210104.01.05.ПЗ	Пояснювальна записка	58		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
	A3		<u>Графічні документи</u>	3		

				КвРІПЗ. 210104.01.05.ВЦ			
Зміг.	Арк.	№ докум.	Підпис	Дата	Лім.	Арк.	Аркуші
Виконав		Трохимчук О.В.				1	1
Керієник		Онишко О.Г.					
Рецензент							
Н. Контр.		Праворська Н.І.					
Зав. каф.		Бедратюк Л.П.					
Сайт портфоліо вебстудії Відомість документів					ХНУ, ІПЗс-21-1		

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	7
ВСТУП.....	9
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	13
1.1 Аналіз та визначення видів, методів та змісту інформаційного забезпечення предметної області.....	13
1.2 Аналіз досліджень, засобів розробки та існуючих рішень.....	12
1.3 Постановка задачі та перелік задач для реалізації.....	17
1.4 Висновок першого розділу.....	18
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ.....	19
2.1 Вибір типу архітектури та зразків проектування.....	19
2.2 Опис декомпозиції.....	24
2.3 Опис залежностей.....	26
2.4 Опис інтерфейсів.....	28
2.5 Детальне проектування та розробка модулів застосунку модулів.....	29
2.6 Детальне проектування даних.....	32
2.7 Висновок другого розділу.....	35
ПРОГРАМНА РЕАЛІЗАЦІЯ.....	36
3.1 Опис структури даних та моделі бази даних.....	36

КвРПЗ.210104.01.05.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Трохимчук О.В.		
Керівник		Онишко О.Г.		
Рецензент				
Н. Контр.		Праворська Н.І.		
Зав. каф.		Бедратюк Л.П.		
Сайт портфоліо вебстудії				
		Літ.	Арк.	Акрушіє
			5	1
КвРПЗ.210104.01.05.ПЗ ХНУ, ІПЗс-21-1				

3.2 Керівництво користувача.....	49
3.3 Налагодження та тестування веб-сайту.....	52
3.3.1 Функціональне тестування сайту.....	58
3.3.2 Валідація.....	65
3.4 Висновки до третього розділу .....	66
Висновки.....	68
Перелік джерел посилання .....	69
Додаток А .....	70
Додаток Б.....	84
Додаток В .....	87
Додаток Г .....	89
Графічна частина .....	96

					<i>КвРІПЗ.210104.01.05.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Конструктор веб-сайтів – програмна система для розробки веб-сторінок без застосування користувачем мов програмування. Зазвичай конструктор це окремий сервіс, але часто надається хостинговими компаніями, як додатковий сервіс.

Бриф – попередня форма ТЗ – скорочена письмове технічне завдання, яка складається виконавцем та замовником. В брифі коротко прописують головні основні технічні параметри програмного застосунку.

Сайт-візитка – сайт з мінімальними вимогами інженерії програмного забезпечення, що зазвичай містить одну вебсторінку (або декілька) та містить головну інформацію про діяльність фірми, її співробітників, інформацію про компанію, про товари чи послуги компанії, контактні дані тощо.

CMS – система керування змістом;

A/B-тестування – один з методів маркетингового дослідження, основна суть полягає в контролі групі елементів, що порівнюються з набором вибраних тестових груп, де один або декілька показників були замінені, для з'ясування змін, що покращують цільовий показник.

RGB – це спосіб кодування кольорів з допомогою упорядкованого набору числових значень базових компонентів кодування червоного, зеленого та синього кольорів .

PHP – інтерпретована мова програмування, у код якої вбудовується ' html-код веб-сторінок, які коректно оброблюється браузером та PHP-інтерпретатором.

CSV – (від англ. Comma-Separated Values) формат файлів для зберігання табличних даних, де поля даних відокремлюються комою, а записи переходом на новий рядок.

CRM – (від англ. Customer Relationship Management). – система управління відношень з клієнтами – програмне забезпечення для відділів

					<i>КвРІПЗ.210104.01.05.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

продажу.

RSS – компонент (формат), який дозволяє виводити контент веб-сайту зовні у спеціалізованій структурованій формі.

SEO – процес коригування програмного HTML-коду, текстового наповнення (контенту), структури веб-сайту, контроль усіх зовнішніх чинників для відповідності вимогам алгоритму для пошукових систем, з метою підняття позиції веб-сайту в результатах пошуку за певними запитами користувачів. Чим вище позиція веб-сайту в результатах пошуку, тим більша ймовірність того, що відвідувач перейде на сайт з пошукових систем, оскільки люди зазвичай йдуть за першими такими посиланнями.

FTP – програма клієнт для доступу до даних FTP-сервера.

JPEG – графічний растровий формат збереження графіки, з використанням алгоритму стиснення інформації з втратами якості самого зображення.

PNG – графічний растровий формат збереження графіки, з використанням алгоритму стиснення інформації без втрат якості самого зображення.

WebP – веб-орієнтований графічний формат збереження графіки, з використанням алгоритму стиснення інформації з втратами і без втрат якості зображень.

WYSIWYG – властивість спеціальних прикладних програм веб-інтерфейсів, яка дозволяє відображати вміст розроблювального продукту в процесі редагування.

					<i>КвРІПЗ.210104.01.05.ПЗ</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Актуальність нашого дослідження обумовлена цілим рядом факторів. Зокрема, це постійно зростаюча потреба в різних проєктах з різних предметних галузей, які призначені для застосування у процесі при вирішенні проблем конкретної галузі.

Веб розробка та проектування, загалом – це свого роду науково-практична діяльність, що націлена на вирішення актуальних та перспективних соціальних проблем та розробку оптимальних варіантів їх розв’язання. Це, також, конструювання особою, деякою групою чи організацією дій, які спрямовані на досягнення конкретної соціально значимої мети, локалізованої за визначеним місцем, конкретним часом і визначеними ресурсами. Сама сутність соціального проектування полягає зокрема у проектуванні бажаних станів продукту. Для забезпечення якісної та комфортної роботи користувача з такими проєктами необхідно досить виважено підходити до їх проектування та розробки, забезпечуючи при цьому усі необхідні умови для досягнення максимально ефективного і комфортного опрацювання потрібних даних.

Виходячи з огляду на невинне та постійне зростання популярності різноманітних веб-видань, зростає нагальна потреба підвищення статусу таких проєктів та пошуку шляхів для ефективного обробки інформації, зацікавленості споживачів таких продуктів та привертання їхньої уваги до продукту.

Таким чином, виникає потреба в дослідженні теоретичних та практичних аспектів проектування, конструювання, макетування та практичній реалізації подібних проєктів, зокрема також їх різновиду, таких як сайт-візитівка , сайт-портфоліо та обґрунтування умов їхнього ефективного використання в подальшій своїй роботі.

					КвРІПЗ. 210104.01.05	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		



знаходити нові нестандартні рішення та застосовувати їх в своїх проектах в, враховуючи всі сучасні переваги та усувати недоліки, які були виявлені на схожих проектах.

Як зазначалось вище, сайт портфоліо відіграє роль стислої презентації. Різноманітність таких сайтів досить значна – для тематичних робіт, видів послуг, окремих людей, групи осіб, фірми, компанії та відображати корисну інформацію про них загалом.

Сайт-портфоліо дозволяє в стислій, зручній формі надати потенційному клієнту загальне уявлення про роботи, послуги, товари. Тобто це звичайне портфоліо яке постійно доступне в мережі інтернет для усіх. Відповідно, такий сайт має привертати увагу, містити не тільки стандартні елементи, але й бути яскравим для залучення потенційних клієнтів і відповідно доходу.

Отже, сайт-портфоліо є дуже важливою складовою для ведення успішного бізнесу. Такі сайти потрібні усім підприємцям без винятку, так і можуть бути для окремих людей різних спеціальностей, не тільки для програмістів, веб-дизайнерів, кіно- та фото-операторів. На таких сайтах-візитках можна викладати свої послуги, роботи та інше.

Кошти, вкладені в сайт-портфоліо досить швидко оправдовують себе, та починають приносити замовнику йому дохід, адже ця непряма реклама швидко починає залучати клієнтів, які зацікавлені в тих послугах, що наведені на самому сайті. Якісно зроблений сайт-портфоліо швидко притягує потенційних клієнтів, оскільки інформація на ньому дає суттєве уявлення про необхідні потреби клієнта та його вподобання.

Отже, серед основних переваг сайту-портфоліо, окрім того, що замовники отримують масу переваг перед своїми конкурентами, це:

- доступний кожному користувачу;
- зручний і простий пошук його в мережі Інтернет;
- надає можливість розміщення реклами;
- надає можливість працювати напряму з клієнтом без посередників.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						12
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Сайт-портфоліо досить сильно необхідний фізичним особам підприємцям, які за незначні кошти та засилля, можуть просувати на ринок свої послуги та продукти, що досить вигідно їх виділить серед конкурентів. Так сам сайт знайти клієнтів і стати помічником у просуванні товарів та послуг.

Об'єктом дослідження у данму проекті є процес організації обміну, збереження та керування інформацією про діяльність веб-студії.

Метою роботи є вибір методів та засобів розробки та створення сайту портфоліо веб-студії. Також метою і призначенням цього проекту є реалізація певного середовища, яке буде здійснювати автоматизацію процесів для організації роботи сайту веб-студії, а також реалізація самого інтерфейсу сайту, яке в сукупності має забезпечити можливість ефективної, зручної та простої роботи з інформацією про діяльність веб-студії.

Практичне значення роботи полягає в тому, що спроектований нами макет сайту-портфоліо може використовуватися в процесі подачі інформації на поліграфістами, програмістами та графічними дизайнерами в своїй професійній діяльності, а також компанією, яка займається підтримкою, що допомагає вдосконалити роботу в даній сфері за допомогою таких скринінгів.

Загалом задачі кваліфікаційної роботи є такими:

- дослідити понятійний апарат даної теми;
- визначити особливості сайту-портфоліо, як різновиду тематичного сайту;
- здійснити аналіз та вибір ресурсів за визначеною темою;
- визначити етапи конструювання та проектування сайту-портфоліо;
- визначити апаратно-програмне забезпечення для реалізації сайту;
- спроектувати дизайн сайту;
- реалізувати програмно сайт портфоліо веб-студії.

Теоретичне обґрунтування обраної теми кваліфікаційної роботи полягає в подальшому використанні та запропонованих опрацьованих матеріалів з результатами теоретичного та практичного дослідження для вдосконалення їх і подальшого використання дизайнерами та розробниками про проектуванні сайтів

					КвРІПЗ. 210104.01.05	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

портфоліо веб-студій.

Використанні теоретичні методи дослідження, що були застосовані нами під час виконання кваліфікаційної роботи, це: методи порівняння, аналізу, синтезу та узагальнення.

Практичні методи розробки, які використано нами під час виконання кваліфікаційної роботи, полягають в конструюванні, проектуванні, макетуванні і та розробки для створення макету сайту.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						14
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



Для роботи сайту потрібно набір взаємопов'язаних елементів, таких як унікальне ім'я, сервер на якому він буде працювати, система управління контентом тощо.

По-перше, файли сайту мають зберігатися і оброблятися на певному ресурсі – сервері або, як мінімум хостингу. Сервер виконує цю функцію. Це спеціальні комп'ютери, які забезпечують веб-проекти пам'яттю на високошвидкісних носіях, оперативною пам'яттю та іншими технічними ресурсами. Один сервер може обробляти декілька сайтів. Простір і обчислювальні ресурси, що виділяються для оренди, називаються хостингом.

Сайт має мати доменне ім'я - це адреса сайту, наприклад: site.miy.com nazva-saitu.ua. Доменні імена можуть бути розміщені в зонах різного рівня і тематики. Це слід враховувати при просуванні сайтів в пошукових системах. Вартість адрес високого рівня є найвищою. У деяких зонах неможливо купити домен-це, наприклад, публічний домен державних установ .gov.

Сайт має керуватись не тільки фізично, але й програмно. Для цього служить система управління контентом (CMS). В даний час системні файли для веб-проектів зазвичай не створюються вручну, в веб-розробники використовують CMS-спеціальний рушій, який включає в себе основні функції сайту. Найбільш популярними системами управління контентом є Opencart, Wordpress, Squarespace, Prestashop, Joomla. Сама cms та база даних, щомістить фотографії, текст та інший вміст, зберігаються на сервері.

Таким чином, можна зробити висновок, що Інтернет-сайт-це CMS, розміщена на сервері і має файл з унікальним доменним ім'ям.

Будь-який веб-ресурс створюється для визначеного кола завдань. За цим критерієм їх поділяють на інформаційні, комерційні та некомерційні.

Інформаційні сайти створюють для надання певної інформації своїм клієнтам. В свою чергу їх поділяють на новинні, розважальні, за певною тематикою, тематично-популярні, наукові, окремих спільнот тощо. Зазвичай такі

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

сайти створюють з метою їх подальшої монетизації, включаючи в них пряму рекламу та посилання на інші ресурси.

Сайти комерційного напрямку створюють для отримання прибутку від продажу послуг чи товарів, для створення позитивного образу організації чи підприємства та подальшого розширення своєї аудиторії. Класичним прикладом комерційного сайту є інтернет-магазин. В ньому всі елементи створені для продажу та для стимулювання клієнтів до їх купівлі.

До веб-ресурсів некомерційного призначення відносять різні онлайн-платформи державних установ, начальних та медичних закладів, соціальних служб тощо. Метою їх створення є в основному інформування своєї аудиторії з визначених питань.

Розрізняють також класифікацію сайтів за їх структурно-функціональним призначенням.

Для того щоб сайт успішно справлявся зі своїм завданням, необхідно правильно підібрати структурне та функціональне наповнення. Так, використовувати CMS, призначених для створення інтернет-магазинів, в створенні інформаційних та розважальних проектів немає сенсу. У зв'язку з цим виділяють такі типи веб-ресурсів:

– цільова сторінка (лендінг) – це односторінковий сайт з потужною рекламою. Його завдання – привернути увагу відвідувача сайту і підвести його до задуманих дій щодо зацікавленості в пропонованому продукті;

– сайт-візитівка – невеликий ресурс з простою доступною структурою. Подібні проекти створюються для інформування відвідувачів сайту про діяльність компаній і окремих підприємців та фахівців;

– особистий блог – цей тип сайту присвячений конкретній людині або її діяльності. Блог може бути цікавим, біографічним, науковим, інформативним тощо;

– сайти-сервіси, їх структура і функціональність – це ресурси, які дозволяють компаніям зручно і ефективно надавати свої послуги;

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						17
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

– Інтернет-магазин – це особливий формат комерційного сайту, який дозволяє зручно вибирати і замовляти послуги та товари. У його структуру зазвичай входять товари, послуги з описом та поданням, фільтр, форми зворотного зв'язку, кнопки «перегляд», «замовлення», «купити», «кошик» тощо. Крім того такі сайти містять адміністративну частину для власника інтернет магазину.

До інших категорій слід віднести такі типи сайтів, як соціальні мережі, поштові сервіси, різноманітні пошукові системи, хмарні сховища, хмарні обчислення, відеохостинги та файлообмінники. Крім того, ресурси, як правило, великі, і їх розробка, функціонування та підтримка вимагають значних фінансових вкладень.

В нашому випадку ми обрали сайт портфоліо. Даний тип сайту досить схожий з типом сайтів візиток за своїм призначенням та функціональністю.

Сайт візитка чи сайт-портфоліо – це сайт організації чи певного приватного підприємця з оригінальним дизайном, зручною CMS, та з можливістю автоматичної синхронізації з корпоративним сайтом.

Сайти-портфоліо призначені для залучення нових клієнтів та партнерів, забезпечення інформації про зворотній зв'язок, проведення ефективної реклами та зниження витрат на комунікацію і презентацію послуг, робіт, товарів тощо. Завдяки сайту-портфоліо позиції компанії в конкурентній боротьбі зміцнюються.

Сайт-портфоліо – найкращий вибір для усіх, хто планує організувати продаж товарів та послуг в мережі інтернет, не залучаючи значних коштів та інших ємнісних ресурсів.

Сайт-портфоліо умовно розділяється на незалежні функціональні частини:

- набір веб-сторінок;
- новини, огляди, акції;
- каталог послуг, товарів;
- прайслисти;
- пошук;
- модулі авторизації та реєстрація користувачів;

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						18
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

– модулі синхронізації бази даних тощо.

Серед замовників сайтів-портфоліо можна зустріти не тільки людей приватних підприємців, представників малого, середнього та великого бізнесу, такі як різні туристичні агентства, салони краси, приватні клініки, а й також великі підприємства та організації, які самі не потребують у розміщенні і частому оновленні великих обсягів даних.

Суть в тому, що з точки зору реклами сайт-портфоліо не менш ефективний серед інших видів веб-ресурсів, але його реалізація обійдеться замовнику набагато дешевше й адмініструвати досить легко.

Отже, сайт-портфоліо – це невеликий сайт, який складається з декількох веб-сторінок (від однієї до десятка) та містить лише основну інформацію про фірму чи особу: її назва, рід діяльності, контактні дані, дані про послуги тощо.

Головна відмінність сайту-портфоліо від інших веб-ресурсів полягає в її призначенні, а не в самому її форматі, адже його призначення в тому, щоб дати його відвідувачу усю необхідну інформацію про фірму та її пропозиціях і послугах.

Таким чином, стандартні розділи сайту-портфоліо будуть такими:

Інформація про саму веб-студію. Тут може бути відображена історія її розвитку, її головні успіхи і досягнення, інформація про найбільш значимих клієнтів чи її партнерів тощо

Каталог. Даний розділ містить у собі перелік пропозицій та послуг, які пропонує студія, із зазначенням їх ціни та основних характеристик, але самої без можливості оформлення конкретного замовлення безпосередньо на сайті

Контакти. Крім стандартної вказівки номерів телефонів, адреси та електронної пошти сайти-портфоліо часто доповнюють різними функціональними модулями, такі як контактна форма або інтерактивна карта проїзду

Портфоліо. Список успішних послуг та проектів з переліком та детальним описом виконаних робіт, може бути виконаний у формі сайту фотогалереї з великою кількістю фото та картинок

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						19
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



розглянуті нами варіанти мають значні недоліки, як в побудові самої семантики сайту так і в самому інтерфейсі для користувача. Головним з недоліків усіх розглянутих нами рішень виступає досить однотипний та несучасний спосіб відображення самої інформації, зокрема на сучасних мобільних пристроях.

Тому, можливим рішенням для задоволення наших потреб, які подано вище, є розробка нового веб-сайту портфоліо веб-студії, який буде по'єднувати у собі усі заявлені нами функціональні можливості, серед яких:

- зручний перегляд інформації на одній сторінці;
- збереження даних;
- організація даних;
- швидкий та зручний пошук інформації;
- зручний інтерфейс для користувачів сайту;
- тісну інтеграцію з базою даних;
- зручний функціонал для редагування інформації;
- зручний функціонал для передачі інформації тощо.

Загалом, Наш проект буде реалізований в вигляді веб-застосунку на мові програмування html, css, javascript та фрейморка React, і Noode.js саме для того, щоб не обмежувати наш застосунок в певній операційній системі. База даних буде створена в СКБД MySQL, яка є найбільш прийнятною та швидкою. Для розробки цього проекту нами була обрана клієнт-серверна архітектура, де у ролі клієнтської частини виступає веб-браузер, а серверною частиною є веб-сервер, щоб міг бути доступ до цього сервісу в будь-який момент часу та, відповідно, без встановлення додаткового програмного забезпечення.

### 1.3 Постановка задачі та перелік задач для реалізації

Метою розробки даного проекту є створення веб-сайту портфоліо вебстудії.

					КвРІПЗ. 210104.01.05	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

У нашому застосунку мають бути передбачені режим адміністратора та режим користувача. Адміністратор має заносити детальну інформацію про персонал, дизайнерські роботи, готові рішення, клієнтів..

Завданням проекту є також створення простого інтерфейсу для дій введення та виведення даних їх редагування, збереження, обробки та аналізу у базі даних.

В проекті необхідно передбачити такі дії та ресурси:

- авторизацію користувача адміністратора та користувача в системі;
- інтуїтивно зрозумілий інтерфейс застосунку;
- сторінка для адміністрування веб-сайту;
- пошукову система, за наперед заданими нами фільтрами;
- зручне подання та масштабування інформації та її редагування;
- перегляд та редагування даних в базі даних;
- формування різноманітних звітів системи в залежності від наявних отриманих результатів.

Також було розроблене технічне завдання, яке можна використовувати, як основу для нашої подальшої розробки веб-застосунку для управління поданою інформацією.

Технічне завдання – це документ, в якому зафіксовані основні вимоги до застосунків, які повинні бути реалізовані в ході проектування та створення сайту або програмного застосунку.

Технічне завдання розробляється на основі особистої бесіди розробника із замовником, що пред'являються їм вимог до самого проекту та заповнення відповідного брифу.

Бриф – це коротка письмова інструкція, яка складається виконавцем та замовником. В брифі, попередній формі замовлення, коротко прописуються усі основні технічні параметри застосунку. Остаточні характеристики застосунку відображаються в самому технічному завданні.

Далі, на підставі параметрів, які прописані в брифі, відбувається складання технічного завдання.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						22
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



- порядок прийому виконаного завдання.

Технічне завдання – це невід'ємний інструмент для комунікації між замовником і виконавцем, так як він дозволяє обом сторонам:

- представити майбутній програмний засіб;
- перевірити його більш точно, по всіх пунктах;
- не допустити помилок, та знизити їх кількість.

Існує загалом така думка, що ТЗ робиться в основному для замовників, а не для розробників. Розробник, який професійно займається створенням сайтів або розробкою програмного забезпечення, прекрасно розуміє всю свою послідовність і тривалість ходу роботи над проектом. Таким чином технічне завдання – продовження принципу: клієнт завжди правий.

Верстка сайту– це процес створення веб-сторінок за готовим шаблоном, розробленим в різних спеціальних графічних редакторах, за допомогою різноманітних мов програмування HTML, CSS, JavaScript.

Іноді розробка здійснюється і без шаблонів, тобто дизайн створюється «на ходу», але це тільки в тому випадку, коли графічний дизайнер і верстальник є однією людиною. У більшості випадків дизайн самого сайту розробляється окремим спеціалістом.

Створення дизайну здійснюється в таких програмах:

- Corel Draw;
- Adobe Photoshop;
- GIMP;
- Кгі тощо.

Сама ж верстка сайтів виконується на різних мовах:

- HTML;
- CSS;
- PHP;
- JavaScript;
- Python.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Іноді сайти створюються за допомогою спеціальних програм конструкторів, де знання мов програмування не потрібно. Сторінки, створені за їх допомогою, з технічної точки зору, та їх код виглядають схоже. Просто в цьому випадку розробнику замість суєти з кодом пропонують створювати різні елементи, вписувати назви різних заголовків та текст, регулювати різні елементи, що відповідають за певні параметри тощо.

Дизайн сайту-портфолію. З точки зору оформлення сайту необхідно пам'ятати про те, що сайт-портфолію має містити елементи корпоративного стилю фірми, зокрема в назві компанії та її логотипі. Однак, з точки зору стилістики обмежень для цього немає: сайт-портфолію буде однаково добре виглядати, як у класичному стилі, так і в різних стилях арт-деко чи хай-тек.

У питанні адміністрування сайту краще вдається до системи управління контентом CMS, а не використовувати простий html-код. Система управління контенту дозволяє швидко і досить якісно проводити оновлення усього вмісту сайту, додавати та редагувати інформацію, а також редагувати застарілий і неактуальний контент. Важливо також пам'ятати про те, що навіть якщо студії не потрібно занадто часто оновлення даних та інформації, робити це все ж необхідно, тому що неоновлюваний сайт виглядає застарілим і викликає серйозні підозри у всіх потенційних клієнтів.

База даних сервісу сайту здійснює постійне оновлення, адже постійно в вебстудії приймаються багато різних замовлень послуг, тому необхідно, також вести облік значної кількості послуг та клієнтів.

#### 1.4. Висновок першого розділу

У першому розділі нами було проведено глибокий аналіз предметної області. Здійснено детальний аналіз та визначено види, методи та зміст обраної предметної області. Здійснено постановку задачі та окреслено загальний перелік задач для проектування та реалізації.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						25
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Зокрема, в розділі проведено детальний аналіз предметної області, та доведено, що сайт-портфоліо має велику кількість беззаперечних переваг над іншими типами такого роду рішень, серед яких необхідно і доцільно відзначити, що він є потужним інструментом для реклами та просування. Окрім просування сайту безпосередньо в веб-просторі, його також можна подавати в рекламі, взитках співробітників та при листуванні і переписці в месенджерах. Тобто, потенційні клієнти таким чином не будуть втрачені, адже зайти на сайт та отримати необхідну їм інформацію вони зможуть в зручний для час, а не в робочий час веб-студії.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						26
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ

### 2.1 Вибір типу архітектури

Як було вказано попередньо, в даній системі буде використана архітектура «клієнт-сервер».

Дана технологія «клієнт-сервер» широко застосовується при роботі з різноманітними СКБД та базами даних в мережі. Ця технологія відома вже досить давно і найчастіше застосовувалась у малих та великих компаніях. Сьогодні, з розвитком мережі Internet, дана технологія найчастіше приваблює погляди різних проектувальників та розробників програмного забезпечення, оскільки в світі відбувається нагромадження величезної кількості інформації та даних з різних питань, яка зберігається в різноманітних базах даних. Технологію клієнт-сервер загалом можна описати таким алгоритмом:

- на стороні клієнта формується та відправляється новий запит до бази даних серверу БД, а точніше до застосунку, який обробляє ці запити;
- сервер здійснює маніпуляції з інформацією бази даних, яка розташована на сервері, та у відповідності до запиту та формує певний результат, який передає його клієнту;
- далі клієнт отримує цей результат та відображає його на пристрої та очікує подальших дій від користувача. Даний цикл повторюється, поки користувач не завершить роботу з застосунком.

Перевага архітектури клієнт-сервер полягає у наступному:

- відсутності дублювання програмного коду функціоналу сервера у клієнтських програмах;
- низький рівень вимог до технічного обладнання клієнтського обладнання, через те, що всі маніпуляції та дії відбуваються на сервері;
- всі дані, що зберігаються на самому сервері, який як правило, має набагато кращий захист ніж більшість клієнтських терміналів.

Для проектування структури роботи застосунку нами було обрано патерн

										Арк.
										27
Змн.	Арк.	№ докум.	Підпис	Дата						

КвРІПЗ. 210104.01.05



(MVC)) робить поділ складових програмного забезпечення на три різних складових. У цій трійці до обов'язків складової «Модель» (Model) належать функції частини зберігання інформації та забезпечення інтерфейсу доступу до них самих. Складова «Вигляд» (View) містить код, який відповідальний за надання відображення даних користувачам. Складова «Контролер» (Controller) здійснює управління цими компонентами та отримує різні сигнали через реакції на дії користувача, відповідно передає інформацію про зміни складовій Model. Таке подання внутрішньої структури застосунку в цілому дозволяє розділити модуль на окремі самостійні частини та відповідно розподілити відповідальність та контроль між різними її складовими.

Модель MVC ділить проектоване програмне забезпечення на три незалежні окремі частини: компонент введення та виведення даних, компонент обробки інформації та компонент взаємодії між модулями. Ця модель, як було відмічено вище, встановлює ядро основних даних та головний функціонал для обробки даних. Також компонент Model є незалежним від самого процесу введення та виведення даних. Складова виведення View може містити декілька взаємопов'язаних регіонів, зокрема таблиці, поля для форм де відображаються дані. У функції складової Controller входить відповідальність за моніторинг подій, які виникають в результаті деяких дій користувача. Наприклад, зміна положення курсору, натиснення певної кнопки мишки чи введення даних в поле на формі.

Зареєстровані події кожного модуля передаються в різні запити до застосунку, що спрямовуються складовим Model чи об'єктам, які відповідають за відображення самих даних. Відокремлення Model від View дозволяє використовувати ці складові незалежно для відображення певних даних. Отже, якщо користувач з допомогою складової Controller внесе зміни до складової Model, то інформація, подана одним чи декількома візуальними складовими, автоматично відкоригується згідно до внесених змін.

Стандартне програмне забезпечення, що реалізується за технологією клієнт-

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						29
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

сервер, має прекрасну масштабованість, саме через ефективне використання наявного апаратного та відповідного програмного забезпечення. Також наявна стійкість у її роботі, захист від різних видів несанкціонованого доступу, продуктивність при роботі з великими застосунками та з великими базами даних.

Коректність роботи застосунку залежить від того, де саме знаходиться сервер і яким чином клієнт буде під'єднаний до нього. Користувач на своєму браузері заповнює обрану ним форму та виправляє подальші дії. Браузер користувача при натисненні певної кнопки миші на формі передає дані із цієї заповненої ним форми, або знову ж відображає отримані дані в результаті певної операції. При цьому не важливо, яким чином до мережі під'єднаний клієнт. Він може бути також віддаленим користувачем і з'єднуватися будь-яким способом чи методом. Сам модуль приймає дані, здійснює їх перевірку, а також формує запити до СКБД звідки отримує результуючі дані. На самому віддаленому сервері зберігається база даних, яка змінюється відповідно за запитами клієнта. У цьому режимі роботи застосунку забезпечується досить високий рівень безпеки для даних бази даних, як і від збоїв обладнання, різних сторонніх програм, так і від несанкціонованого доступу інших користувачів. Так забезпечується висока продуктивність роботи застосунку, навантаження на мережу при цьому невисокі, але зростають вимоги до характеристик самого сервера.

При проектуванні ми використали також шаблон проектування “Abstract fabric”. Abstract fabric — це такий шаблон проектування, який дає змогу забезпечити інкапсуляцію окремих його модулів під єдиною схемою, пропускаючи, що є їхня деталізація. Цей шаблон належить до класу твірних шаблонів застосунків.

У таких типових задачах застосування, клієнтський код створює певну реалізацію цієї абстрактної фабрики, а далі використовує самий загальний універсальний інтерфейс цієї фабрики, для створення усіх створених екземплярів та об'єктів, що є частиною схеми застосунку. Тут клієнтський код не може знати, знає, тобто не приймає до уваги, які саме об'єкти він буде отримувати від таких фабрик, оскільки він використовує універсальний інтерфейс для їх створення.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		



написана надана інформація. Якщо все скидане один на одне, а колір шрифту також майже співпадає з фоном сторінки то це буде відштовхувати користувачів від сайту, а не навпаки втримувати чи заволікати. Тому дуже важливою є задача щоб вся інформація, яка розміщена на сайті була відображена сприйнятним та розбірливим шрифтом, як для зручності так і більш продуктивної роботи, що в наслідку приведе для покращення роботи поліклініки. Адже, якщо інтерфейс такого сайту не приваблює користувача, а відштовхує його, то користувач навряд чи щось замовить на ньому.

Існує декілька видів та підходів до побудови сайту. Вони називаються логічними структурами сайту, оскільки назва їхньої структури відповідає їх застосуванням та використанням. Логічні структури сайтів поділяють на лінійну структуру, лінійну структуру з різними альтернативами та варіантами, лінійну структуру з розгалуженнями, деревовидну структуру, а також решітчасту структуру сайту.

При створенні нашого інтернет-застосунку ми використаємо деревовидну структуру для побудови сайту, оскільки вона нам найбільше підходить для реалізації наших завдань, а саме реалізацію простоти, навігації та швидкість користування сайтом.

Деревовидна структура на сьогодні досить популярна, оскільки вона є досить зручною та зрозумілою в користуванні. Деревовидна структура, по суті означає, що при вже вході на головну сторінку сайту користувач опиняється перед дилемою: «куди піти далі та який розділ необхідно обрати». Така логічна структура підходить для реалізації майже всіх типів такого роду сайтів, оскільки одразу чітко видно всі розділи сайту, а тому користувач не втрачаючи часу на пошуки може відкрити та продивитись необхідну йому інформацію. Після переходу у потрібний йому розділ користувач підбирає необхідний йому підрозділ і, далі по ієрархії поки не досягне необхідного йому рівня. Реалізація такої моделі користування веб-сайтом є дуже швидкою та зручною, оскільки користувач майже одразу потрапляє на потрібну йому сторінку.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						32
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Відкривши наш інтернет-застосунок користувач відразу бачить головну сторінку, на якій відображені усі головні елементи: блок реєстрація користувача, новини та останні події, пропоноване меню, де відображаються наявні лікарі, амбулаторії, розділи та підрозділи та пошук по інтернет-сайту. Далі користувач може обирати необхідні йому розділи, або пропоноване меню.

## 2.2 Аналіз вимог до програмного забезпечення, зразки проектування

Важливим моментом у програмуванні будь-яких веб-сайтів є використання готових шаблонів для проектування та його реалізації на подібних сайтах, які уже були створені успішно раніше. Зокрема, таким рішенням цих шаблонів є такі рішення, як патерни. Для нашої задачі є три основних групи таких шаблонів.

Породжуючі шаблони — це шаблони проектування, які абстрагують головний процес інстанціювання застосунку. Вони допомагають зробити веб-сайт незалежним від його способу створення, декомпозиції та представлення на ньому об'єктів.

Шаблон, породжувальних класів, використовує механізм успадкування, саме для варіації створюваних класів. Шаблон, який створює об'єкти у ньому делегує інстанціювання іншому об'єкту застосунку чи класу. Такі шаблони досить важливі у проєктів, коли проєктований застосунок більше залежить від декомпозиції його об'єктів та даних, ніж від спадкування самих його класів.

Таким чином, тут замість прямого кодування програмного коду та програмування фіксованого набору певних його поведінок, визначається певний невеликий набір головних фундаментальних функцій, де за допомогою декомпозиції можна отримувати інші та складніші об'єкти.

Структурні шаблони — це такі шаблони проектування, у яких здійснюється розгляд завдань про те, яким чином із їх класів та об'єктів можна утворити складніші за розмірами інші різноманітні структури.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						33
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Структурні шаблони рівня класу у застосунку використовують спадковість для утворення декомпозицій із інтерфейсів, класів та реалізацій.

Структурні шаблони на рівні об'єкту та класу об'єднують інші об'єкти для одержання нової функціональності застосунку. Додаткова гнучкість застосунку в такому випадку пов'язана з можливістю його зміни композиції певних об'єктів та класів під час їхнього виконання, а це є неприпустимим для такого набору статичної композиції класів.

Шаблони поведінки — це такі шаблони проектування, які пов'язані з алгоритмами, що несуть відповідальність за розподіл обов'язків між об'єктами та класами. Мова у них йде не тільки про певні об'єкти та класи, але й про інші типові способи їхньої взаємодії. Шаблони поведінки, загалом, характеризують складний потік керування процесом, який досить важко контролювати розробнику під час роботи веб-сайту. Тут, головне, основна увага зконцентрована саме не на потоці їх керування, а на їхніх зв'язках між конкретними об'єктами та класами.

Для нашого застосунку варті уваги в застосуванні наступні патерни:

Репозиторій. Дані, які спільно у застосунку використовуються різними підсистемами, тут зберігаються у головній базі даних застосунку, яка доступна усім підсистемам нашого веб-сайту. Патерн репозиторій є пасивним елементом застосунку. У ньому керування ним покладено саме на різні підсистеми. Даний патерн доцільно у застосунку використовувати тоді, коли система обробляє досить великі обсяги інформації, оскільки спільне їх використання в застосунку є досить ефективним, і тут не потрібно передавати різні дані з одної підсистеми в іншу підсистему. Загалом, тут підсистема не повинна знати, де й як використовуються певні дані в інших підсистемах, таким чином зменшується ступінь їх зв'язаності. У подібних застосунках безпека, керування самим доступом, резервне копіювання та відновлення інформації досить централізовані, оскільки вони самі входять у систему управління цим репозиторієм. Однак, у таких системах є певні свої недоліки: усі її підсистеми мають бути досить строго узгоджені зі розробленою структурою репозиторію та, відповідно, його моделлю даних. Змінювати таку

										Арк.
										34
Змн.	Арк.	№ докум.	Підпис	Дата						

КвРІПЗ. 210104.01.05

модель даних у застосунку досить складно та важко. Окрім того, до різних підсистем застосунку висуваються й різні вимоги щодо її безпеки, копіювання та відновлення інформації. У самому патерні «Репозиторій» до всіх його підсистем застосовна однакова політика (Рисунок 2.2).

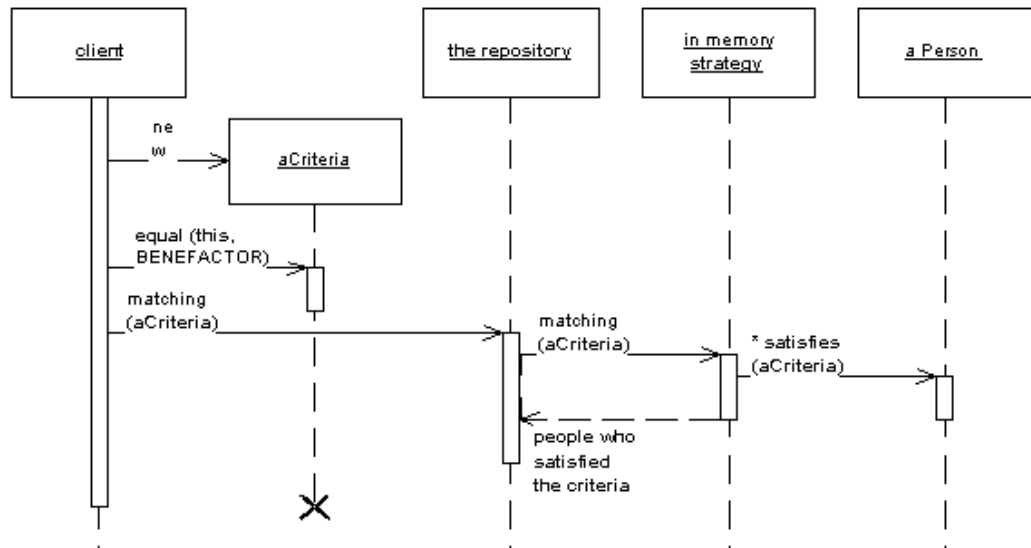


Рисунок 2.2 – Патерн «Репозиторій»

Фабричний метод. Даний вид патерну визначає інтерфейс застосунку для створення об'єктів. Він залишає можливість підкласам вирішувати, який саме клас має для цього інстанціюватись. Фабричний метод надає дозвіл об'єкту та класу делегувати інстанціювання іншим його підкласам. Фабричний метод доцільно використовувати тоді, коли:

- клас не знає заздалегідь, об'єкти яких класів йому буде потрібно створювати для застосунку;
- клас застосунку спроектовано таким чином, щоб різні його об'єкти, які він створює, щоб вони специфікувалися іншими його підкласами;
- у ньому об'єкт чи клас делегує певні свої обов'язки одному з інших допоміжних його об'єктів та підкласів, де необхідно локалізувати інформацію, про те, який дійсно його підклас має приймати на себе ці обов'язки.

Принцип роботи цього патерну «Фабричний метод» наведено нижче на

рисунку 2.3:

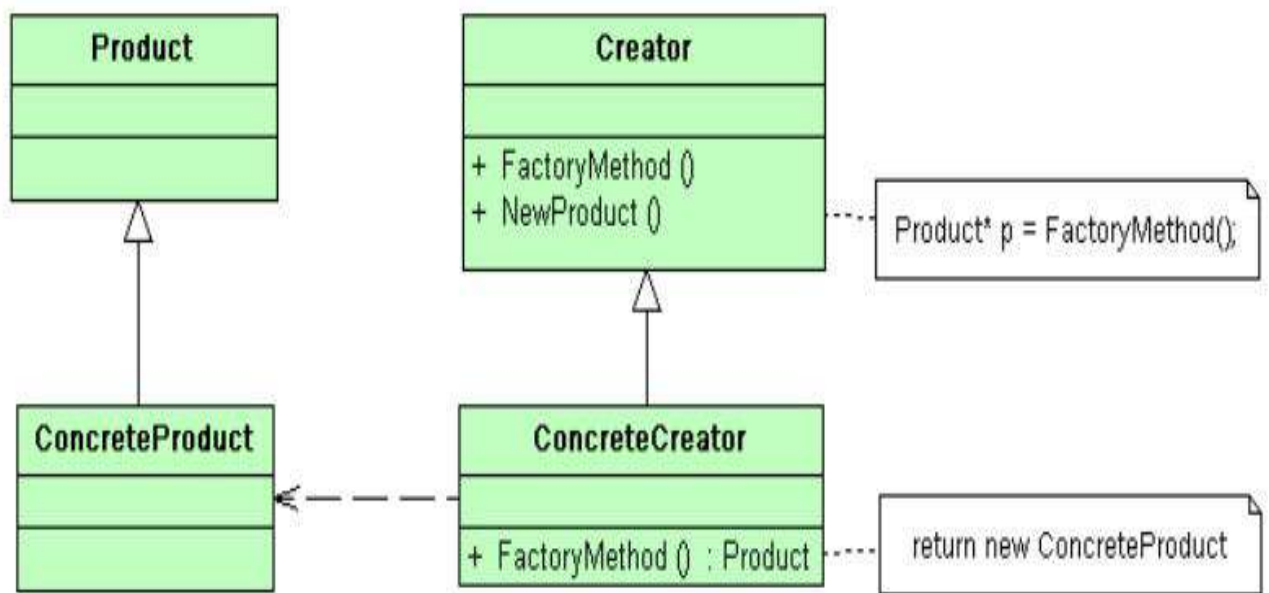


Рисунок 2.3– Патерн «Фабричний метод»

Адаптер — це такий структурний шаблон проектування, який призначений в основному для організації використання різних функцій класу, що недоступний для зміни, але через спеціально для цього створений окремий інтерфейс. Він адаптує інтерфейс одного об'єкту чи класу в інший клас, який очікується клієнтом застосунку. Тобто, він дозволяє включати уже певні існуючі деякі об'єкти в нові створені об'єктні структури, в їх незалежності від відмінностей в інтерфейсах. Шаблон «Адаптер» забезпечує роботу різних об'єктів і класів з несумісними інтерфейсами. Найчастіше його застосовують тоді, коли програмний виріб підтримує потрібні дані та його поведінку, проте має невідповідний для цього якийсь інтерфейс. Патерн «Адаптер» передбачає в собі створення нового класу-оболонки з необхідним саме йому інтерфейсом. Принцип роботи цього патерну показано на рисунку 2.4:

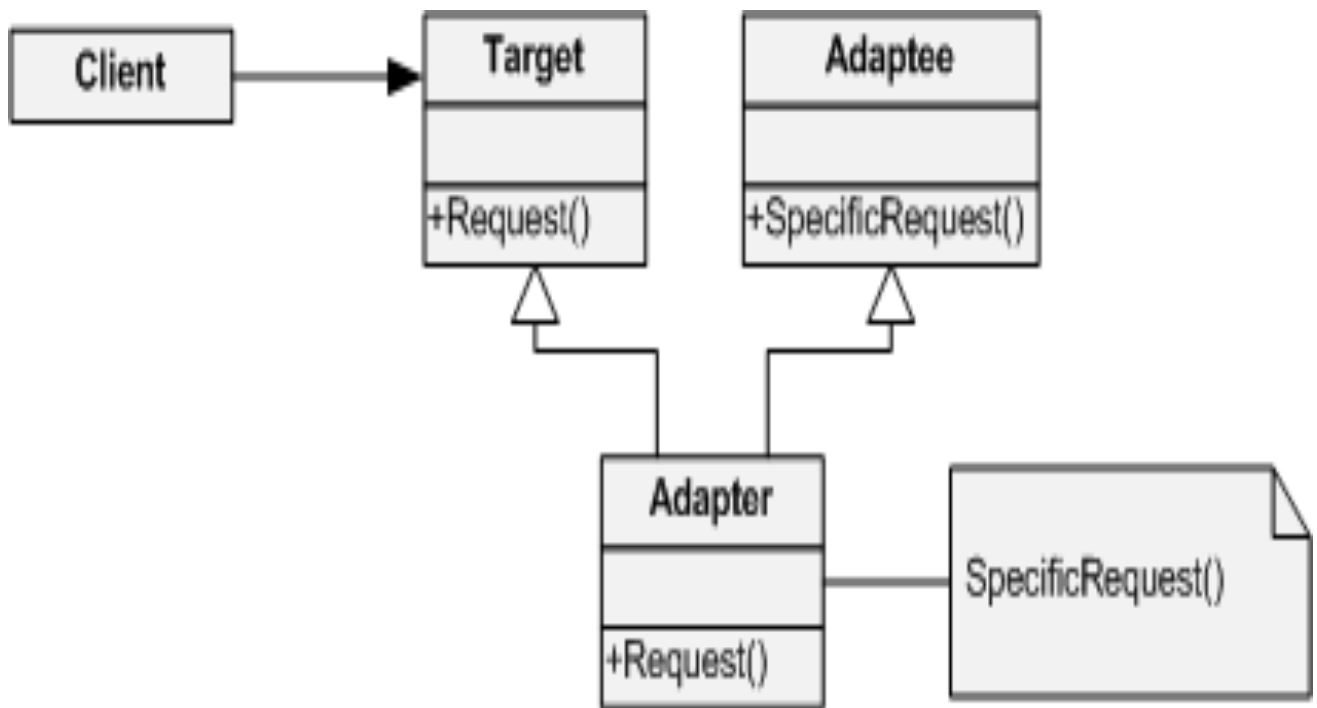


Рисунок 2.4 – Патерн «Адаптер»

Шаблон «Клієнт-сервер». Дані та процеси у таких застосунках розподілені між декількома різними системами. Даний патерн має три основних складові:

- набір окремих серверів, які надають сервіси іншим підсистемам програмного застосунку;
- набір підсистем-клієнтів застосунку, які викликають сервіси, що надаються їм серверами системи безпосереднь;
- мережа. Вона саме служить для доступу запитів від клієнтів до сервісів сервера застосунку. Клієнти, тут, знають імена усіх своїх серверів та усіх необхідних сервісів, а сервери не повинні знати імена своїх клієнтів та їхню кількість загалом. Тут, клієнти отримують певний доступ до серверів за допомогою віддаленого виклику різних процедур та функцій. Такий підхід використовується для реалізації застосунків, які ґрунтуються на репозиторії, що працює, як сервер системи. У ньому усі підсистеми, які мають доступ до його репозиторію саме і є його клієнтами. Загальний принцип роботи цього патерну «Клієнт-сервер» зображено на рисунку 2.5:

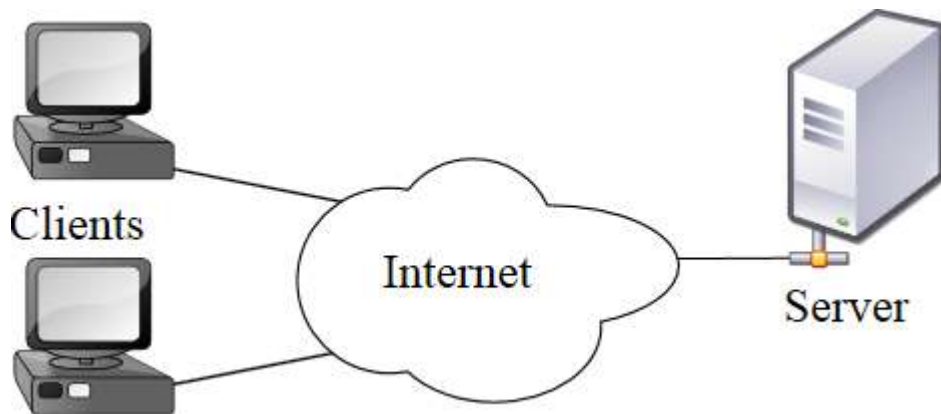


Рисунок 2.5 – Патерн «Клієнт-сервер»

Клієнт-серверний патерн формує спеціальну розподілену архітектуру застосунку. Таку архітектуру доцільно та ефективно використовувати в різноманітних мережних системах, які мають велику кількість розподілених підсистем. У такий застосунок досить легко додати новий сервер та, відповідно, здійснити інтеграцію його з іншими його частинами самої системи, оновити інші його сервіси, та таким чином не впливати на інші його частини. Основні недоліки тут полягають у тому, що при його роботі, сервери та клієнти тут обмінюються різними даними, а це при досить великих обсягах інформації можуть виникати проблеми з великою пропускнуою здатністю та неполадками в мережі.

«Клієнт-сервер» повністю задовольняє вимогам нашого веб-сайту потрфолю, тому його нами буде використано за основу для архітектури нашого проєктованого застосунку.

На основі цього, нами було обрано середовище керування баз даних MySQL, адже середовище розробки Visual Studio Code, мови html, Css та Java. MySQL забезпечить нашому застосунку клієнт-серверну архітектуру, яка необхідна для підтримки значної кількості потенційних користувачів. Таким чином під час проєктування та розробки застосунку буде забезпечуватись зручність роботи та досить висока і якісна синхронізація даних.

						<i>КвРІПЗ. 210104.01.05</i>	Арк.
							38
Змн.	Арк.	№ докум.	Підпис	Дата			

### 2.3 Опис декомпозиції

Декомпозиція – це такий метод розкриття структури застосунку, при якому за певною ознакою її ділять на окремі відокремлені складові.

Декомпозиція у застосунку використовується при побудові так званого «дерева цілей», а також для прив'язки генеральної мети застосунку із способами та методами їх досягнення загалом, що сформульовані у вигляді завдань окремими виконавцями системи.

Розглянемо загальні технологічні аспекти побудови цього «дерева цілей». Не існує таких універсальних методів для побудови «дерева цілей». Його способи побудови залежать від певного характеру цілі системи, який обраний методологічний підхід, а також він залежить від того, хто розробляє це «дерево цілей», та як він являє собі поставлені для нього завдання, та як він «бачить» їхній взаємозв'язок між собою.

У декомпозиції коду на початку сегментуються дані, а потім уже здійснюється алгоритм їхньої обробки та редагування даних. Дані тут розбиваються на окремі сегменти, які приблизно однакові за розміром. З фрагментами різних даних також пов'язуються їх операції обробки, з яких формуються підзадачі системи. У загальному визначаються так потрібні правила обміну даними між ними. Перекриття для різних частин, на які розбивається задача, у цих випадках додатку є мінімальною. Таким чином вдається уникнути у застосунку дублювання різних обчислень. Схема розбиття самої системи на підсистеми може в подальшому дещо уточнюватися. Часто, за необхідності, для зменшення такої кількості обмінів, допускається збільшення незначного ступеня перекриття різних фрагментів задачі застосунку.

При декомпозиції сама вихідна система розташовується на так званому нульовому рівні. Далі, після її розбиття, йдуть підсистеми першого та наступних рівнів. Розчленування таких підсистем системи, або деяких із них, призводить до появи нового рівня підсистем (другого та третього рівнів) тощо.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						39
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Спрощене графічне подання такої декомпозиції системи називається ієрархічною структурою програмного застосунку.

В основу розробки різнотипних модульних застосунків закладено принцип модульної декомпозиції. Загалом, відомо два основних типи моделей модульної декомпозиції \_ це так звана модель потоку даних та модель об'єктів.

В основі моделі потоку даних лежить здійснення розбиття за окремими функціями, які використовуються при розробці процедурно-орієнтованих програмних застосунків. Сама ж модель об'єктів полягає у слабозчеплених сутностях системи, які мають свої, власні набори даних інформації, свої певні стани та набори операцій .

Поставлену задачу було нами розділено на декілька модулів:

- модуль навігації, який включає зокрема в себе інтуїтивно зрозумілі для користувача головне меню, панель інструментів, різноманітні кнопки та контекстне меню для кожного з блоків веб-сайту;
- модуль редагування даних, який містить усі необхідні та потрібні елементи інтерфейсу, за допомогою яких користувачеві дозволяється додавати, змінювати, редагувати а також видаляти дані. Він призначений для зміни інформації у базі даних застосунку;
- модуль фільтрації, який дозволяє користувачам шукати інформацію відповідно до його запитів та призначений для коректного відображення інформації;
- модуль перегляду, який включає в себе усі таблиці та поля з інформацією та послуги;

Модуль роботи адміністратора сайту забезпечує можливість виконання робіт по створенню, перегляду, обробці, редагування та видалення даних про працівників, клієнтів, послуги та графіку роботи веб-студії. Виходячи з поставленої мети декомпозиції, декомпозуємо модуль до таких функцій: "Створення інформації про клієнта", "Перегляд портфоліо", "Створення нового користувача", "Корегування інформації", "Корегування інформації про послуги та роботи". Ці

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

функції отримують введені адміністратором системи дані та передаються в базу даних. Модуль роботи з базою даних у застосунку забезпечує можливість опрацювання та вибірки даних, а також для роботи з редагування, перегляду та видалення даних адміністратором застосунку. В загальному, база даних застосунку призначена для зберігання інформації про працівників, співробітників, клієнтів, послуги, роботи та даних про них. На етапі програмування вибірка у застосунку здійснюється за допомогою мови SQL-запитів.

Для запобігання проблем у зв'язку з внесенням та відображенням некоректних даних, особливо, при роботі з базою даних застосунку, зв'язками між таблицями, ключовими полями, нам необхідно створити певну систему обмежень, та, зокрема, передбачити роботу загальної логіки застосунку.

У застосунку необхідно провести декомпозицію таким чином, щоб його окремі елементи працювали, як можливо незалежно один від одного. З існуючих методів проведення декомпозиції, нами було обрано модель потоків даних, де основі моделі лежить розбиття системи на окремі функції. Результатом нашої декомпозиції є модель застосунку, яка складається з ієрархічно впорядкованих наборів діаграм, які спрощуються із кожним наступним рівнем ієрархії.

Для початку створимо контекстну діаграму, яка опише даний сайт в цілому та зв'язок з предметною областю задачі. Від зовнішніх сутностей бази даних до сайту поступають такі сигнали та дані:

- дані про співробітників;
- дані про роботи співробітників;
- дані про портфоліо співробітників;
- дані про клієнтів сайту;
- інформація про спеціалізацію співробітників;
- дані про послуги.

Сайт повертає наступні інформаційні дані:

- інформація про офіс та його підрозділи;
- дані з інформацією про співробітників;

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						41
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- дані з інформацією про послуги веб-студії;
- дані про портфоліо співробітників;
- дані з інформацією про часи прийому адміністрацією;
- відгуки клієнтів про співробітників;
- відгуки клієнтів про роботи та послуги;
- рейтинги для кожного роботи та послуги.

## 2.4 Опис залежностей

Усі наші залежності зобразимо у вигляді UML-діаграми (Component diagram). Загалом діаграма компонент відображає залежності, які наявні між компонентами сайту з включенням компонентів вихідного коду та різні компоненти, а також про інші складові, які виконуються у застосунку.

Сюди входять такі користувачі: адміністратор, співробітник, клієнт, послуга.

Основний модуль компонента «Клієнт» керує такими модулями: інтерфейсів, звітів та модулем візуалізації (Рисунок 2.6):

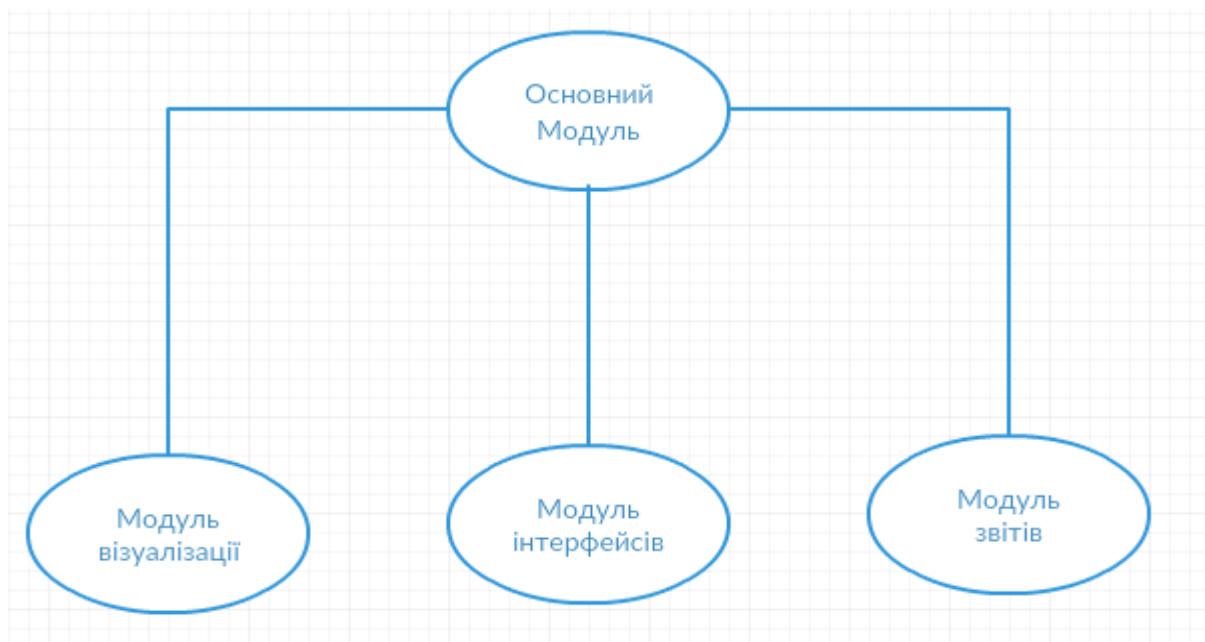


Рисунок 2.6 - Залежності між модулями пакету «Клієнт»

Файл-серверна архітектура застосунку висуває певні вимоги до користувачької частини сайту, хоч зі сторони користувача необхідна наявність лише будь-якого браузера. Так, архітектура застосунку вимагає також коректного відображення усіх стилів застосовного фреймворку та виконання JQuery-скриптів. Найоптимальнішим є використання таких версій браузерів як Chrome, Opera, Edge, Safari тощо.

Для успішного функціонування браузера для нашого сайту, характеристики комп'ютера повинні відповідати таким основним вимогам:

- процесор з тактовою частотою не нижче 1,1 ГГц;
- оперативна пам'ять – 4 Gb та вище;
- об'єм вільного місця на диску не менше 1 Gb;
- операційна система з графічним інтерфейсом;
- маніпулятор мишка, або тачпад, клавіатура;
- наявність підключення до мережі.

Для початку роботи застосунку користувач необхідно встановити будь-який веб-браузер. Далі йому необхідно ввести у поле пошуку веб-адресу URL для посилання на веб-сайт та, звісно, натиснути кнопку «Перейти».

## 2.5 Опис інтерфейсів

Опис інтерфейсів сайту зобразимо у нашій моделі у вигляді систми. До цих інтерфейсів буде також належати вибір певного портфолію, вибір певної послуги, опрацювання даних співробітника, опрацювання даних користувачів, а також додаткові послуги.

Інтерфейсу застосунку у нас буде підпорядкована сутність, а цій сутності – підпорядковані модулі, контролери, генератори, сервіси та також їх граничні сутності. Крім того, у застосунку наявна база даних. Шаблоном проектування нами було обрано патерн «Abstract Fabric».

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

Розроблена діаграма інтерфейсів програмного модуля вибору роботи наведена у додатку Д.

## 2.6 Детальне проектування та розробка модулів застосунку модулів

При реалізації детального проектування нашого застосунку необхідно збудувати його модульну модель. Під модульною моделлю тут розуміється реальна модель проєктованого сайту. Процес побудови модулів включає в себе уточнення моделі бази даних з обов'язковою її нормалізацією для подальшої генерацією SQL-запитів. Також здійснюється уточнення структури користувацького інтерфейсу. Здійснюється побудова структурних схем, які відображають логіку роботи інтерфейсу та модель бізнес-логіки сайту. Загальний алгоритм роботи модуля «Оформлення замовлення» сайту представлений у вигляді «діаграми послідовності» на рисунку 2.8.

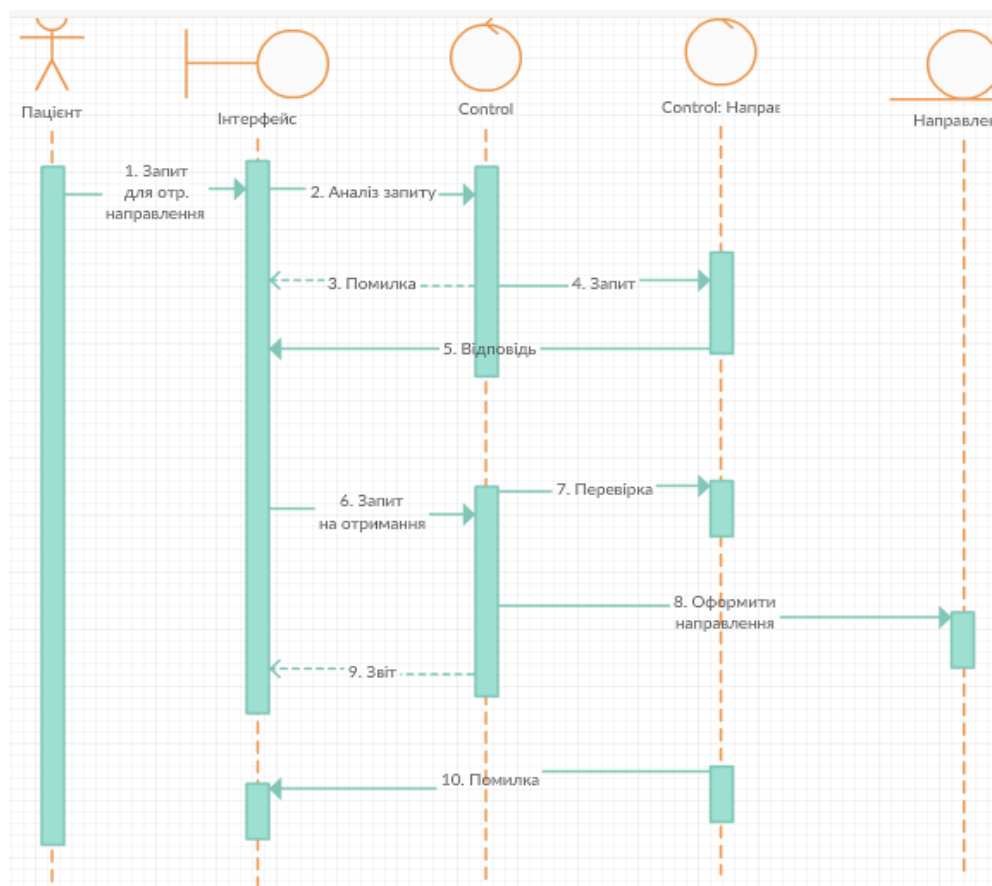


Рисунок 2.8 – Детальне проектування модуля «Оформлення послуги»



клас, що наслідує основний клас (Controller) та містить у собі методи, що відповідають за генерацію відповідних веб-сторінок та веб-представлень. Кожен контролер застосунку взаємодіє з шаблоном об'єктно-реляційної поведінки Unit Of Work, в задачі якого входять відстеження змін об'єктів та класів під час транзакцій, що відбуваються в базі даних, та при роботі окремих функцій та і всього застосунку.

Все це відображено на діаграмі класів, яка наведена у додатку Е, на якій зображено діаграму основних класів модуля Model. Так, на цій діаграмі представлено наступні класи з їхніми полями:

- клієнт;
- реєстрація;
- послуга;
- адміністратор.

## 2.7 Детальне проектування даних

Загалом, початок детального проектування даних застосунку, має починатись з реалізації бази даних сайту.

База даних нами використовується для введення-виведення даних, збереження, редагування, обробки та видалення інформації. Бази даних володіють досить суттєвими перевагами відносно інших способів та методів організованої певним чином інформації:

- для баз даних характерною особливістю є однократне введення та багатократне використання даних, доступні дані застосовуються для вирішення багатьох завдань та забезпечується їй багатоцільове та сумісне використання;
- бази даних в системах існують та працюють незалежно від конкретних застосунків, що забезпечує, в свою чергу, уніфікацію для засобів організації її даних та незалежність програмних застосунків від організації самих даних;

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

- бази даних володіють певною модальністю, структурованістю, яка відображає певну предметну ділянку окремої задачі;
- бази даних також дозволяють встановити мінімально необхідний рівень для надлишковості даних, тобто дані не будуть дублюються при їх використанні різними засобами;
- в базі даних забезпечується конкретне дотримання стандартів представлення даних, що спрощує їх для створення та обслуговування загалом;
- в базах даних забезпечується централізоване керування інформацією, синхронна підтримка усіх даних для всіх застоснків, включаючи мови запитів та засоби їх захисту.

В нашому проєкті буде використовуватиметься саме реляційна модель даних, адже дає змогу забезпечити повну незалежність даних, а також має визначені правила проєктування, що базуються математичним апаратом.

Обрано було саме було СКБД MySQL через її високу продуктивність, високу швидкодію та низьку вартість. Також MySQL може використовуватися в середовищі багатьох різних системах LINUX (UNIX), а також у середовищі Microsoft Windows Server.

Структура розробленої таблиці БД «Співробітники» показана на рисунку 2.9:



Рисунок 2.9 – Структура таблиці «Співробітники» в СКБД MySQL

Структура таблиці БД «Замовлення» показана на рисунку 2.10:



Рисунок 2.10 – Структура таблиці БД «Замовлення» в СКБД MySQL

Створюємо таким чином модель бази даних, та нормалізуємо її. Для приведення до третьої нормальної форми, ми дані, що повторно з'являються в декількох рядках таблиці виносимо в окремі рядки. Відношення, яке містило назву міста та її область розбиваємо на два відношення, та відношення, яке буде містити індифікатор замовлення послуги, назву замовлення послуги та на відношення, яке буде містити назву послуги, індифікатор послуги, індифікатор замовлення послуги. Аналогічно проводимо роботи із іншими відношеннями, які підлягають такій нормалізації.

Для того, щоб привести все до третьої нормальної форми, потрібно винести поля, що залежать від основного ключа відношення.

Тому для реалізації зв'язку «many-to-many» ми створюємо додаткові проміжні відношення «Місця роботи співробіників» та «Місця роботи», таким чином співробітник може працювати в багатьох офісах. Аналогічно робимо із іншими відношеннями, які підходять даній нормалізації.

У результаті проведення нормалізації нами було отримано наступні таблиці: admins, calendars, certificates, tauns, docspec, special, spvnzes, spworkplaces, favorites, feedbacks, messages, newmeseg, option, pages, posts, spost, price, ratings, , subscription, telephon, user\_settings, vsnzes, workplaces.

## 2.7 Висновок другого розділу

В даному розділі було здійснено опис залежностей застосунку та опис самого інтерфейсу. Проведена нами декомпозиція, на основі розробленої архітектури програмного застосунку (сайту). Здійснено детальне проектування сайту та розробка модулів сайту, а також та детальне проектування даних.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						49
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

#### 3.1 Опис структури даних та моделі бази даних

Розглянемо основні модулі сайту, які будуть розроблені для нашого сайту. Першими відповідно будуть модулі для роботи із базою даних сайту. Їх загальна кількість буде десять та кожен із них, відповідно, буде відповідати за роботу із окремими таблицями в базі даних. Перерахуймо їх та опишемо: користувач (user) для роботи із відношенням Користувачі в БД;

- таблиця login - для роботи із відношенням Логіни користувачів в БД;
- таблиця klient - для роботи із відношенням Клієнти в БД;
- таблиця worker - для роботи із відношенням Працівники в БД;
- таблиця work - для роботи із відношенням Роботи в БД;
- таблиця consult - для роботи із відношенням Консультації в БД;
- таблиця comment- для роботи із відношенням Коментарі в БД;
- таблиця poslugu - для роботи із відношенням Послуги БД;
- таблиця order - для роботи із відношенням Замовлення в БД;
- таблиця reestr - для роботи із відношенням Реєстр в БД.

Кожен із наведених модулів буде в застосунку реалізовувати CRUD-операції для доступу до різних відношень в нашій базі даних, відповідно до встановленого нами шаблону «Об'єкт доступу до даних».

Після таблиць ідуть розроблені нами класи, що реалізуються патерном «Unit of Work». Цей патерн об'єднує усі таблиці та надає ряд інструментів та засобів для роботи з самою базою даних. Наступним кроком йдуть у нас контролери, які оброблюють запити користувачів сайту.

Перерахуймо та опишемо їх:

- контролер автентифікації користувача;
- контролер для роботи із користувацькими даними;

										Арк.
										50
Змн.	Арк.	№ докум.	Підпис	Дата						

- контролер для роботи із даними про співробітників;
- контролер для роботи із даними про послуги офісу;
- контролер для управління веб-системою;
- контролер для керування аккаунтами користувачів.

Далі необхідно проаналізувати усі засоби розробки програмного застосунку за допомогою яких можна реалізувати сайт.

Визначимо структуру для кожного модуля сайту.

Сховище для роботи із користувачами нашого сайту відношення містить CRUD-операції для роботи із повязаними відношеннями в самій базі даних. Дане сховище має назву UserRepository. Це сховище перевизначається з базової основи шаблону Identity Framework.

Модуль для роботи із відношенням послуг співробітників в базі даних містить CRUD операції, які призначені для роботи із цим відношенням в спроектованій нашій базі даних. Цей модуль має назву PrRepository.

Модуль для роботи із відношенням співробітники в базі даних містить CRUD-операції для роботи із цією таблицею в спроектованій базі даних. Даний модуль має назву WorkerRepository.

Модуль для роботи із таблицею послуг у в БД містить CRUD-операції для роботи із цією таблицею в спроектованій базі даних. Даний модуль має назву workRepository.

Модуль для роботи із таблицею керування в БД містить CRUD-операції для роботи із цією таблицею в спроектованій базі даних. Даний модуль носить ім'я MangRepository.

Модуль для роботи із таблицею аккаунти в базі даних містить CRUD-операції для роботи із цією таблицею в спроектованій базі даних. Даний модуль носить ім'я PostRepository.

Модуль для роботи із таблицею послуги в БД містить CRUD-операції для роботи із цією таблицею в спроектованій базі даних. Даний модуль носить ім'я ConsRepository.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Усі спроектовані відношення, що наведені вище побудовані на базі шаблону проектування Repository.

І на останок у нас є група модулів, що належать до контролерів застосунку . Архітектура цих модулів базується на основі архітектури Controller. Тому всі вони будуть мати схожу та ідентичну будову.

### 3.2 Керівництво користувача

В результаті роботи нами створено веб-застосунок сайту-портфоліо веб-студії. При заході на цей сайт будь-якому користувачу буде потрібно зареєструватись, від чого, в подальшому, буде залежати відповідний функціонал роботи його на сайті. Неавторизований користувач не зможе здійснювати зміни на сайті.

Якщо користувач зайшов уперше на цей сайт та ще не є зареєстрованим у самій системі, то йому потрібно буде натиснути на кнопку «Створити акаунт». При натисканні на кнопки «Увійти» користувач переходить на окрему сторінку авторизації, де йому буде необхідно ввести логін та пароль. Вигляд головної форми сайту-портфоліо можна побачити на рисунку 3.1.

					КвРІПЗ. 210104.01.05	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		



			
<p><b>Strategy</b> Our goal is to identify the business problem to walk away with the perfect and creative solution.</p>	<p><b>Punctuality</b> Bring the key message to the brand's audience for the best price within the shortest possible time.</p>	<p><b>Diligence</b> Research and confirm brands that present the strongest digital growth opportunities and minimize risk.</p>	<p><b>Technologies</b> Design practice focused on digital experiences. We bring forth a deep passion for problem-solving.</p>

### Our Team





 <p><b>Mark Guerrero</b> Product Designer</p> <p><a href="#">Instagram</a> <a href="#">Twitter</a> <a href="#">Facebook</a> <a href="#">LinkedIn</a></p>	 <p><b>Tom Ford</b> Frontend Developer</p> <p><a href="#">Instagram</a> <a href="#">Twitter</a> <a href="#">Facebook</a> <a href="#">LinkedIn</a></p>	 <p><b>Camila Garcia</b> Marketing</p> <p><a href="#">Instagram</a> <a href="#">Twitter</a> <a href="#">Facebook</a> <a href="#">LinkedIn</a></p>	 <p><b>Daniel Wilson</b> UI Designer</p> <p><a href="#">Instagram</a> <a href="#">Twitter</a> <a href="#">Facebook</a> <a href="#">LinkedIn</a></p>
---	--	---	--

Рисунок 3.1 – Головна сторінка сайту-портфоліо.

Після авторизації на сайті, користувач потрапляє на сторінку портфоліо, веб-студії який зображено на рисунку 3.2.

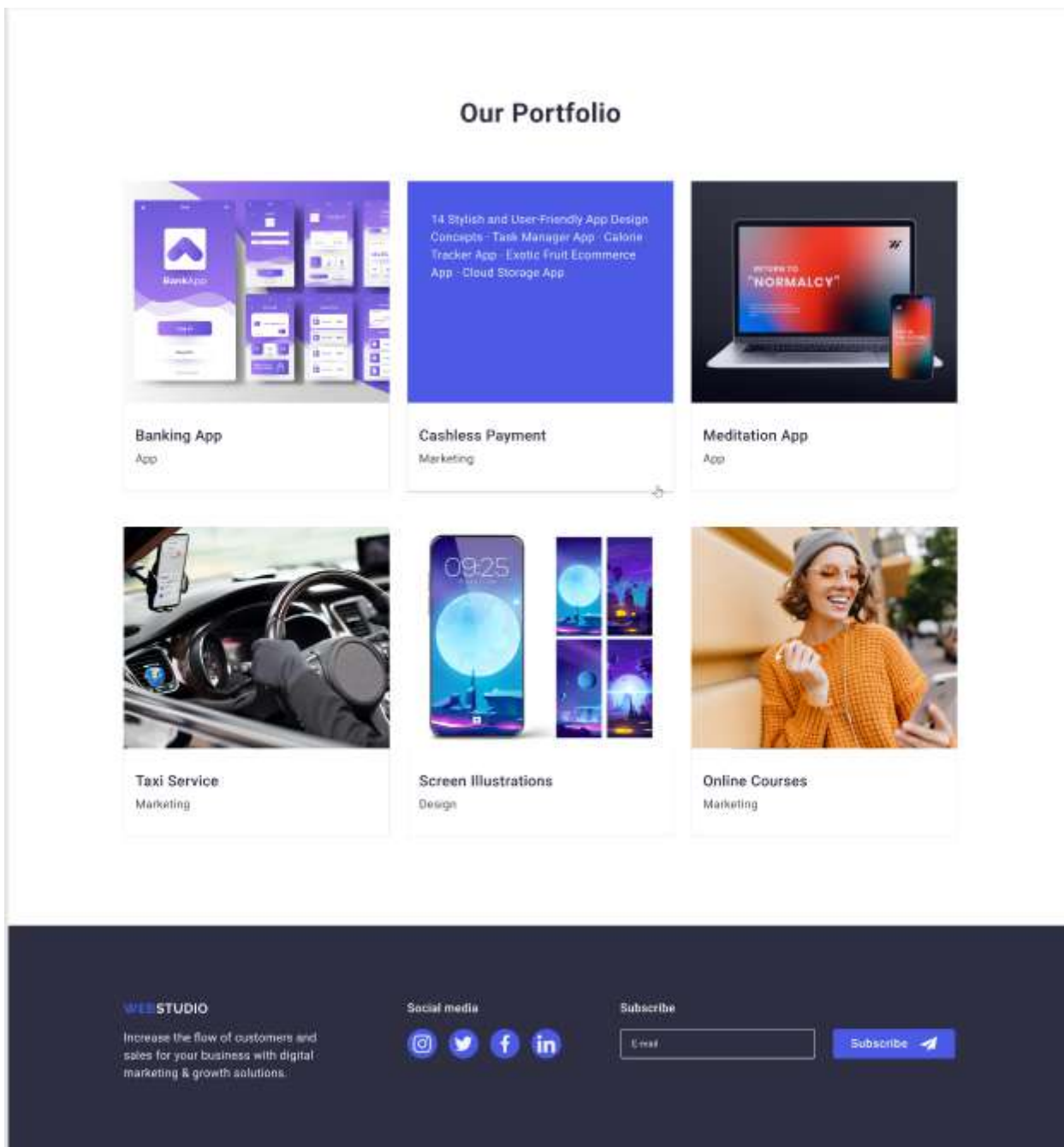


Рисунок 3.2 – Сторінка зворотнього зв’язку «ПОРТФОЛІО»

Перейшовши до сторінки зворотного зв’язку з розробниками, користувач може записатися до нього та охарактеризувати роботу, та замовити її (Рисунок 3.3):

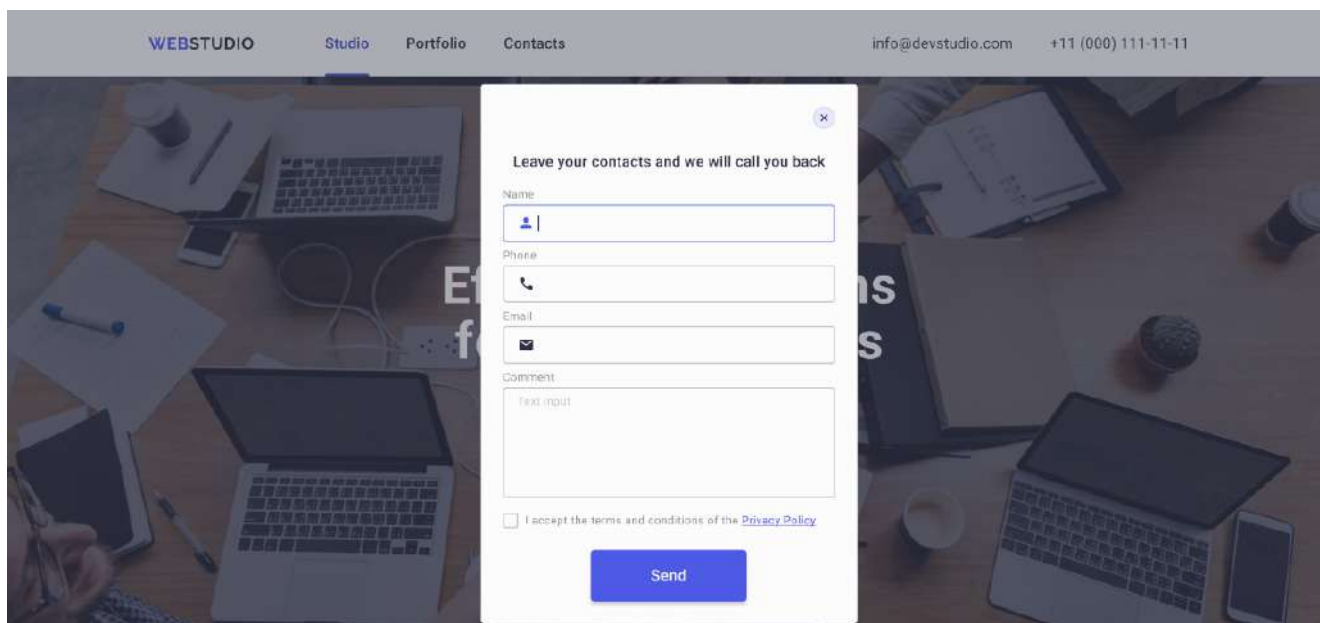


Рисунок 3.4 – Модальне вікно запису

Користувацький досвід експлуатації сучасних сайтів користувачами багато у чому залежить від реалізації його інтерфейсу та реалізації його дизайну, зокрема розміщення елементів керування на ньому та кольорової гамми його оформлення. Тому при виконанні проекту нами були взяті до уваги сучасні вимоги до проектування інтерфейсів.

### 3.3 Налаштування та тестування веб-сайту

Для перевірки працездатності розробленого веб-застосунку здійснимо процес його тестування.

Загалом тестування програмного забезпечення включає такі компоненти, як аналіз, планування, проектування та виконання тестів для перевірки його роботи.

Поняття якості ПЗ Quality Assurance (QA) – це забезпечення якості застосунків. Тестувальник ПЗ (QA-тестувальник) здійснює перевірку, контроль, та відповідає і забезпечує якість роботи усіх розроблювальних програмних застосунків у компанії. Також на ньому висить відповідальність за певні, окремі етапи розробки певного застосунку, зокрема: за вибір інструментів для

											Арк.
											55
Змн.	Арк.	№ докум.	Підпис	Дата							

проектування та розробки ПЗ, він передбачає конкретні проблеми та нештатні ситуації, безпосередньо приймає участь у процесі вдосконалення ПЗ та інше. До зони його відповідальності входять усі етапи розробки ПЗ, починаючи від аналізу та здійснення опису проекту, подальший супровід та тестування на етапі проектування, сам процес тестування уже готового програмного продукту, а також його тестування в процесі експлуатації при внесенні змін до нього.

Поняття якості ПЗ Quality Control (QC) – це контроль якості самого програмного застосунку. Основна задача QC спеціаліста з якості, це перевірка певного конкретного програмного застосунку, який включає в процес тестування аналіз коду застосунку, контроль якості його дизайну та, відповідно, сам процес тестування. QC-спеціаліст повинен розробляти стратегію тестування для кожного типу тестування програмного застосунку. Також він має здійснювати взаємодію з розробниками програмного застосунку та здійснювати тестування розроблювального програмного застосунку.

Загалом, основна робота тестувальника якості QA з тестування полягає у виконанні тестів над програмним застосунком. Загалом, весь процес тестування, це загалом перевірка результату, як працює програмний виріб на відповідність його наперед визначеним та заданим критеріям. В сферу відповідальності QA-тестувальника входить тестування всього програмного застосунку, а також його окремих компонентів. Процес тестування програмного застосунку відіграє досить важливу роль у забезпечення якості та працездатності програмного застосунку.

Процес тестування QA-тестувальником на сьогодні визначений стандартами та підходами до тестування і відображається різними регламентованими документами, які коротко розглянемо нижче.

Test Plan (TP) – план тестування. Він відображений у документі, де описується обсяг виконуваних робіт із тестування програмного застосунку, який має початок з опису самого об'єкта тестування. Далі розписано початок та закінчення процесу тестування, застосовні стратегії тестування, розклад процесу тестування, визначених критеріїв для тестування, необхідні вимоги до обладнання,

						<i>КвРІПЗ. 210104.01.05</i>	Арк.
							56
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			

що необхідне в процесі роботи тестувальника, спеціальних знань тестувальника та також оцінки різного роду ризиків, які можуть виникнути, разом із можливими варіантами для їх вирішення в процесі тестування.

Test Design (TD) – тест дизайн сайту . Це ще один етап процесу тестування сайту, на якому здійснюється сам процес проектування та створення різних тестових випадків. Це так звані тест-кейси, у відповідності до визначених нами критеріїв якості, а також загальною метою тестування.

Test Case (TC) – це тестовий випадок, або так званий «Тест кейс» – це документ, у якому описуються усі кроки, задані умови та їх параметри, які необхідні нам для перевірки та реалізації функцій сайту який тестується, або деякої його частини.

Bug Report (BR) – звіт про помилки і дефекти – документ у якому загалом описується ситуація та послідовність певних дій, які привела до некоректної роботи сайту чи деякої функції, із вказуванням причин їх появи та результату який мав очікуватись (Expected Result).

Test Coverage(ТC) – так зване «Тестове Покриття» – це одна з метрик тестування, яка здійснює оцінку якості самого процесу тестування сайту. Вона передбачає із себе щільність покриття усіма тестами вимог до виконуваного коду сайту.

Test Case Specification (TCS) – це «Деталізація Тест Кейсів» —визначений рівень деталізації тесту з описом певних тестових кроків тестування та необхідного результату його виконання, при якому забезпечується певне визначене співвідношення часу для проходження тесту до самого тестового покриття.

Test Case Pass Time (TCPT) – це «Час Проходження Тест Кейсу» – визначений час від самого початку проходження визначених нами кроків тест-кейсу до самого моменту отримання усіх результатів тестування сайту.

Враховуючи визначені вимоги і стандарти створимо наш план тестування для розробленого сайту-портфоліо вебстудії.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						57
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Метою складання даного ТЗ є опис самого процесу тестування нашого веб-сайту портфоліо. Даний документ дозволяє отримати дані про уявлення запланованих робіт з тестування сайту.

Вихідними даними для нашого тестування є програмний сайт портфоліо веб-студії, який дозволяє користувачеві переглянути роботи та послуги співробітників фірми.

Метою нашого тестування сайту портфоліо є перевірка роботи на правильність та коректність всіх його функціональних дій та можливостей у різних версіях веб-браузерів із використанням різних сценаріїв при його використанні. Частина виділеного нами часу (приблизно десять відсотків) буде використана в ньому для здійснення тестування різних нетипових, потенційно можливих спричинених помилок при роботі різних функцій сайту та сценаріїв його використання.

Підсумком процесу проведеного нами тестування будуть наступні документи:

- головний загальний висновок QA-тестувальника, щодо його загального стану застосунку, який він дає користувачам, розробникам, програмістам, а також менеджерам цього застосунку, та загальне бачення коректності роботи сайту у різного роду браузерів;
- загальний звіт про основні результати тестування окремих його функціональних частин та типові сценарії для використання користувачами веб-браузерів;
- задокументовані неточності, помилки та помилки у баг-трекері для розробників сайту.

Тестування нашого сайту будемо проводити ручним методом з точки зору кінцевого користувача веб-сайту портфоліо.

Далі визначимо головні умови для тестування для нашого сайту.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						58
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Наш сайт повинен задовольняти потреби усіх користувачів в різного роду його активностях, які пов'язані пошуком портфоліо та замовленнями різних послуг і робіт.

Далі визначимо стратегію самого процесу тестування нашого програмного застосунку (сайту).

Стратегія нашого плану тестування є формальною, оскільки для побудови розширеного плану нам необхідно розуміти саму суть поточного стану проекту. В результаті нашої першої ітерації запланованих функціональних та нефункціональних тестів до нашого тест-плану внесемо певні зміни для його покращення. Перша така ітерація функціональних тестів сайту дасть нам чітке уявлення про рівень відмовостійкості та стабільності самої системи та певним чином визначений набір усіх тест-кейсів, які будуть виконані нами для в кожній конфігурації.

Такий підхід дасть нам можливість отримати досить детальний звіт щодо сайту, який ми тестуємо та зосередити нам максимально увагу на усіх виявлених вузьких місцях на сайті.

Замовнику надаватимуться постійно різноманітні звіти про хід нашого тестування сайту. Зокрема, це інформація про знайдені дефекти та пропозиції щодо покращення роботи сайту та оформлення його інтерфейсу і дизайну загалом. Усі виявлені нами помилки та дефекти заносяться також до так званого баг-трекера замовника у вигляді різних файлів, які використовуються для потрібних кроків щодо їх виправлення.

Загалом для нашого сайту було заплановано п'ять етапів проведення усього процесу тестування розробленого веб-сайту студії:

– на першому етапі тестування здійснюється аналіз ТЗ, складається основний тест-план, а також здійснюється часткове тестування функціональних усіх тестів;

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						59
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

– на другому етапі тестування застосовується до детального тестування функціональних тестів, де здійснюється виявлення та опис багів, помилок та дефектів;

– третій етап тестування передбачає проведення тесту роботи у різних браузерах з описом усіх виявлених помилок та багів;

– а четвертому етапі тестування здійснюється перевірка усунених розробниками багів та помилок та проведення регресійного тестування сайту;

– на п'ятому етапі тестування здійснюється тестування дизайну та інтерфейсу з описом усіх знайдених недоліків та помилок.

При такому типі плану тестування, досягається максимальна деталізація для усієї глибини проведеного тестування програмного застосунку. Вона дозволяє нам більш детально та точніше визначити різні параметри ресурсів на затрати та дозволяє самим розробникам сайту виправляти різні помилки та неточності на якомога ранніх етапах проектування.

Визначимо операційні системи для проведення нашого тестування:

- MS Windows 10;
- MS Windows 11;
- Android версій 8-12;
- iOS 17.

Визначимо операційні системи для проведення нашого тестування:

- Google Chrome;
- MS Edge;
- Firefox;
- Opera;
- Safari.

Визначимо та опишемо типи тестування проведеного з нашим веб-сайтом

### 3.3.1 Функціональне тестування сайту

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Мета тестування:

Виявлення різного роду функціональних та не функціональних помилок, невідповідностей технічному завданню тощо. Виявлення очікувань від користувача шляхом здійснення як стандартних так і нестандартних тестових сценаріїв для цього.

Опис процесу функціонального тестування сайту:

- авторизація та реєстрація користувача на сайті;
- авторизація користувача на сайті;
- реєстрація користувача на сайті;
- вхід для анонімного користувача сайту;
- вхід за логіном адміністратора системи сайту;
- відновлення логіну пароллю сайту;
- редагування облікового запису користувача сайту.

Тестування персонального кабінету користувача:

- створення інформації в анкеті користувача
- редагування інформації з анкети користувача;
- видалення інформації з анкети користувача;
- статус заявки користувача;
- статус затвердження заявки користувача;
- статус підтвердження та скасування замовлення;
- вхід/вихід з персонального кабінету користувача;
- тестування зворотнього зв'язку з сайтом;
- статус прикріплення вкладень до повідомлень;
- статус повідомлення сайту.

Тестування функції пошуку та фільтрації на сайті:

- здійснення пошуку та фільтрації за розділами сайту;
- здійснення пошуку різноманітних публікацій;

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

– здійснення поширення та пошуку інформації у соцмережах.

Тестування відображення рисунків та фотографій:

- перелистування сторінок;
- відображення списку рисунків та зображень.

Тестування функції банерів сайту:

- перегортання рисунків та зображень.

Тестування функцій коментарів сайту:

- додавання коментаря на сайт;
- редагування коментаря сайту;
- видалення коментаря на сайті.

Тестування функцій фільтрування сайту:

- фільтрування за певними категоріями послуг;
- фільтрування по різних послугах;
- фільтрування співробітників.

Тестування функцій сортування та фільтрації:

- сортування та фільтрація за категоріями послуг;
- сортування та фільтрація за співробітниками фірми;
- сортування та фільтрація за послугами;
- сортування та фільтрація за цінами.

Тестування функцій оплати послуг фірми:

- оформлення послуг.

Тестування функцій сторінки користувача:

- перегляд інформації про роботу;
- перегляд фотографій на сайті;
- додавання коментарів на сайті;
- перегляд та вибір послуг.

Тестування функцій кошика сайту:

- додавання послуг до кошика сайту;
- видалення послуг з кошика сайту .

					КвРІПЗ. 210104.01.05	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Тестування функцій кросбраузерності здійснюється відповідно до головної мети сайту, а також з метою перевірки коректності його роботи, загального вигляду інтерфейсу а також дизайну сайту у різноманітних браузерах.

Тестування у різних браузерах:

- Google Chrome;
- Edge;
- Firefox;
- Opera;
- Safari.

Проведемо регресійне тестування та перевірку виправлених помилок та дефектів програмного застосунку.

Головна мета тестування будь-якого програмного застосунку, це перевірка усіх внесених змін, що зроблені розробниками для впевненості, що виправлена версія програмного коду застосунку не містить різних помилок та відхилень на усіх протестованих функціях програмного застосунку.

Загалом, у ході проведеного нами регресійного тестування програмного застосунку було здійснено наступні види тестування:

- тестування версії програмного застосунку;
- тестування основного функціоналу програмного застосунку;
- тестування допоміжного функціоналу програмного застосунку;
- верифікаційне тестування програмного застосунку.

Тестування функціоналу веб-сайту та дизайну його інтерфейсу.

Метою проведення такого тестування є перевірка відповідності роботи інтерфейсу розробленій специфікації у технічному завданні.

Опис процесу тестування веб-сайту:

- перевірка входу на сайт;
- перевірка реєстрації на головній сторінці сайту;
- перевірка клікабельності пошти та телефону;
- перевірка здійснення пошуку на сайті;

					КвРІПЗ. 210104.01.05	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

- перевірка правильності роботи гіперпосилань;
- перевірка здійснення масштабування;
- перевірка завантаження картинок;
- перевірка роботи модальних форм;
- перевірка переходу по веб-сторінках;
- перевірка роботи на мобільному пристрої Android;
- перевірка замовлення послуг;
- перевірка роботи переходу на соціальні мережі.

Результати формування нашого плану робіт наведено у таблиці 3.1.

Таблиця 3.1 – План робіт щодо тестування сайту-портфоліо веб-студії

Завдання	Час задіяної роботи	Дата початку	Дата закінчення
Складання плану тестування	8 год	10.05.2024	10.05.2024
Проведення тестів	8 год	12.04.2024	12.05.2023
Аналіз отриманих результатів тестування	6 год	12.04.2024	12.05.2024
Формування звітів	8 год	12.04.2024	12.05.2024

Кінцевим результатом проведення нашого тестування є оформлений звіт з кінцевими результатами процесу тестування програмного виробу з описами виявлених помилок, дефектів, багів з відповідними рекомендаціями та методами щодо вдосконалення програмного застосунку з точки зору його кінцевого користувача.

Тестування функціональних можливостей та функційної поведінки застосунку необхідне для перевірки відповідності запланованому функціоналу та специфікаціям ТЗ.

Тестування, при якому ігноруються внутрішні механізми програмного застосунку, або його окремого компонента, який концентрується на вхідних та вихідних даних, з метою отримання результату на введення та виведення даних користувачем та умови виконання різних сценаріїв.

Загалом, ціллю функціонального тестування програмного забезпечення є перевірка та відповідність його потрібній функціональній специфікації, що зазначена в ТЗ з проектування.

В результаті такого виконання функціонального тестування програмного застосунку створюються та звані чеклисти – одного із фундаментальних інструментів для тестування програмних застосунків. Протоколи чеклистів дозволяють постійно мати перелік пунктів для тестування. Вони дозволяють фіксувати результати тестування, а також здійснювати контроль самого процесу тестування та вести додаткову статистику про стан програмного застосунку на конкретному етапі його розробки чи експлуатації.

Загальні правила складання протоколів чеклистів:

- кожен з пунктів тестування – це визначена окрема операція;
- всі пункти протоколу чеклиста – це атомарні та повні операції;
- в списку тестів, які подібні між собою, потрібно операції оформлювати окремими пунктами тестування;
- кожен з пунктів тестування має починатися з іменника, або дієслова в невизначеній формі;
- протокол чеклиста та його окремі пункти тестування допускає, що можна застосовувати різні рівня деталізації. Ця умова залежить від вимог замовника до звітності, рівня складності застосунку тощо.

Головною метою протоколу чеклиста виступає необхідність врахування як більшого списку дій для найбільш точної перевірки тестами програмного забезпечення. Підбираючи пункти для тесту слід дотримуватися уніфікованої форми.

Протокол чеклиста потрібен при тестуванні для наступного:

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

- встановлення переліку типів для тестових завдань певного застосунку;
- поділу завдання за різними рівнями кваліфікації;
- збереження звітності про результати тестування.

Протокол чеклиста містить інформацію про:

1. Список тестів.

- детальні назви пунктів кожного тесту;
- інформація про відповідний стан перевірки;
- примітки (за умови необхідності).

2. Оточення перевірки:

- версія збірки, з на якій проводилося тестування;
- тестове оточення у випадку наявності;
- відображає інформацію, в випадку декількох тестувальників, про них.

3. Результат перевірки;

– відображається у вигляді таблиці та може бути також доповнений деталізованим описом перевірки з відповідними рекомендаціями.

Перевіримо основний функціонал роботи веб-сайту та відобразимо отримані дані у таблицю.

Таблиця 3.2 – Результати чек-листа перевірки сайту-портфоліо веб-студії

Номер тесту	Назва тесту	Результат тестування	Примітки
1.	Перевірка відображення текстових написів на сайті	Успіх	
2.	Перевірка коректності відображення кольорової гами інтерфейсу	Пройдено	
3.	Перевірка розташування елементів інтерфейсу (меню та кнопки)	Пройдено	
4.	Перевірка коректності роботи кнопок	Успіх	
5.	Перевірка коректності роботи банерів	Успіх	

6.	Перевірка коректності роботи гіперпосилань	Успіх	
7.	Перевірка переходу з головного меню на сторінки	Успіх	
8.	Перевірка роботи фокуса на елементах	Пройдено	
9.	Перевірка часу завантаження веб-сторінок	Пройдено	
10.	Перевірка дій на веб-сторінках сайту	Успіх	
11.	Перевірка дій на модальних веб-формах	Успіх	
12.	Перевірка прокрутки веб-сторінок сайту	Успіх	

Таблиця 3.3 – Результати чек-листа перевірки функціоналу сайту-портфоліо у браузерях

№	Назва тесту	Google Chrome	Edge	Firefox	Opera	Safari
1.	Перевірка входу на сайт	Успіх	Успіх	Успіх	Успіх	Успіх
2.	Перевірка реєстрації на головній сторінці сайту	Успіх	Невдача	Успіх	Успіх	Успіх
3.	Перевірка клікабельності пошти та телефону	Успіх	Успіх	Успіх	Успіх	Успіх
4.	Перевірка здійснення пошуку на сайті	Успіх	Успіх	Успіх	Успіх	Успіх

5.	Перевірка правильності роботи гіперпосилань	Успіх	Успіх	Успіх	Успіх	Успіх
6.	Перевірка здійснення масштабування	Успіх	Успіх	Успіх	Успіх	Успіх
7.	Перевірка завантаження картинок	Успіх	Успіх	Успіх	Успіх	Невдача
8.	Перевірка роботи модальних форм	Успіх	Успіх	Успіх	Успіх	Успіх
9.	Перевірка переходу по веб-сторінках	Успіх	Успіх	Успіх	Успіх	Успіх
10.	Перевірка роботи на мобільному пристрої Android	Успіх	Успіх	Успіх	Успіх	Успіх
11.	Перевірка замовлення послуг	Успіх	Успіх	Успіх	Успіх	Успіх
12.	Перевірка роботи переходу на соціальні мережі	Успіх	Успіх	Невдача	Невдача	Успіх

### 3.3.2 Валідація

У програмному коді важлива кожна кома, лапка, імена атрибутів та тегів, наявність усіх відкриваючих та закриваючих тегів тощо. За всім цим досить важко

					КвРІПЗ. 210104.01.05	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

встежити і не тільки новачкам. Тому, для цього використовується спеціальний інструмент перевірки — валідатор, який виконує валідацію коду.

Валідація коду потрібна для виявлення помилок у синтаксисі програмного коду, а саме розмітки HTML-документа та виявлення розбіжностей з HTML-специфікацією, згідно вказаної в `<!DOCTYPE >`. Сервіс для такої перевірки називається валідатором. Результатом валідації документ або проходить повністю валідацію, або отримує список помилок та рекомендованих виправлень.

Як зробити валідацію програмного коду? Для цього є спеціальний онлайн сервіс-валідатор, функції якого полягають у перевірці коду веб-сторінок на відповідність стандартам та правилам. Для перевірки кодк необхідно всього два кроки:

- передати у валідатор вихідний код (фалом, текстом);
- натиснути кнопку для перевірки.

Надіслати код для валідації можна таким чином:

- завантажити файлb htm,css;
- передати посилання на live сторінку;
- скопіювати текст і завантажити код в валідатор.

Результати перевірки коду нашлго сайту наведено у Додатку Е.

### 3.4. Висновки до третього розділу

У третьому розділі нами було здійснено опис структури даних та модель сайту-портфолію. Здійснено його практичну реалізацію, наведено інструкцію по користуванню і проведено тестування самого сайту різними засобами.

Таким чином, за допомогою створеного сайту-портфолію з'являється можливість в реальному часі оперативно інформувати усіх клієнтів веб-офісу, як існуючих так і потенційних та партнерів про оновленні продукції і послуг, нові

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

акційні пропозиції, при цьому не використовувати трудомістки та затратні ресурси, такі як масова розсилка електронних листів, повідомлень та телефонування.

					КвРІПЗ. 210104.01.05	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

## Висновки

В нашій роботі реалізовано сайт портфоліо вебстудії. Більшість функцій визначених на етапі проектування вдалося реалізувати у повній мірі.

Автоматизація реалізується за рахунок сучасних засобів введення-виведення інформації та реалізації пошуку та фільтрів.

Основні переваги проекту:

- інтуїтивний інтерфейс користувача;
- зручне додавання, видалення та редагування інформації;
- відсутність прив'язки до певної конкретної апаратної платформи;
- досить низькі системні вимоги для сайту.

У роботі досліджено понятійний апарат нашого дослідження і визначено особливості сайту-портфоліо. Встановлено, що сайт-портфоліо – це інтернет-ресурс, який включає в себе об'єднані посиланнями та загальною структурою документи та дані.

Проведено аналіз сайтів, та також з'ясовано, що існує декілька видів веб-ресурсів: лендинг, сайт-візитка, сайт-портфоліо, інтернет-магазин, сайти послуг, персональний блог тощо.

Опрацьовано та визначено етапи проектування сайту-портфоліо вебстудії: ознайомлення з технічним завданням, підготовка текстового та відображувального матеріалу, підключення до бази даних

Розроблено концепцію та дизайн сайту портфоліо вебстудії.

При створенні видання виконано підбір шрифтового та дизайнерського оформлення, обрано певне колірне рішення для сайту.

Визначена структура сайту, що складається з обкладинки, різнотипних інформаційних блоків, форм замовлення та відображення інформації.

					КвРІПЗ. 210104.01.05	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kevin Yank. Build Yours Owns Database Driven Web Site Using PHP and MySQL. Four Edition. 2021. p.512
2. Davey Shafik - The PHP Anthology. Essential Tips, Tricks and Hacks, 2nd Edition. 2019. 247 p.
3. Юрченко Б.Г. Access 20. Бази даних та додатки. К: "ДіаСофт", 2018. 512с.
4. Ящевський В.Д., Макарчук О.М., Палійський М.О. Практичний курс інформатики" за редакцією академіка АПН України Паламарчука В.М., Київ, 2019. 418 с.
5. ISO/IEC/IEEE 42010, Systems & software engineering. Architecture description.
6. ISO/IEC 2382:2015. Information technology. Vocabulary, Part 1: Terms & definitions.
7. Patrick Rauland. WooCommerce Explained: Your Step-by-Step Guide to WooCommerce. WooExperts Press, 2019. 346 p.
8. Lisa Sabin-Wilson. WooCommerce For Dummies. For Dummies, 2019. 416 p.
9. Brad Williams, David Damstra, Hal Stern. Professional WordPress: Design and Development. Wiley, 2021. 552 p.
10. Brad Williams, Jeff Starr, Lisa Sabin-Wilson. WordPress for Beginners 2023: A Visual Step-by-Step Guide to Mastering WordPress. O'Reilly Media, 2023. 576 стор.
11. 16. Robbert Ravensbergen. Building E-commerce Solutions with WooCommerce: Customize and extend WooCommerce for your online store. Packt Publishing, 2020. 470 p.
12. Lisa Sabin-Wilson. WordPress All-in-One For Dummies. For Dummies, 2022. 912 p.
13. Patrick Rauland, Raúl P. García. "WooCommerce Cookbook". O'Reilly Media, 2020. 384 p.

					КвРІПЗ. 210104.01.05	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Yannick Lefebvre. WordPress Plugin Development Cookbook. Packt Publishing, 2019. 358 p.
15. David Trounce. WooCommerce Made Easy: A Step-by-Step Guide to Setting Up an Online Store". CreateSpace Independent Publishing Platform, 2019. 224 p.
16. Sufyan bin Uzayr. Mastering WooCommerce: Build a Professional Online Store with WordPress and WooCommerce. Packt Publishing, 2020. 320 p.
17. Karol Krol. WordPress 5 Complete: Build beautiful and feature-rich websites from scratch. Packt Publishing, 2018. 616 p.
18. Brad Williams, David Damstra, Hal Stern. Professional WordPress: Design and Development. Wiley, 2021. 552 p.
19. Eric Schwartzman. WooCommerce Tutorial: Build an E-commerce Website. Independently published, 2022. 152 p.
20. Vladimir Prelovac. WordPress Plugin Development: Beginner's Guide. Packt Publishing, 2019. 208 p.
21. Lisa Sabin-Wilson. WooCommerce For Dummies. For Dummies, 2019. 416 p.
22. Simon Brown. Software Architecture for Developers. Leanpub, 2019, 408 p.
23. Eric Evans. Domain-Driven Design Reference: Definitions and Pattern Summaries. Leanpub, 2019, 112 p.
24. Microsoft Azure. Software Architecture Guide. Microsoft Press, 2019, 229 p.
25. Neal Ford, Rebecca Parsons, Patrick Kua. Building Evolutionary Architectures: Support Constant Change. O'Reilly Media, 2017, 206 p.
26. Cornelia Davis. Cloud Native Patterns: Designing Change-Tolerant Software. O'Reilly Media, 2020, 466 p.
27. Anthony Molinaro - SQL Cookbook: Query Solutions and Techniques for Database Developers.: O'Reilly Media, 2019, 576 p.
28. Anthony DeBarros. Practical SQL: A Beginner's Guide to Storytelling with Data. No Starch Press, 2018, 504 p.
29. Seyed M. M. Learning MySQL: Get a Handle on Your Data. O'Reilly Media, 2020, 472 p.

					<i>КвРІПЗ. 210104.01.05</i>	Арк.
						73
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

30. Nathan Marz, James Warren. Data Management at Scale: Lessons Learned from Building Large Distributed Systems. O'Reilly Media, 2020, 200 p.

31. Baron Schwartz, Peter Zaitsev, Vadim Tkachenko. High Performance MySQL: Optimization, Backups, and Replication. O'Reilly Media, 2012, 828 p.

					<i>КвРІПЗ. 210104.01.05</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		74

ДОДАТОК А  
(обов'язковий)

**КОД (ЛІСТИНГ) ПРОГРАМИ**

## Index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/modern-
normalize@2.0.0/modern-normalize.min.css">
    <link rel="preconnect"
href="https://fonts.googleapis.com" />
    <link rel="preconnect"
href="https://fonts.gstatic.com" crossorigin />
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght
@400;500;700;900&display=swap" rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css2?family=Raleway:wgh
t@800&display=swap" rel="stylesheet">
    <link href="./css/styles.css" rel="stylesheet" />
    <title>WebStudio</title>
  </head>
  <body class="allpage">
    <!-- Header section-->
    <header class="hero-header">
      <nav class="hero-header-nav">
        <a class="logo" href="./index.html">Web<span
class="span-logo">Studio</span></a>

```

```

        <ul class="menu">
            <li class="menu-item"><a class="menu-link"
href="./index.html">Studio</a></li>
            <li class="menu-item"><a class="menu-link"
href="">Portfolio</a></li>
            <li class="menu-item"><a class="menu-link"
href="">Contacts</a></li>
        </ul>
</nav>
<!-- адреса -->
<address class="adress">
    <ul class="adress-list">
        <li class="adress-item"><a class="adress-
link"
href="mailto:info@devstudio.com">info@devstudio.com</l
i>
        <li class="adress-item"><a class="adress-
link" href="tel:+110001111111">+11 (000) 111-11-11</li>
    </ul>
</address>
</header>
<main>
    <!-- Hero image -->
    <section class="hero-image">
        <h1 class="hero-image-header">Effective
Solutions for Your Business</h1>
        <button class="btn" type="button">Order
Service</button>
    </section>
    <!-- Benefits section-->

```

```

<section class="benefits">
  <h2 hidden class="benefits-mainheader">Benefits
Your solutions business</h2>
  <ul class="benefits-list">
    <li class="benefits-item">
      <h3 class="benefits-header">Strategy</h3>
      <p class="benefits-text">
        Our goal is to identify the business
problem to walk away with the
        perfect and creative solution.
      </p>
    </li>
    <li class="benefits-item">
      <h3 class="benefits-
header">Punctuality</h3>
      <p class="benefits-text">
        Bring the key message to the brand's
audience for the best price
        within the shortest possible time.
      </p>
    </li>
    <li class="benefits-item">
      <h3 class="benefits-header">Diligence</h3>
      <p class="benefits-text">
        Research and confirm brands that present
the strongest digital
        growth opportunities and minimize risk.
      </p>
    </li>
    <li class="benefits-item">

```

```

        <h3                                class="benefits-
header">Technologies</h3>
        <p class="benefits-text">
            Design practice focused on digital
experiences. We bring forth a
            deep passion for problem-solving.
        </p>
    </li>
</ul>
</section>
<!-- Our Team -->
<section class="ourteam">
    <h2 class="ourteam-header">Our Team</h2>
    <ul class="ourteam-list">
        <li class="ourteam-item">
            
            <h3 class="person">Mark Guerrero</h3>
            <p                                class="personposition">Product
Designer</p>
        </li>
        <li class="ourteam-item">
            
    <h3 class="person">Tom Ford</h3>
    <p
        class="personposition">Frontend
Developer</p>
</li>
<li class="ourteam-item">
    
    <h3 class="person">Camila Garcia</h3>
    <p class="personposition">Marketing</p>
</li>
<li class="ourteam-item">
    
    <h3 class="person">Daniel Wilson</h3>
    <p class="personposition">UI Designer</p>
</li>
</ul>
</section>
<!-- Portfolio -->
<section>

```

```

<h2 class="portfolio">Our Portfolio</h2>
<ul class="portfolio-list">
  <li class="portfolio-item">
    
    <h3 class="portfolioproject">Banking App</h3>
    <p class="portfolio-type">App</p>
  </li>
  <li class="portfolio-item">
    
    <h3 class="portfolioproject">Cashless
Payment</h3>
    <p class="portfolio-type">Marketing</p>
  </li>
  <li class="portfolio-item">
    

```

```

    <h3 class="portfolioproject">Meditation App</h3>
    <p class="portfolio-type">App</p>
</li>
<li class="portfolio-item">
    
    <h3 class="portfolioproject">Taxi Service</h3>
    <p class="portfolio-type">Marketing</p>
</li>
<li class="portfolio-item">
    
    <h3 class="portfolioproject">Screen
Illustrations</h3>
    <p class="portfolio-type">Design</p>
</li>
<li class="portfolio-item">
    
        <h3 class="portfolioproject">Online Courses</h3>
        <p class="portfolio-type">Marketing</p>
    </li>
</ul>
</section>
</main>
<footer class="hero-footer">
    <a
        href="./index.html">Web
        class="footer-logo"
        <span
            class="span-footer-
            logo">Studio</span></a>
    <p class="footer-paragraf">
        Increase the flow of customers and sales for
        your business with digital
        marketing & growth solutions.
    </p>
</footer>
</body>
</html>

```

## Styles. Css

```

/* Base styles */
/* font styles */
body {
    font-family: 'Roboto', sans-serif;
}
ul, ol {list-style-type: none;}
a {text-decoration: none;}
button{cursor: pointer;}
/* body */
.allpage{
    color: #434455;
    background-color: #FFFFFF;
}

```

```

}
/* Header 1 */
.hero-header{
  background: var(--WHITE, #FFF);
}
.hero-header-nav {
  color: var(--NAVY-BLUE, #2E2F42);
}
.logo
{
  color: var(--IRIS, #4D5AE5);
  font-family: Raleway,sans-serif;
  font-size: 18px;
  font-weight: 700;
  line-height: 1.17; /* 116.667% */
  letter-spacing: 0.03em;
  text-transform: uppercase;
}
.span-logo
{
  color: var(--NAVY-BLUE, #2E2F42);
}
.menu-link{
  color:#2E2F42;
  font-size: 16px;
  font-weight: 500;
  line-height: 1.50; /* 150% */
  letter-spacing: 0.02em;
}
.adress{
  font-style: normal;
}
.adress-link{
  color: var(--SLATE, #434455);
  font-family: Roboto;
  font-size: 16px;
  font-style: normal;
  font-weight: 400;
  line-height: 1.50; /* 150% */
  letter-spacing: 0.02em;
}
.menu-link:hover{
  color:var(--OCEAN,#404BBF);
}
.menu-link:active{

```

```

        color:var(--OCEAN,#404BBF);
    }
    .menu-link:focus{
        color: #404bbf;
    }
    .adress-link:hover{
        color:var(--OCEAN,#404BBF);
    }
    .adress-link:focus{
        color: #404bbf;
    }
    .hero-image {
        background: var(--NAVY-BLUE, #2E2F42);
    }
    .hero-image-header{
        /*background: var(--NAVY-BLUE, #2E2F42);*/
        /* background-color: background: var(--navyblue,
#2E2F42);*/
        color: var(--white, #FFF);
        text-align: center;
        /* text-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25); */
        font-family: Roboto;
        font-size: 56px;
        font-weight: 700;
        line-height: 1.07; /* 107.143% */
        letter-spacing: 0.02em;
    }
    .btn{
        background-color:#4D5AE5;
        color:#FFFFFF;
        font-size: 16px;
        font-style: normal;
        font-weight: 500;
        line-height: 1.50; /* 150% */
        letter-spacing: 0.04em;
    }
    .btn:hover, .btn:focus{
        background-color:#404BBF;
    }
    .benefits-mainheader{
        letter-spacing: 0.02em;
    }
    .benefits-header{
        color: var(--NAVY-BLUE, #2E2F42);
        font-size: 20px;

```

```
    font-weight: 500;
    line-height: 1.20; /* 120% */
    letter-spacing: 0.02em;
}
.benefits-text{
    color: var(--SLATE, #434455);
    font-size: 16px;
    font-weight: 400;
    line-height: 1.50; /* 150% */
    letter-spacing: 0.02em;
}
.ourteam{
    background-color: #F4F4FD;
}
.ourteam-list{
    background-color: #FFFFFF;
}
.ourteam-item{
    background-color: #FFFFFF;
}
.ourteam-header{
    color: var(--NAVY-BLUE, #2E2F42);
    text-align: center;
    font-size: 36px;
    font-weight: 700;
    line-height: 1.11; /* 111.111% */
    letter-spacing: 0.02em;
    text-transform: capitalize;
}
.person{
    color: var(--NAVY-BLUE, #2E2F42);
    text-align: center;
    font-size: 20px;
    font-weight: 500;
    line-height: 1.20; /* 120% */
    letter-spacing: 0.02em;
}
.personposition{
    color: var(--SLATE, #434455);
    text-align: center;
    font-size: 16px;
    font-style: normal;
    font-weight: 400;
    line-height: 1.50; /* 150% */
    letter-spacing: 0.02em;
```

```

}
.portfolio{
  color: var(--NAVY-BLUE, #2E2F42);
  font-size: 36px;
  font-style: normal;
  font-weight: 700;
  line-height: 1.11; /* 111.111% */
  letter-spacing: 0.02em;
  text-transform: capitalize;
  text-align: center;
}
.portfolioproject{
  color: var(--NAVY-BLUE, #2E2F42);
  font-size: 20px;
  font-weight: 500;
  line-height: 1.20; /* 120% */
  letter-spacing: 0.02em;
}
.portfolio-type{
  color: var(--SLATE, #434455);
font-size: 16px;
font-weight: 400;
line-height: 1.50; /* 150% */
letter-spacing: 0.02em;
}
.hero-footer{
  background: #2e2f42;
  font-family: Raleway, sans-serif;
}
.footer-logo{
  color: #4d5ae5;
  font-family: Raleway, sans-serif;
  font-size: 18px;
  font-weight: 700;
  line-height: 1.17; /* 116.667% */
  letter-spacing: 0.03em;
  text-transform: uppercase;
}
.span-footer-logo{
  color: #F4F4FD;
}
.footer-paragraph{
  color: #F4F4FD;
  font-size: 16px;
  font-weight: 400;
}

```

```
line-height: 1.50; /* 150% */  
letter-spacing: 0.02em;  
}
```

## .prettierrc.json

```
{  
  "printWidth": 80,  
  "tabWidth": 2,  
  "useTabs": false,  
  "semi": true,  
  "singleQuote": true,  
  "trailingComma": "es5",  
  "bracketSpacing": true,  
  "jsxBracketSameLine": false,  
  "arrowParens": "avoid",  
  "proseWrap": "always"  
}
```

**ДОДАТОК Б**  
**(обов'язковий)**

**ДІАГРАМИ**

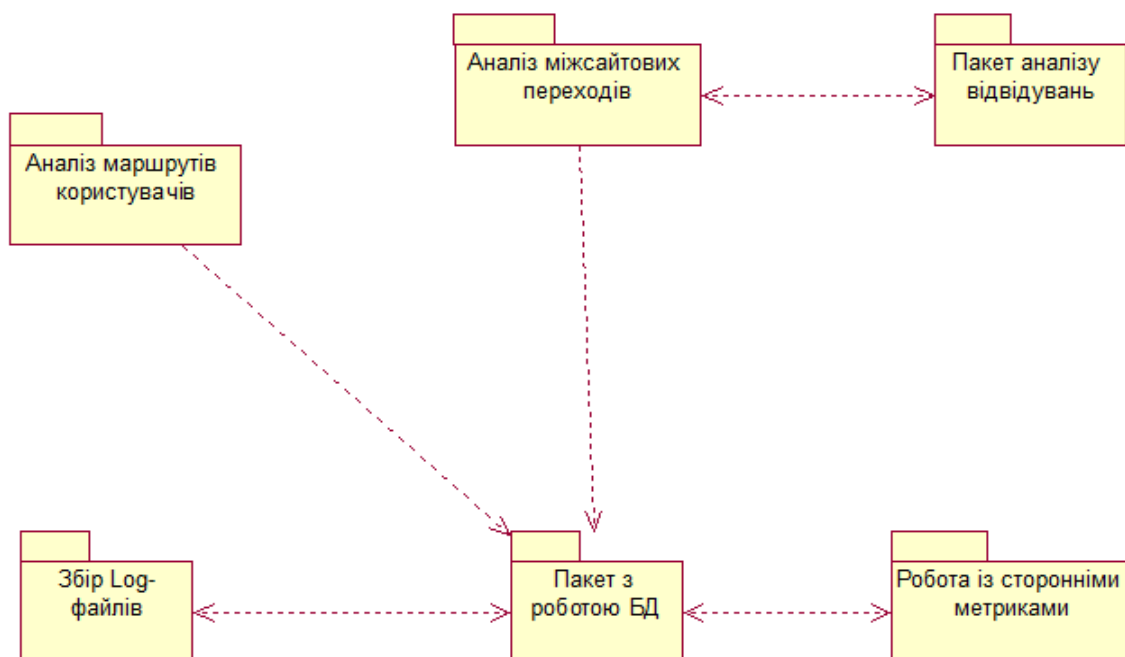


Рисунок Б.1 - Діаграма залежності пакетів

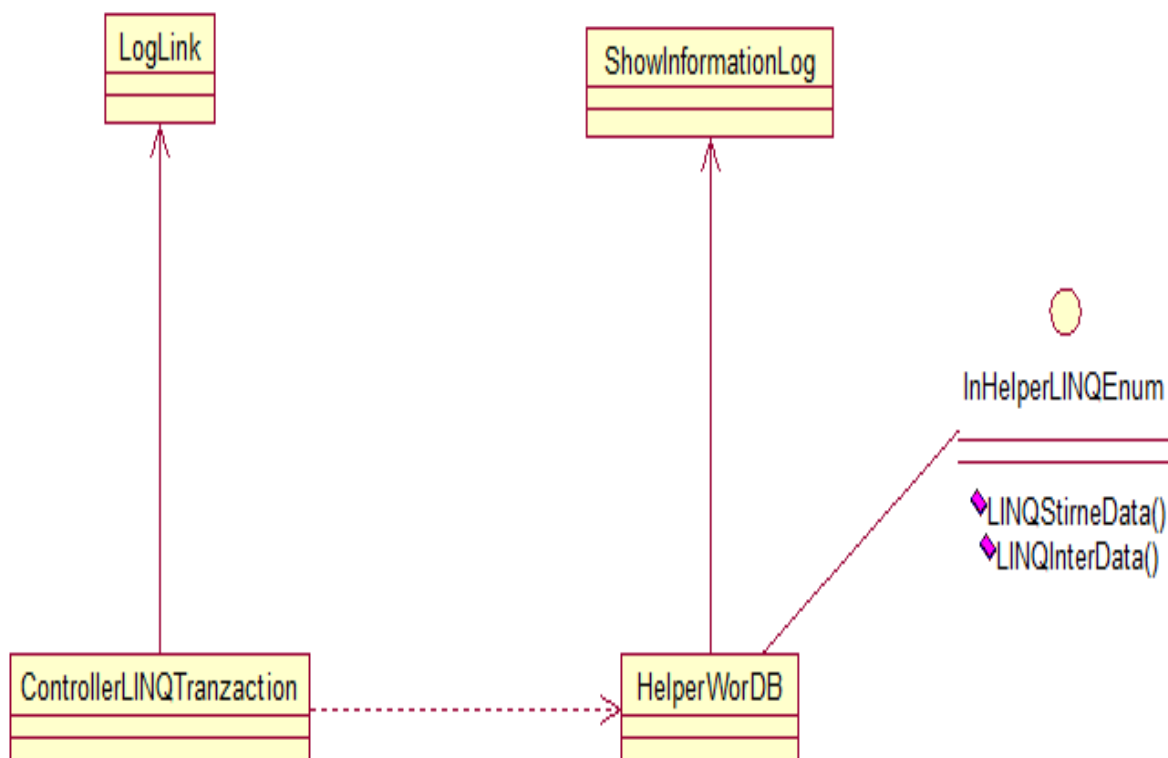


Рисунок Б.2 - Опис інтерфейсів програмного модуля

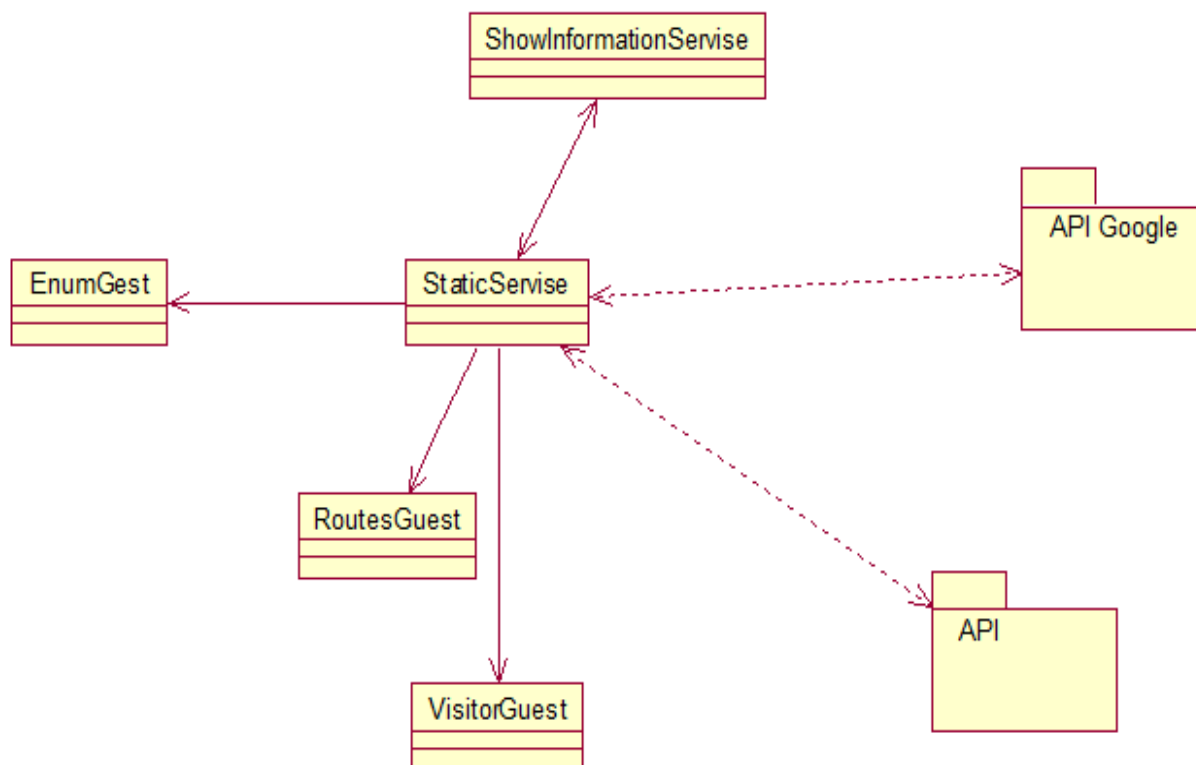


Рисунок Б.3 - Діаграма класів інтерфейсу модуля обробки статистичних даних відвідування

ДОДАТОК В  
(обов'язковий)

**ВАЛІДАЦІЯ**

1. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
From line 4, column 5: to line 4, column 28  
`head> <meta charset="UTF-8" /> <`
2. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
From line 5, column 5: to line 5, column 76  
`8" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> <`
3. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
From line 7, column 5: to line 7, column 65  
`ss"> <link rel="preconnect" href="https://fonts.googleapis.com" /> <`
4. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
From line 8, column 5: to line 8, column 74  
`m" /> <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />`
5. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
From line 11, column 5: to line 11, column 53  
`eet"> <link href="./css/styles.css" rel="stylesheet" /> <`
6. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
From line 78, column 13: to line 83, column 14  
``
7. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
From line 88, column 13: to line 93, column 14

Рисунок В.1 – Валідація файлу index.html

Список правил

Show  source  outline  image report Options...

Check by   CSS

```

color: var(--NAVY-BLUE, #2E2F42);
}
.logo
{
color: var(--IRIS, #4D5AE5);
font-family: Raleway, sans-serif;
font-size: 18px;
font-weight: 700;
line-height: 1.17; /* 116.667% */
letter-spacing: 0.03em;
text-transform: uppercase;
}
span-logo
{
color: var(--NAVY-BLUE, #2E2F42);
}

```

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Warning** This document appears to be written in English. Consider adding `lang="en"` (or variant) to the `html` start tag.  
From line 1, column 1: to line 1, column 17  
`/* Base styles *//* fo`  
For further guidance, consult [Declaring the overall language of a page](#) and [Choosing language tags](#).  
If the HTML checker has misidentified the language of this document, please [file an issue report](#) or [send e-mail to report the problem](#).
2. **Error** Non-space characters found without seeing a doctype first. Expected `<!DOCTYPE html>`.  
From line 1, column 1: to line 1, column 17  
`/* Base styles *//* fo`

Рисунок В.1 – Валідація файлу styleless.css

ДОДАТОК Г  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

Хмельницький Національний Університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

## Кваліфікаційна робота

Тема: Сайт портфоліо вебстудії

Студент: Трохимчук О. В.

Керівник: к.п.н., доцент Онишко О.Г.

### Метою дослідження є:

- Визначення методів та засобів створення сайту портфоліо веб-студії та його розробка
- Реалізація певного середовища, яке буде здійснювати автоматизацію процесів для організації роботи сайту веб-студії
- реалізація інтерфейсу, що забезпечить можливість ефективної, зручної та простої роботи з інформацією про діяльність веб-студії.

## Дослідження має декілька задач , а саме:

---

1. Дослідити категоріально-понятійний апарат дослідження та визначити особливості соціального проєкту як різновиду тематичного сайту-візитки.
2. Провести аналіз ресурсів за темою дослідження
3. Визначити етапи проєктування сайту-візитівки
4. Здійснити аналіз апаратно-програмного забезпечення реалізації сайту- візитки соціального проєкту.
5. Розробити концепцію, дизайн та програмно реалізувати сайт-візитівку соціального проєкту вебстудії

## Опис декомпозиції

---

Поставлену задачу було розділено на такі модулі:

- модуль навігації, що включає в себе інтуїтивно розташовані головне меню, панелі інструментів та контекстні меню для кожного блоку редагування;
- модуль редагування даних містить усі необхідні елементи інтерфейсу, за допомогою яких можна додавати, редагувати та видаляти дані, призначений для зміни інформації у базі даних;
- модуль фільтрації дозволяє шукати інформацію відповідно до запитів користувача та призначений для наглядного відображення даних;
- модуль перегляду включає в себе таблиці та поля з інформацією.

## Аналіз досліджень та існуючих рішень

### Цільова класифікація сайтів:

**Комерційні.** Створюються для прямого отримання продажів з інтернету, створення позитивного образу компанії і розширення охоплення цільової аудиторії. Класичним комерційним сайтом є інтернет-магазин – тут всі елементи створені для того, щоб стимулювати відвідувачів до покупки.

**Інформаційні.** Працюють для передачі певної інформації своєї аудиторії. Це можуть бути новинні, науково-популярні, розважальні або тематичні майданчики. Найчастіше інформаційні сайти створюються з метою їх монетизації за допомогою публікації реклами або партнерських посилань.

**Некомерційні.** До таких веб-ресурсів належать онлайн-майданчики різних державних органів, соціальних служб, навчальних закладів, міністерств і подібних установ. Мета їх створення та існування полягає лише в інформуванні з певних питань.

## Аналіз досліджень та існуючих рішень

### Структурно-функціональна класифікація сайтів:

– **лендінг** – це лише однієї сторінки сайт з потужною рекламної складової. Його завдання – заволодіти увагою відвідувача і привести його до конверсійним дій.

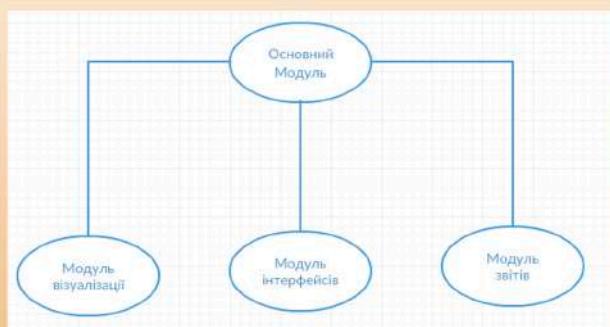
– **сайт-візитка** – це невеликий ресурс з простою структурою. Такі проекти створюються для інформування аудиторії про діяльність компанії або окремого спеціаліста;

– **інтернет-магазин** – спеціальний комерційний формат сайту, що дозволяє зручно вибирати і замовляти товари. У його структуру зазвичай входять кошик, форма зворотного зв'язку, фільтри, вікна з відгуками, кнопка «Купити» тощо;

– **персональний блог** – цей вид сайтів присвячений конкретній людині або його діяльності. Блоги можуть бути розважальними, біографічними, науковими, інформаційними та тощо;

– **сайти послуг** – це ресурси, структура і функціонал яких дозволяє бізнесу зручно і ефективно пропонувати свої послуги.

## Опис залежностей

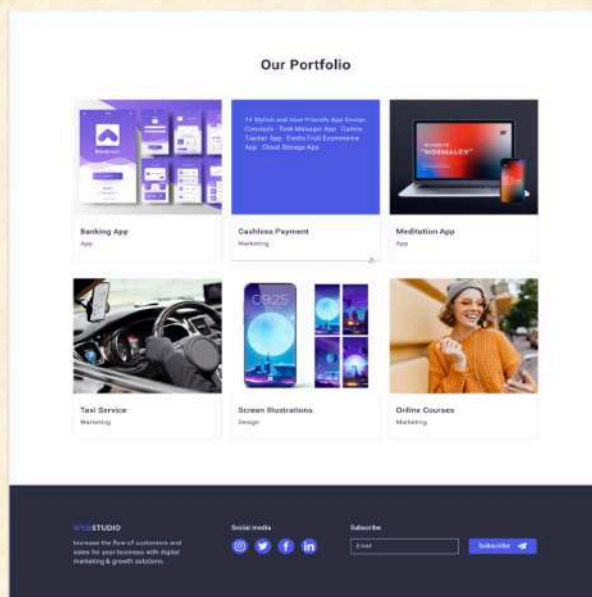


Залежність між модулями пакету Пациент

## Головна сторінка



## Сторінка портфоліо



## Сторінка замовлення

The screenshot shows a contact form on a website. The form is titled "Leave your contacts and we will call you back" and is set against a background image of a desk with laptops and documents. The form fields are as follows:

- Name:** Input field with a person icon.
- Phone:** Input field with a phone icon.
- Email:** Input field with an envelope icon.
- Comments:** Text area with a "Text input" label.

Below the form fields, there is a checkbox labeled "I accept the terms and conditions of the [Privacy Policy](#)". At the bottom of the form is a blue "Send" button.

The website header includes the "WEBSTUDIO" logo, navigation links for "Studio", "Portfolio", and "Contacts", and contact information: "info@devstudio.com" and "+11 (000) 111-11-11".

## Висновок

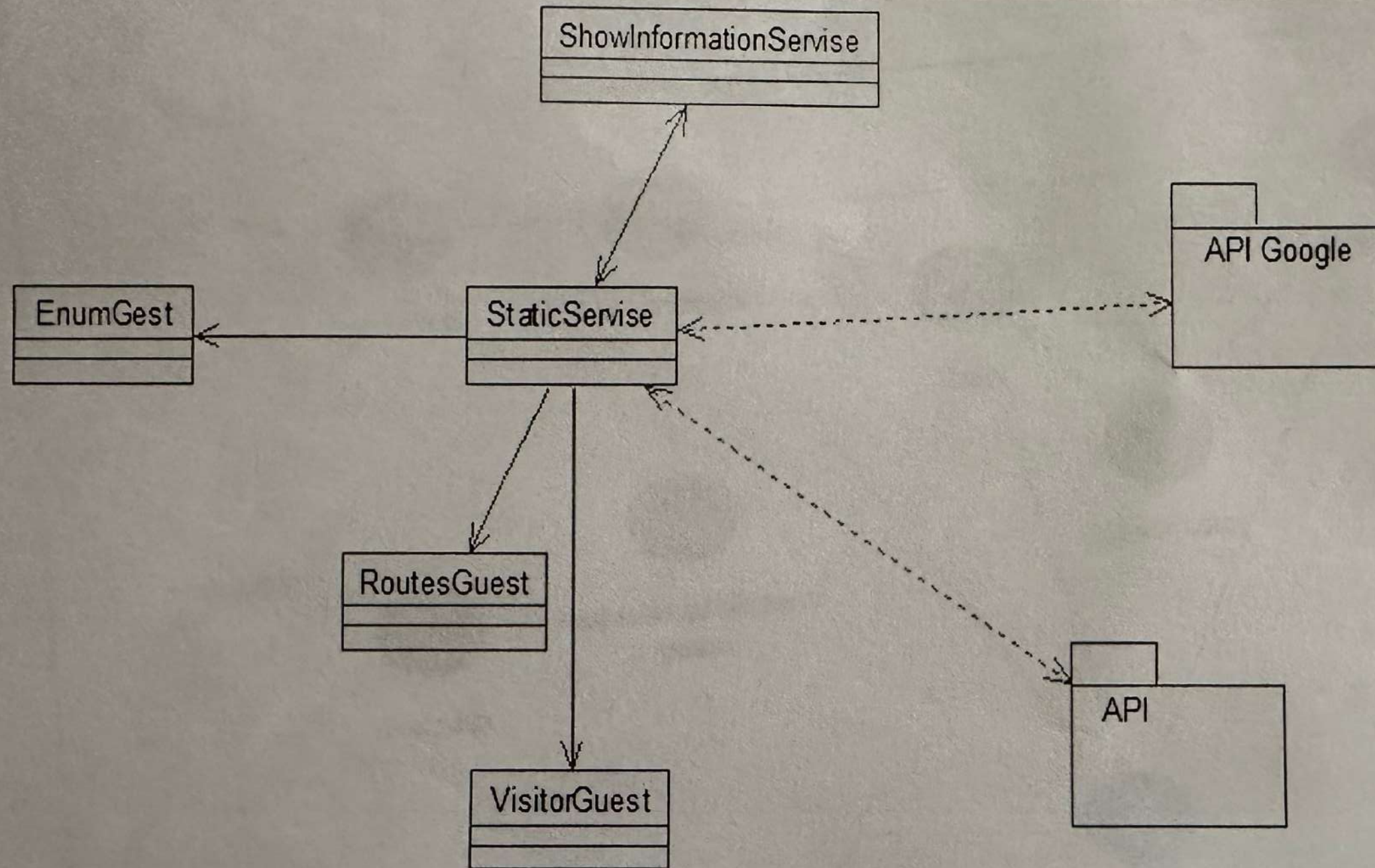
В роботі реалізовано сайт портфоліо вебстудії. Більшість функцій визначених на етапі проектування вдалося реалізувати у повній мірі.

Автоматизація реалізується за рахунок сучасних засобів введення-виведення інформації, реалізації пошуку та фільтрів.

Переваги проекту:

- інтуїтивний інтерфейс користувача;
- зручне додавання, видалення, редагування інформації;
- відсутність прив'язки до певної апаратної платформи;
- відносно низькі системні вимоги.

## **ГРАФІЧНА ЧАСТИНА**



					<b>КвРІПЗ.210104.01.05</b>			
Зм.	Арк.	Надокум.	Підпис	Дата	<b>Сайт портфоліо веб-студії</b>	Літера	Маса	Масштаб
Розробив		Трохимчук О.	<i>[Signature]</i>	10.06				
Керівник		Онишко О.Г.	<i>[Signature]</i>	10.06				
Консульт.						Аркуш	Аркушів	з
Н.Контр.		Яшина О.М.	<i>[Signature]</i>	10.06	<b>ХНУ, ІПЗс-21-1</b>			
Зав.каф.		Бєфратюк Л.П.	<i>[Signature]</i>	10.06				

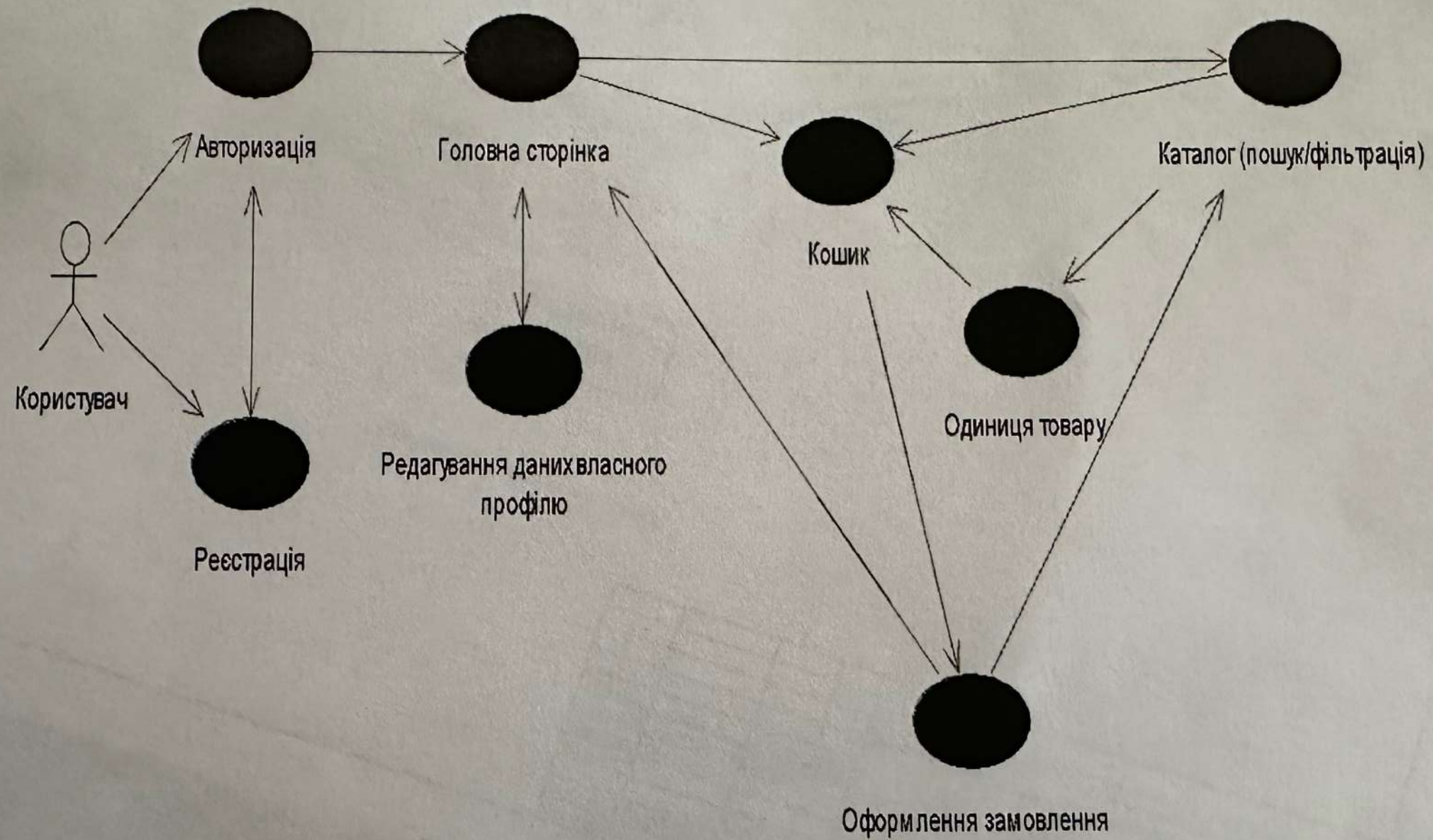


Рисунок 2 – Діаграма використання для користувачів

					<b>КвРІПЗ.210104.01.05</b>		
					Сайт портфоліо веб-студії		
					Літера	Маса	Масштаб
					Аркуш	Аркушів	3
					ХНУ, ІПЗс-21-1		
Зм.	Арх.	Недокум.	Підпис	Дата			
Розробив		Трохимчук О.	<i>[Signature]</i>				
Керівник		Онишко О.Г.	<i>[Signature]</i>				
Консульт							
Н.Контр.		Яшина О.М.	<i>[Signature]</i>				
Зав.каф.		Бєфратюк Л.я	<i>[Signature]</i>				

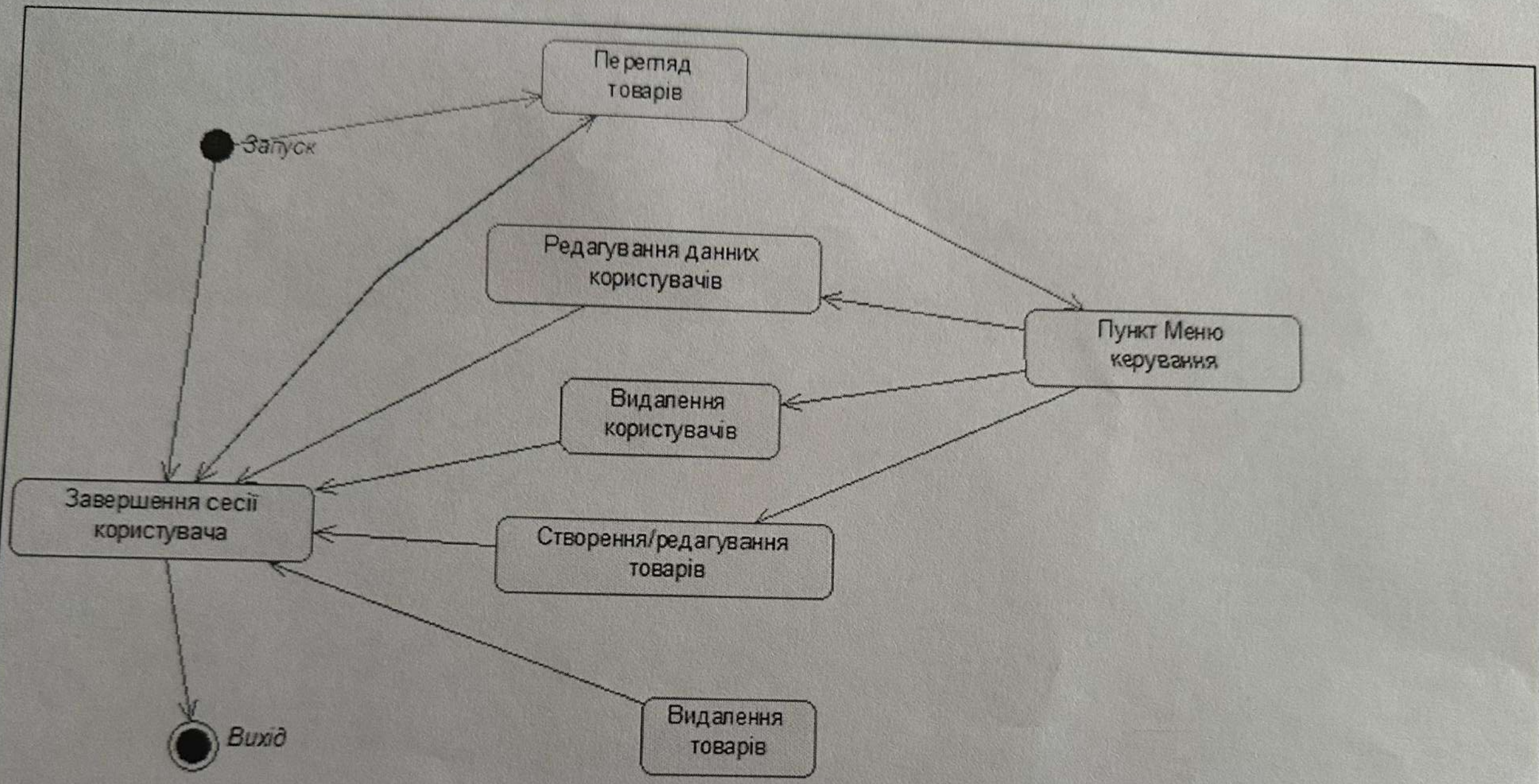


Рисунок 3 – Діаграма станів

					<b>КвРІПЗ.210104.01.05</b>		
					Сайт портфоліо веб-студії		
					Літера	Маса	Масштаб
Зм.	Арк.	Недокум.	Підпис	Дата	Аркуш		
Розробив		Трохимчук О.	<i>[Signature]</i>		Аркушів 3		
Керівник		Онишко О.Г.	<i>[Signature]</i>				
Консульт.							
Н. Контр.		Яшина О.М.	<i>[Signature]</i>		ХНУ, ІПЗс-21-1		
Зав. каф.		Бедратюк Л.Г.	<i>[Signature]</i>				

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

здобувача вищої освіти

Трохимчук О.В

Прізвище, ініціали

факультет ІТ, 3 курс, група ІПЗс-21-1

### ЗАЯВА

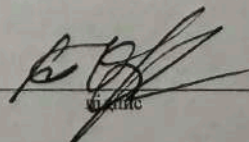
З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті», згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомена. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщена та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного 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 текстових збігів у роботах.

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

01.06.2014

дата

  
місце

## Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 7.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилко в документах: 13%

ID: 129203 Назва: БКР_Сайт портфоліо веб-студії_Трохимчук_Онишко Додано в БД: 2024-06-10 Автора: Трохимчук О.В. Керівники: Онишко О.Г. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	86180	739	6420 (7%)	65 (9%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:  
ІПЗ

Дата перевірки:  
07.06.2024 23:02:09 EEST

Дата звіту:  
08.06.2024 05:48:27 EEST

ID перевірки:  
1016333772

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100012953

Назва документа: БКР\_сайт портфоліо вебстудії\_Трохимчук\_Онишко

Кількість сторінок: 71 Кількість слів: 13254 Кількість символів: 101375 Розмір файлу: 1.35 MB ID файлу: 1016133994

## 15.7% Схожість

Найбільша схожість: 13.9% з джерелом з Бібліотеки (ID файлу: 1015142746)

11.9% Джерела з Інтернету 470 ..... Сторінка 73

14.7% Джерела з Бібліотеки 127 ..... Сторінка 75

## 0.49% Цитат

Цитати 4 ..... Сторінка 76

Посилання 1 ..... Сторінка 76

## 0% Вилучень

Немає вилучених джерел

## Модифікації

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

Замінені символи 5

**РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ  
освітнього ступеня «Бакалавр»**

Дипломник Трохимчук Ольга Вікторівна

Тема Сайт портфоліо вебстудії

Спеціальність 121 – Інженерія програмного забезпечення

**Обсяг кваліфікаційної роботи:**

Кількість листів креслень 4 ; кількість сторінок записки 69

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі досліджено і проаналізовано предметну область, визначено усі функціональні та нефункціональні вимоги. Був проведений аналіз існуючих програм на ринку, розглянуто їх переваги і недоліки, та доведено актуальність розробки нового програмного забезпечення. Розглянуто інструменти для реалізації спроектованих рішень, в результаті чого створено програмне забезпечення. Також було проведено тестування програми, за результатами якого доведено, що розроблене програмне забезпечення працює коректно та готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення. У другому розділі проведено аналіз сучасних архітектур, розглянуто їх переваги і недоліки та визначено, що система буде відповідати монолітній архітектурі та моделі клієнт-сервер. У третьому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт. Також у цьому розділі виконано модульне тестування системи та проведено його у відповідності до функціональних вимог, в результаті було підтверджено коректну роботу програми.

4. Позитивні сторони роботи Основна перевага даної роботи полягає у гнучкості її використання та налаштування, що надає можливість досить ефективно та максимально якісно її використовувати на будь-яких типах пристроїв.

5. Негативні сторони роботи Можна реалізувати більш зручний інтерфейс для наповнення сайту з боку адміністратора.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики проектування. Графічний матеріал дає можливість наочно побачити деталі проектування системи.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ к.т.н. доцент кафедри Комп'ютерної інженерії та інформаційних систем ХНУ

“ 07 ”

серпня

2024 р.

(підпис)

М.В. Канюка

**РІШЕННЯ ЕКСПЕРНОЇ КОМПІЇ  
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продуктованими програмно-технічним засобом (ами) перевірки текстів на плагіат:

Назва: Сайт портфоліо вебстудії

Автор: Трохимчук Ольга Вікторівна

Спеціальність: 121 – Інженерія програмного забезпечення

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

Науковий керівник: Онишко Оксана Григорівна, канд. пед. наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
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 й 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 навмисні 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 кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у 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 оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

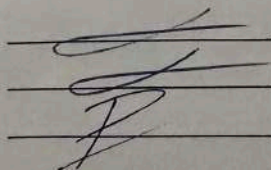
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 15,7% і адресується до титульної сторінки, відомості документів, змісту, заголовків пунктів, літературних джерел та звіту з переддипломної практики, який містить у собі частину кваліфікаційної роботи, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 10.06.2024

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Оксана ОНИШКО