

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Веб-застосунок для продажу будівельних інструментів

Назва теми

Рівень вищої освіти Перший (бакалаврський)

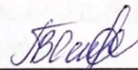
Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр КвРПЗ.190138.01.15.ПЗ

Виконав студент IV курсу, група ПЗ-19-1


Підпис

Пенцак В.О.

Ініціали, прізвище

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


Підпис

Бедратюк Л.П.

Ініціали, прізвище

Нормоконтролер канд.тех.наук., доцент.
Науковий ступінь, звання



Підпис

Гурман І.В.

Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення


Підпис

Бедратюк Л.П.

Ініціали, прізвище

5 4 грудня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

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

ЗАТВЕРДЖУЮ
Завідувач кафедри 173
Л. П. Бедратюк
01 03 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Пенцаку Вадиму Олександровичу
Прізвище, ім'я, по батькові студента

1. Тема кваліфікаційної роботи Веб-застосунок для продажу будівельних матеріалів

Керівник кваліфікаційної роботи Бедратюк Леонід Петрович, д-р фіз.-мат. наук, проф.
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом роботи на кафедру 01.06.2023 р.

3. Вихідні дані до роботи Матеріали переддипломної практики

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

1. Дослідження предметної області

2. Проектування програмного забезпечення

3. Програмна реалізація

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

Три креслення у форматі А3 (UML-діаграма варіантів використання, ER-діаграма бази даних, UML-діаграма послідовностей)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Гурман І.В., канд.тех.наук, доцент	3.06. <i>І.В.Гурман</i>	3.06. <i>І.В.Гурман</i>
Антиплагіат	Гурман І.В., канд.тех.наук, доцент	2.06. <i>І.В.Гурман</i>	2.06. <i>І.В.Гурман</i>

7. Дата видачі завдання « 2 » січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1 Ознайомлення з тематикою кваліфікаційної роботи, визначення та узгодження індивідуальних тем кваліфікаційних робіт(КвР)	01.12 - 31.12.2022	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення(ПЗ), визначення задач та вимог; розробка технічного завдання	02.01-31.01.2023	
3 Проектування програмного забезпечення	01.02-28.02.2023	
4 Програмна реалізація	01.03-10.04.2023	
5 Тестування програмного забезпечення	11.04-30.04.2023	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05-25.05.2023	
7 Попередній захист КвР	травень 2023 (згідно графіка)	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій. Брошування(зшиття) пояснювальної записки	26.05-30.05.2023	
9 Здача КвР на кафедру; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР	3 01.06.2023	

Студент

В.О. Пенцак
Підпис

Пенцак В.О.
Ініціали, прізвище

Керівник роботи

Л.П. Бедратюк
Підпис

Бедратюк Л.П.
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи: «Веб-застосунок для продажу будівельних інструментів».

Автор роботи: Пенцак Вадим Олександрович.

Керівник роботи: Бедратюк Леонід Петрович

Пояснювальна записка: 72с., 41 рис., 13 табл., 3 дод., 40 джерел.

Графічна частина: 3 креслення

Веб-застосунок, інтернет-магазин, ASP.NET MVC, C#, SQL SERVER

Метою кваліфікаційної роботи є розробка веб-застосунку для інтернет-магазину будівельних інструментів, що дозволяє переглядати каталог товарів, здійснювати покупки та оформлення доставки.

У кваліфікаційній роботі визначено специфіку веб-застосунків та інтернет-магазинів; спроектовано логічну модель бази даних, різних модулів серверної частини застосунку та інтерфейсу користувача.

Для реалізації веб-застосунку було застосовано технології платформи ASP.NET, з використанням мови програмування C# та системи керування базами даних SQL SERVER.

У результаті розробки було реалізовано веб-застосунок для продажу будівельних інструментів, проведено тестування та розпочато роботу застосунку.

Підпис студента

В. Пенцак

Дата

05.06.2023

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.190138.01.15.ПЗ	Пояснювальна записка	72		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
5	A3	КвРІПЗ.190138.01.15.E8	UML-діаграма варіантів використання	1		
6	A3	КвРІПЗ.190138.01.15.E8	ER-діаграма бази даних	1		
7	A3	КвРІПЗ.190138.01.15.E8	UML-діаграма послідовностей	1		

					КвРІПЗ.190138.01.15.ВД			
Змн.	Арк.	№ докум.	Підпис	Дата	Веб-застосунок для продажу будівельних інструментів	Літ.	Арк.	Аркушів
Виконав		Пенцак В.О.	<i>[Підпис]</i>	5.06			1	1
Керівник		Бедратюк Л.П.	<i>[Підпис]</i>	5.06	Відомість документів	ХНУ, ІПЗ-19-1		
Н. контр.		Гурман І.В.	<i>[Підпис]</i>	3.06				
Зав. каф.		Бедратюк Л.П.	<i>[Підпис]</i>	5.06				

ЗМІСТ

ВСТУП	4
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	7
1.2 Аналіз наявного програмно-технічного забезпечення предметної області. 11	
1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання.....	16
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	21
2.1 Аналіз та вибір архітектури веб-застосунку	21
2.2 Опис структури даних та моделі бази даних	23
2.3 Проектування серверної частини веб-застосунку	27
2.4 Проектування клієнтської частини веб-застосунку	30
2.5 Аналіз та вибір технологій і методів реалізації веб- застосунку	34
2.6. Висновки до розділу 2	39
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	40
3.1 Розробка бази даних	40
3.2 Розробка програмних модулів	43
3.3 Керівництво користувача	47
3.4 Технічні характеристики веб- застосунку	59
3.5. Аналіз методів тестування веб- застосунку та розробка тестів	60
3.6 Висновки до розділу 3	65
ВИСНОВКИ	66
ПЕРЕЛІК ДЖЕРЕЛ	68
ДОДАТОК А	73
ДОДАТОК Б	79
ДОДАТОК В	93

					КвРІПЗ.190138.01.15.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Веб-застосунок для продажу будівельних інструментів	Літ.	Арк.	Аркушів
Виконав		Пенцак В.О.	<i>В.О.П.</i>	5.06			4	72
Керівник		Бедратюк Л.П.	<i>Л.П.Б.</i>	5.06	Відомість документів	ХНУ, ПЗ-19-1		
Н. контр.		Гурман І.В.	<i>І.В.Г.</i>	3.06				
Зав. каф.		Бедратюк Л.П.	<i>Л.П.Б.</i>	5.06				

ВСТУП

Електронна комерція стрімко набирає своїх оборотів по всьому світу. Статистичні дослідження стверджують про те, що у 2022 році 24% покупок відбувається через Інтернет, при цьому ця частка збільшується останні декілька років. Уже довгий час чітко простежується тенденція того, що нові компанії на ринку продажів надають перевагу входу в сферу саме через онлайн торговельні майданчики, а компанії, що представлені на ринку продажів вже довгий час, все частіше відкривають онлайн майданчики і використовують їх паралельно з своїми фізичними магазинами.

Подібні тенденції також повністю властиві ринку в Україні. Такі зміни у світі комерції є досить послідовними і зумовлюються багатьма факторами. Перший з них – охоплення нових груп клієнтів. Будь-який бізнес будується на цьому, тому логічно, що коли частка онлайн продажів росте з кожним роком, будь-який підприємець зацікавиться виходом на нові ринки.



Рисунок 1 – статистика розміру аудиторії онлайн покупок

Великого поштовху розвитку галузі інтернет магазинів дав Covid-19, адже під час пандемії, у часи, коли відвідування фізичних магазинів суворо обмежувалося, покупки в інтернет магазинах набрали шалених оборотів.

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		4

Наприклад, статистичні дослідження, що охоплювали українські інтернет майданчики для продажу, вказують на цифри у 19 мільйонів відвідувань у травні 2020 року, тобто умовно майже половина населення України відвідали найпопулярніший такий сайт в Україні – Rozetka.com, втім нові обрахунки статистики вказують на те, що наступні роки відвідуваність стрімголов збільшувалася.

Починаючи з лютого 2022 року війна в Україні суттєво змінила перебіг життя українців. Це також вплинуло на трафік інтернет магазинів. Запити користувачів пов'язані тепер не лише з їхніми побутовими потребами, а й із потребами людей у нових реаліях: для військових закупають одяг, взуття, ліки, засоби зв'язку, волонтери також допомагають людям похилого віку та незможним, переселенці та люди, що втратили майно, шукають побутове приладдя першої необхідності.

Сайт	Відвідуваність	
	травень 2021	травень 2022
rozetka.com.ua	23,6 млн	37,8 млн
comfy.ua	2,1 млн	3,5 млн
allo.ua	5 млн	5,9 млн
f.ua	1 млн	963 тис.
prom.ua	19,5 млн	24,3 млн

Рисунок 2 – порівняльна статистика відвідувань сайту у різні роки

Певний вплив на розвиток сфери інтернет-покупок також склав і розвиток сфери доставок. Коли раніше покупцям потрібно було чекати певний період часу, щоб отримати свій товар, і тому вони могли б знехтувати дешевшою ціною в інтернет-магазині, і піти купити товар в фізичному магазині дорожче, але при цьому отримати товар зразу, то зараз темпи доставок настільки швидкі, що клієнт може отримати свій товар буквально на наступний день.

Актуальність теми полягає у тому, що на теперішній час, більшість підприємців взяли курс на електронну комерцію, тобто перехід до введення свого бізнесу саме за допомогою інтернет-магазинів. Попит на створення таких

ресурсів не спадає, а навіть продовжує рости. Об'єктивний погляд дає зрозуміти, що є потреба створення такого програмного забезпечення, адже його створення також є комерційно привабливим, бо таким чином дає змогу отримати нових клієнтів, а також дає змогу дещо модернізувати бізнес-процеси підприємства.

Метою проекту є розробка ПЗ, що дозволить оптимізувати роботу підприємства. Також мета полягає у тому, що створене ПЗ повинно мати сучасний та інтуїтивно-зрозумілий інтерфейс, що забезпечить продаж товарів для клієнтів та систематизує взаємодію продавця та покупця.

Реалізація ПЗ вимагає виконання певних задач:

- визначення актуальності теми та її практичного застосування;
- дослідження наявних методів вирішення та їх аналіз;
- визначення вимог до ПЗ та функціоналу, що має бути реалізовано;
- дослідження використання різних типів баз даних в ПЗ;
- вибір та обґрунтування архітектури ПЗ;
- аналіз методів реалізації ПЗ;
- опис особливостей та функціоналу для інтерфейсу ПЗ;
- здійснення тестування ПЗ;

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		6

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

В сучасних реаліях, у часи, коли велика частина покупок здійснюється дистанційно, тобто за допомогою інтернет шопінгу, покупці щодня стикаються з питанням, який інтернет-майданчик вибрати, на якому сайті здійснити покупку буде легко і швидко, який сайт надає максимально якісний рівень обслуговування та надає найвищий рівень безпеки конфіденційних даних користувачів.

Попит на різні види сайтів для продажу росте величезними темпами, адже оцінити вклад, який вони вносять в індустрію важко, так як з своєю появою, вони запроваджували в сферу величезну кількість нововведень, що колосально покращують сферу покупок.

Розробка сайтів складний та багаторівневий процес, кожен з яких взаємозв'язаний і унеможлиблює виключення якогось із них. Для того, щоб розроблюване ПЗ було якісне і могло конкурувати з іншими представниками з обраної галузі, слід визначити усі потреби кінцевих користувачів, функціональні та нефункціональні вимоги ПЗ та інші критерії, такі як, наприклад, тип сайту або дизайн інтерфейсу.

Аналіз можна розпочати з такого аспекту, як огляд ринку будівельних матеріалів, адже вивчення стану ринку, динаміки та перспектив розвитку дасть змогу зрозуміти, які основні тенденції та потреби є у споживачів, що дозволить адаптувати, наприклад, асортимент інтернет-магазину.

Не менш важливим є огляд технологій та систем управління інтернет-магазинами. Провівши аналіз систем управління або систем оплати чи доставки, дасть зрозуміти, які саме технології можна буде застосувати в розроблюваному ПЗ.

Найбільш важливим аспектом є аналіз потенційних користувачів розроблюваного ПЗ. Визначення основних потреб споживачів, визначення

					КвРПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

основних груп, адже вони можуть коливатися від звичайних роздрібних покупців до великих компаній, що потребують великі гуртові покупки для своїх проектів. Кожна з груп матиме свої вимоги та очікування щодо асортименту товарів, рівня цін, умов доставки та оплати. Тому важливо здійснити аналіз потреб та очікувань різних груп клієнтів, щоб забезпечити максимально ефективну роботу інтернет магазину та задовольнити потреби різних груп клієнтів.

Вибір типу сайту є важливим аспектом при створенні ПЗ. Існує багато різних типів, кожен з яких є влучним рішенням для різних задач. У випадку коли слід обирати тип сайту, на якому будуть відбуватися інтернет-покупки, можна виділити три типи сайтів: сайт-каталог, односторінковий сайт або інтернет-магазин. Кожен з них чудово підходить для інтернет-покупок, але який саме тип слід вибрати можна визначити, проаналізувавши інші аспекти, що варто реалізувати на сайті і вже виходячи з цього, зрівнявши вимоги можна обрати найбільш доцільний тип сайту.

Аналіз функціональних – це останній етап, який слід детально дослідити, адже це дасть змогу визначити основні функції, які має виконувати майбутнє ПЗ, таких як оформлення замовлень, оплата, управління асортиментом тощо. Цей аналіз дозволить розробити детальну специфікацію вимог до майбутнього ПЗ та визначити його функціональні можливості. У випадку з технічними вимогами, це дасть змогу визначитися з такими критеріями, як платформи, мови програмування, бази даних, інтерфейс користувача та інших технічних параметрів.

На основі результатів аналізу можна визначити проблему, що буде вирішена за допомогою майбутнього ПЗ. Одна з головних проблем, яку можна вирішити за допомогою інтернет-магазину - це забезпечення зручності та швидкості процесу купівлі. Традиційно, покупці змушені відвідувати фізичні магазини, знайомитися з асортиментом та цінами, обговорювати деталі з продавцем, витрачаючи на це значну кількість часу та зусиль. За допомогою інтернет-магазину, клієнти можуть знайти та купити потрібні товари в зручний для них час та з мінімальними зусиллями. Також, інтернет-магазин може

					КВРПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

допомогти забезпечити надійний та безпечний процес покупки та збереження персональних даних клієнтів, дотримуючись усіх вимог та стандартів безпеки даних.

Вивчення предметної області дозволяє визначити основні проблеми та не вирішені питання з точки зору впровадження інформаційних технологій, автоматизації виробничих процесів, процесів опрацювання та передачі інформації, а також з'ясувати функціональні особливості, що необхідні для розробки майбутнього програмного забезпечення.

Оскільки звичайний клієнт буде основним користувачем, необхідно забезпечити функціонал, який дозволить переглядати та купувати товари. Однак, слід також враховувати адміністрування сайту, тому потрібно розглянути додатковий функціонал для адміністраторів, що можливий завдяки реєстрації користувачів та наданню їм певних функцій.

Крім основного функціоналу, можна розробити функції, які дозволять переглядати вибрані товари та здійснювати замовлення, проте для цього потрібна реєстрація користувача. Важливо ретельно розподіляти функціонал між користувачами, щоб кожен міг отримати те, що саме йому потрібно.

Отже, після проведення змістовного аналізу можна виділити декілька кілька задач, які слід вирішити при створенні ПЗ:

- каталог будівельних матеріалів: ПЗ повинно містити детальний каталог з різноманітними будівельними матеріалами, включаючи опис, характеристики, фотографії та ціни. Клієнти повинні мати можливість швидко знайти та порівняти різні товари;

- реєстрація користувача та його особистий кабінет: ПЗ повинно мати функціонал, за допомогою якого кожен окремий користувач матиме свій окремий профіль, у якому він зможе відслідковувати обраний товар і стежити за перебігом виконання замовлення;

- управління замовленнями та оплатою: ПЗ повинно мати функціонал для зручного оформлення замовлень, включаючи вибір кількості та доставки;

					КВРПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

– панель адміністратора: ПЗ повинно мати функціонал, за допомогою якого адміністратор матиме змогу додавати, редагувати або видаляти різні записи в каталозі;

– інтуїтивний та зручний інтерфейс користувача: ПЗ повинно мати простий та зрозумілий інтерфейс, що дозволить клієнтам швидко знайти необхідні товари та здійснити покупку без зайвих зусиль;

Для того, що детальніше описати та формалізувати бізнес-процеси підприємства, можна скористатися модельними представленнями. У цьому випадку, можна скористатися IDEF0 діаграми. IDEF0 діаграми можна використовувати для моделювання бізнес-процесів та структури інтернет магазину. Зокрема, можна побудувати IDEF0 діаграму для опису процесу замовлення товарів, діаграму взаємодії між клієнтами та продавцями, а також діаграму структури сайту та його функціональних модулів. Використання IDEF0 діаграм дозволить краще зрозуміти логіку роботи магазину та ідентифікувати можливі проблеми та перешкоди в процесі роботи, що дозволить покращити ефективність та результативність бізнесу.

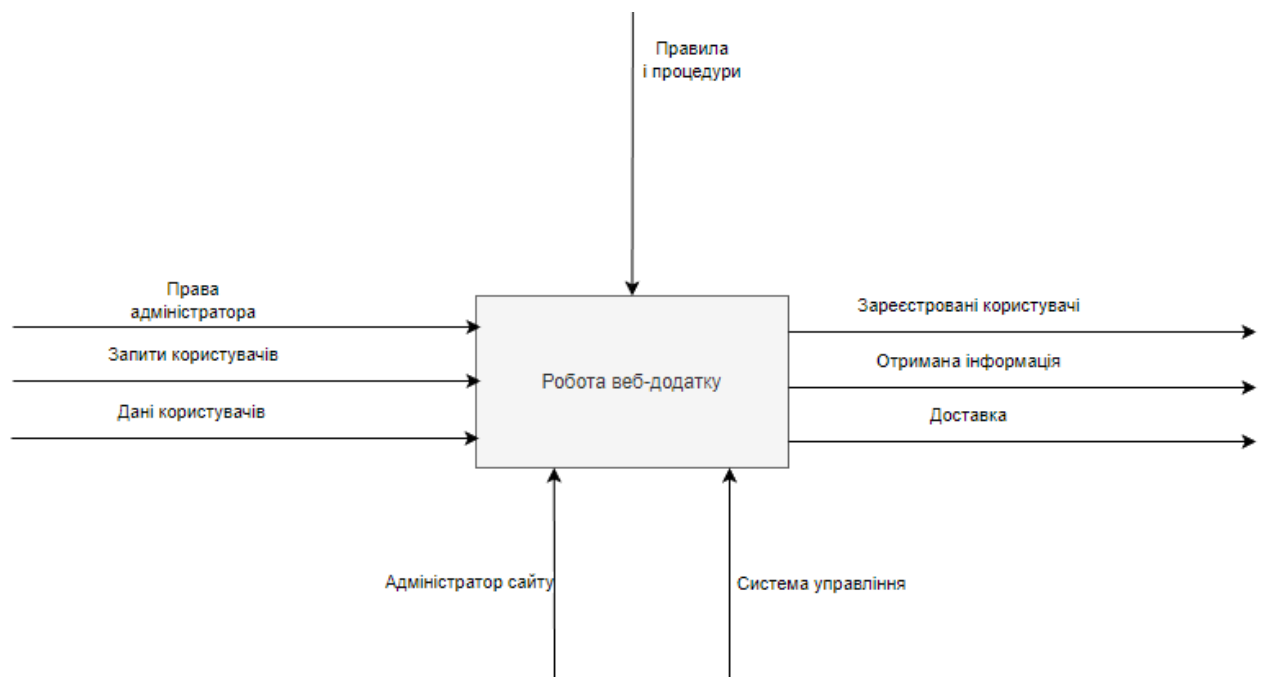


Рисунок 1.1.1 - контекстна IDEF0-діаграма роботи програмного забезпечення

На зображенні 1.1 наведено контекстну діаграму онлайн-магазину, де потоки показують, звідки надходять дані та основну інформацію, яка виходить. Вхідні дані включають інформацію, що надходить від користувачів, їх реєстраційні дані, запити до системи щодо даних про конкретні товари або загальну інформацію.

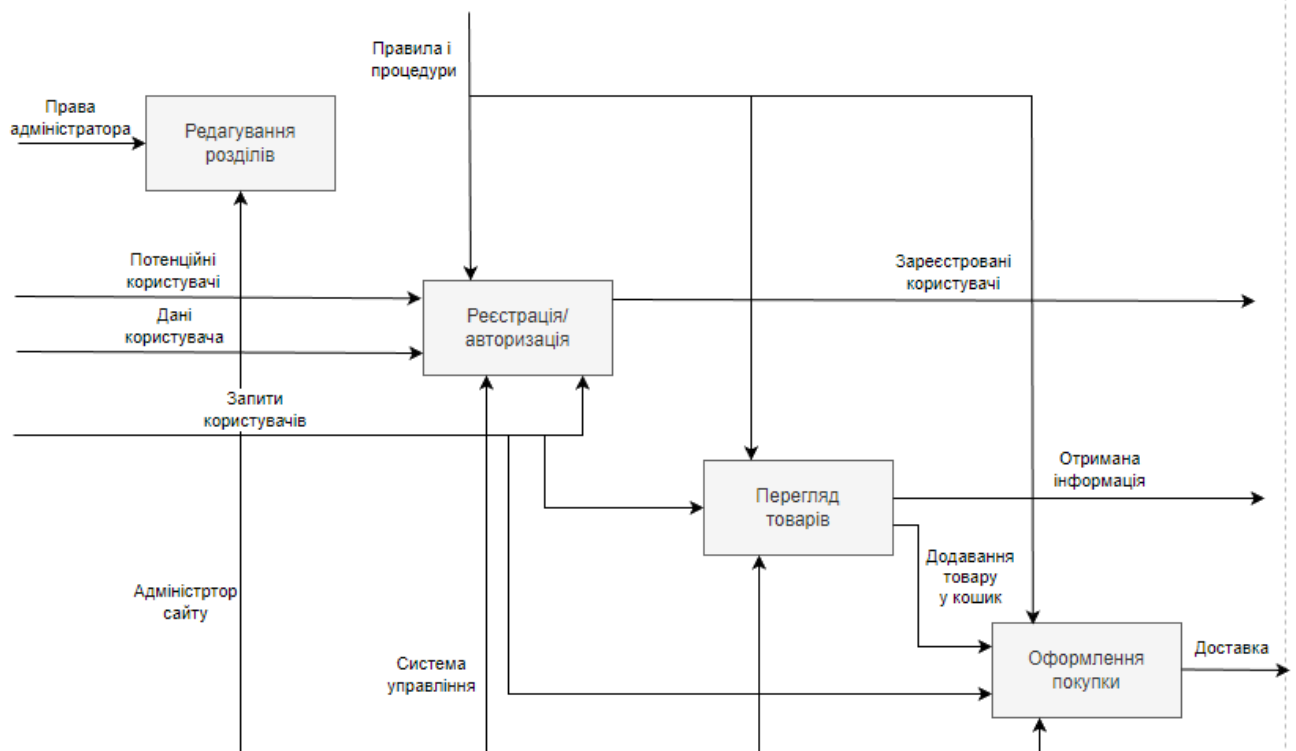


Рисунок 1.1.2 - IDEF0-діаграма декомпозиції першого рівня

На рисунку 1.2 зображена діаграма декомпозиції першого рівня, що розкриває основні функціональні особливості системи. Відповідно, можна виділити базовий функціонал програмної системи.

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Починаючи роботу над новим проектом перше з чого слід розпочати – це аналіз усіх наявних рішень, провести їх дослідження та порівняння. З цього слід

виділити їх переваги та недоліки, щоб врахувати це у своєму проекті. Сфера інтернет-магазинів має велику кількість представників, тому інформації для аналізу буде багато, щоб отримати розуміння того, яким має бути проект.

Перше, що можна відмітити, це різницю, між основними представниками сфери будівельних інструментів. Одна частина сайтів – це вузько-направлені інтернет-магазини, яких увесь асортимент складається саме з інструментів. Друга частина – маркетплейси, тобто величезні інтернет-майданчики, на яких можна знайти буквально все. В багатьох з них є розділи з будівельними інструментами, які по своїй суті виглядають, як маленький інтернет-магазин в середині величезного інтернет-магазину. Для того, щоб дослідити і з'ясувати, які рішення є вдалим і які з них слід запозичити або навпаки, з'ясувати, що є не дуже вдалим рішенням, варто проаналізувати і вузько-направлені інтернет-магазини, і маркетплейси, на яких є розділи, що відносяться до обраної теми.

Аналіз проводиться не лише для того, щоб з'ясувати, які функціональні рішення слід запозичити, а які варто уникати, але ще й для того, щоб дослідити дизайнерські рішення, які стосуються інтерфейсу користувачів.

Статистичні дослідження, що проводилися серед звичайних користувачів інтернет-магазинів доводять, що в першу чергу користувач буде шукати потрібну йому річ на сайті, який привернув його увагу з перших секунд. Якщо за перших декілька секунд сайт не привернув уваги користувача, то скоріше за все, він не продовжить пошук потрібної йому речі. Саме тому слід відмітити, що для утримання користувача потрібно:

- правильно підібрати дизайн сайту, щоб він був приємним для ока;
- додати максимальну кількість інформації, але при цьому зберегти баланс, щоб сайт не виглядав «загромадженим»;
- зробити інтерфейс інтуїтивно-зрозумілим, для того, щоб користувач міг одразу зорієнтуватися;

Після введення пошукового запиту у браузері, перший сайт, який з'являється у пошуку - <https://storgom.ua/ua>. Річ, яку зразу варто відмітити – це достатньо простий і зрозумілий інтерфейс сайту. Дизайн виконаний у

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

мінімалістичному стилі з використанням світлих кольорів. В цілому поєднання досить простого інтерфейсу та вибір дизайну сайту можна вважати вдалим рішенням, яке точно надає перевагу, коли пересічний клієнт обирає інтернет-магазин для здійснення своїх покупок.

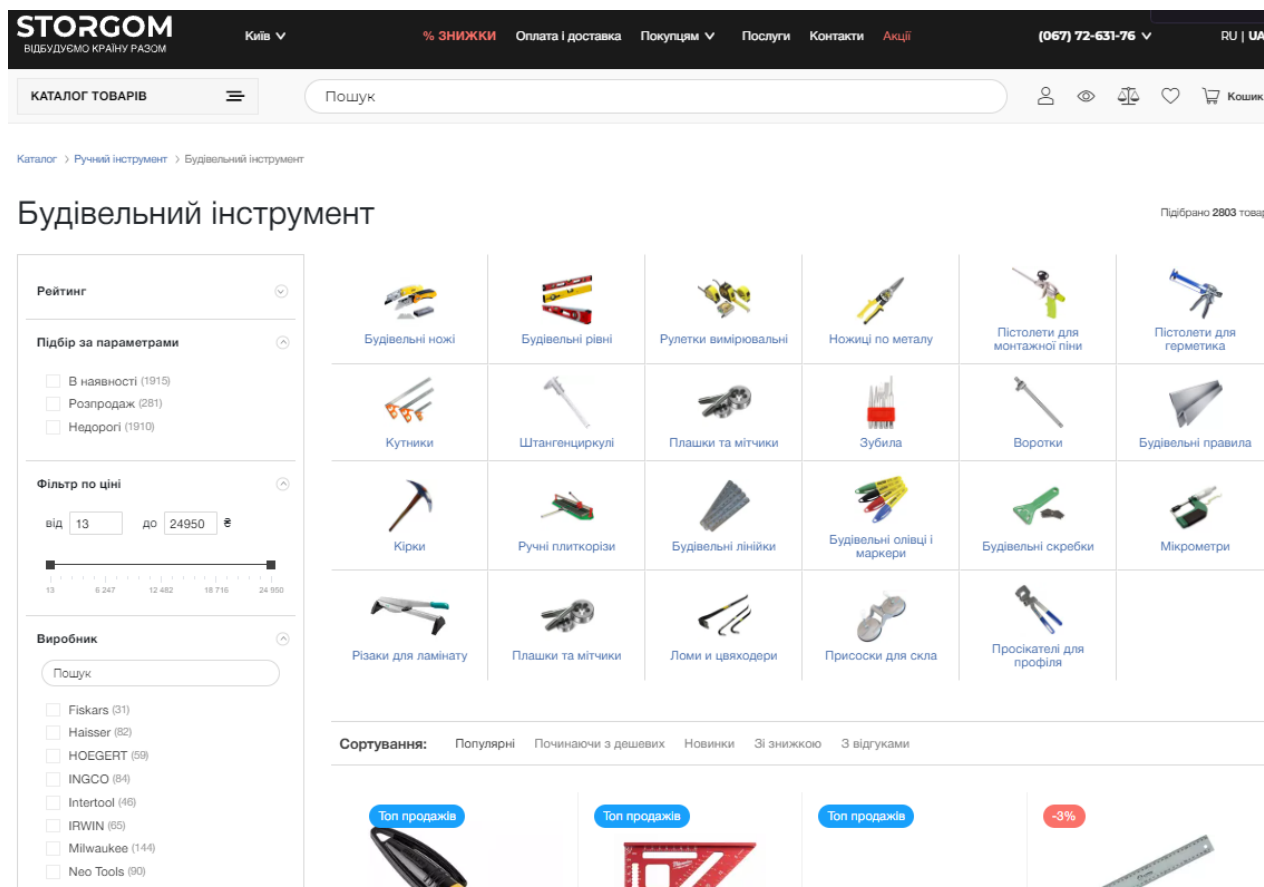


Рисунок 1.2.1 – сайт Storgom – <https://storgom.ua/ua>

На головній сторінці користувач бачить перед собою розгорнуту панель з категоріями інструментів, асортимент яких представлений в магазині. Одразу нижче після цього відображаються інструменти, які є найпопулярнішим вибором серед покупців магазину, при цьому, цей асортимент можна розподілити по фільтрах. Сама панель з фільтрами знаходиться у лівій частині сторінки, займає невеликий обсяг місця, але при цьому надає досить широкий вибір фільтрів для користувача, тому цей аспект точно можна віднести до переваг даного інтернет-магазину. У шапці сайту знаходиться уся контактна інформація для зв'язку з представниками магазину, а також зручна панель для авторизації в особистому профілі, перегляду обраних товарів і іншими корисними функціями. Підводячи підсумки можна сказати, що даний представник інтернет-магазинів будівельних

інструментів має досить багато переваг, має досить зручний і зрозумілий інтерфейс, дизайн сайту виконаний у доречному стилі.

Наступний сайт, який відображається після введення пошукового запиту - <https://ars.ua>. Одразу, що можна підмітити – простий, проте досить добре підібраний дизайн сайту. Дизайн дещо схожий з попереднім представником, тому слід відмітити, що певні збіги можна вважати влучною тенденцією, яку використовують одні з найпопулярніших представників тематичних інтернет-магазинів. Центр екрану займає панель з основними категоріями, ліворуч відображається більш розширена панель з всіма категоріями інструментів, які є в асортименті магазину. В загальному про інтерфейс можна сказати те, що він досить інформативний та інтуїтивно-зрозумілий, тому більшість користувачів не матиме труднощів у користуванні сайтом. Навігація на сайті реалізована добре. Кольорова гама досить світла, без темних і надто яскравих кольорів.

У верхній частині сайту традиційно відображається інформація та контактні дані представників магазину. Також там присутня зручна стрічка пошуку та панель з якої можна перейти до авторизації або перегляду обраних товарів. В цілому сайт є досить вдалим рішенням, тому при вирішенні поставленої задачі можна звернути увагу на його переваги.

На основі аналізу була створена таблиця, що містить зведену інформацію про основні рішення, на які слід звертати увагу при реалізації проекту.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		14

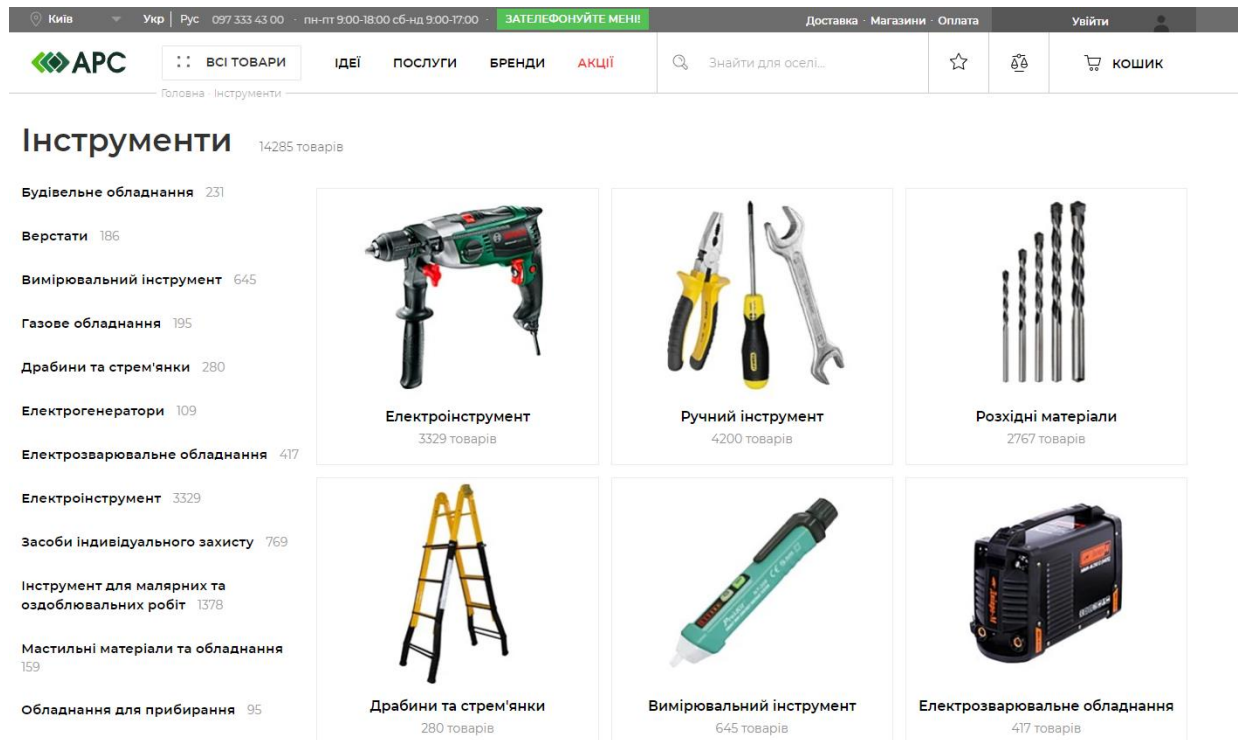


Рисунок 1.2.2 – сайт Ars – <https://ars.ua>

На основі проведеного аналізу існуючих рішень було створено порівняльну таблицю для визначення основних критеріїв аналізу.

Таблиця 1.1.1 – порівняльна таблиця проаналізованих сайтів

Назва	Можливість реєстрації	Можливість пошуку через стрічку	Інтерфейс сайту	Зручність в користуванні функціоналом
strogom.ua	Так	Так	Простий, мінімалістичний, зрозумілий	Досить простий і зручний функціонал
https://ars.ua	Так	Так	Мінімалістичний, інтуїтивно-зрозумілий	Досить простий для використання

1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання

Визначити вимоги до програмного забезпечення можна за допомогою використання UML діаграм.

UML (Unified Modeling Language) - це мова моделювання, яка використовується для візуального представлення структури, поведінки та взаємодії систем.

UML діаграми - це графічні представлення різних аспектів системи з використанням символів та взаємозв'язків, визначених в UML. Вони дозволяють візуалізувати різні аспекти системи, такі як структура класів, взаємодія між об'єктами, поведінка системи та багато іншого.

Існує багато різних UML діаграм: діаграми класів, діаграми послідовностей, діаграми варіантів використання, тощо. Для того, щоб проаналізувати та визначити усі вимоги до програмного забезпечення, а також розробити технічне завдання, варто скористуватися діаграмою варіантів використання. Для цього слід визначити усіх акторів програмного забезпечення.

Таблиця 1.3.1 – актори програмного забезпечення

Актор	Опис
Неавторизований користувач	Може переглядати усі товари та інформацію про них. Має можливість авторизуватися або провести реєстрацію, якщо він ще не має персонального профілю.
Авторизований користувач	До перерахованих вище можливостей, що має неавторизований користувач також може додавати товари до кошику, може редагувати вміст кошику, може проводити оформлення замовлення. Також, кожен авторизований користувач може переглядати свій профіль, а також змінювати персональні дані.

Продовження таблиці 1.3.1

Адміністратор	Може додавати нові записи (створення нових товарів або категорій) на сайті, редагувати та видаляти уже існуючі записи. Може переглядати інформацію про стан усіх замовлень. Може редагувати усі замовлення або у випадку, якщо це потрібно повністю його відмінити.
---------------	---

У наступній таблиці описуються усі варіанти використання для усіх акторів програмного забезпечення

Таблиця 1.3.2 – варіанти використання та їх опис

Актор	Варіант використання	Опис
Неавторизований користувач	Авторизація в системі	Якщо користувач уже має свій особистий профіль, він може здійснити авторизацію на сайті за допомогою особистих даних профілю
Неавторизований користувач	Реєстрація в системі	Якщо користувач ніколи раніше не мав особистого профілю, він може зареєструватися на сайті за допомогою електронної скриньки, після чого він матиме змогу авторизуватися на сайті
Неавторизований користувач авторизований користувач	Перегляд усіх товарів	Будь-який користувач, який відвідує сайт має змогу переглядати товари та будь-яку іншу інформацію про товари, які представлені в каталозі

Продовження таблиці 1.3.2

Авторизований користувач	Додавання товару у кошик	Після авторизації на сайті користувач отримує змогу додавати товари до власного кошику
Авторизований користувач	Редагування вмісту кошику	Якщо авторизований користувач має збережені у кошику товари, він може або змінити їх кількість або видалити товар прямо з кошику
Авторизований користувач	Оформлення замовлення	Якщо користувач має якийсь товар у кошику, він може оформити замовлення після того, як введе дані для відправки та підтвердить оформлення
Авторизований користувач	Зміна даних особистого профілю	Користувач може змінити будь-які дані свого особистого профілю(ім'я, прізвище, пароль, тощо)
Адміністратор	Додавання нових записів	Користувач може додавати нові записи про товари у каталозі
Адміністратор	Редагування або видалення існуючих записів	Користувач може відредагувати вже існуючий запис про товар або взагалі видалити його з каталогу товарів
Адміністратор	Перегляд замовлень	Користувач може відслідковувати статус кожного замовлення для того, що затвердити відправку або позначити замовлення як виконане

Кінець таблиці 1.3.2

Адміністратор	Редагування або видалення замовлень	Користувач може відредагувати оформлене замовлення для того, щоб змінити перелік замовлених товарів, зробити переадресацію або видалити замовлення у випадку, коли його не потрібно виконувати
---------------	-------------------------------------	--

На основі двох таблиць з описом акторів програмного забезпечення та з описом варіантів використання можна створити діаграму варіантів використання (рисунок 1.3.1)

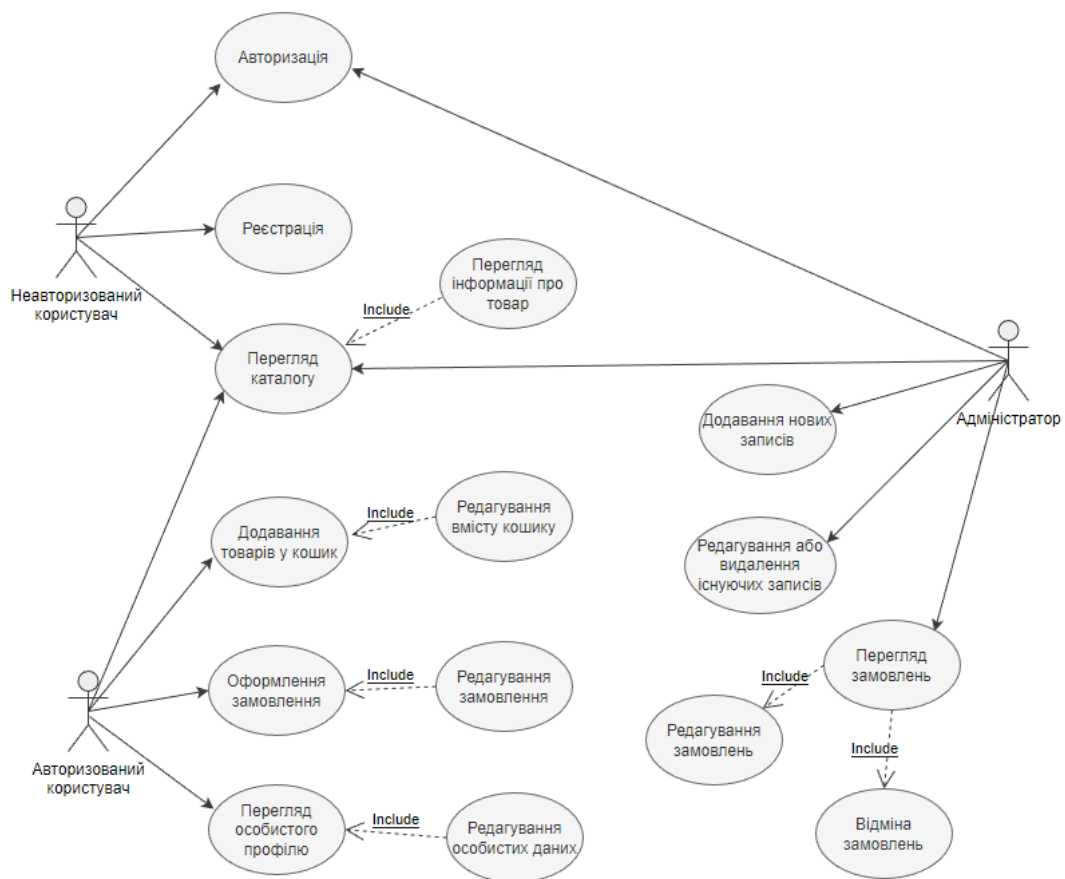


Рисунок 1.3.1 – діаграма варіантів використання

1.5 Висновки до розділу 1

Отже, за результатами цього розділу було проведено дослідження предметної області, виконано аналіз існуючих рішень (проаналізовано декілька найбільш популярних веб-застосунків зв'язаних з темою кваліфікаційної роботи). Побудовано таблицю порівняння існуючих рішень. Також було визначено основні вимоги до програмного забезпечення, визначено основних акторів ПЗ і на основі цього побудовано діаграму варіантів використання. Також, заключним етапом написання розділу 1 стало написання технічного завдання (додаток А).

Поставлені задачі:

- реалізація каталогу товарів;
- реалізація реєстрації та авторизації користувача;
- реалізація налаштування особистого профілю користувача;
- реалізація кошику товарів;
- реалізація оплати;
- реалізація функціоналу керування сайтом для адміністратора (додавання нових записів, редагування та видалення існуючих);
- реалізація функціоналу для керування та відслідковування замовлень для адміністратора;

Інтерфейс користувача має бути простим та інтуїтивно-зрозумілим, функціонал має бути максимально легким у застосуванні як для звичайного користувача, так і для адміністратора.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз та вибір архітектури веб-застосунку

Один із основних етапів розробки програмного забезпечення є етап його проектування. Саме на етапі проектування відбувається вибір архітектури програмного забезпечення. Вибір типу архітектури програмного забезпечення залежить від різних факторів: конкретні потреби проекту (безпека та ефективність), його масштабованість, легкість розширення та підтримки системи та інше. Основні типи архітектури веб-застосунків: клієнт-серверна, мікросервісна та монолітна.

Клієнт-серверна архітектура - розділення функцій між клієнтами і серверами, де клієнти звертаються до серверів для отримання послуг і ресурсів.

До переваг можна віднести:

- масштабованість;
- розділення обов'язків;
- безпека;
- інтегрований інтероперабельність;

До недоліків можна віднести:

- залежність від доступності сервера;
- потреба у розробці клієнтської і серверної частин окремо;

Мікросервісна архітектура - розбиття функціональності на незалежні сервіси, які взаємодіють через мережу.

Переваги мікросервісної архітектури:

- гнучкість;
- швидка розробка;
- легкість впровадження нових функцій;

Недоліки мікросервісної архітектури:

- складність управління декількома сервісами;
- вимога до забезпечення безпеки та надійності зв'язку;

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

Монолітна архітектура - усі компоненти додатку розміщені в одному монолітному коді.

Переваги даного типу архітектури:

- простота розробки і розгортання;
- менша складність управління системою;

Недоліки даного типу архітектури:

- складності розширення;
- обмежена масштабованість;
- великий ризик залежності та проблем з підтримкою;

Кожен тип архітектури має свої особливості і підходить для різних сценаріїв. Клієнт-серверна архітектура забезпечує централізований контроль, розділення обов'язків і використовується для великих проектів з великим навантаженням. Мікросервісна архітектура надає гнучкість, швидку розробку та масштабованість і підходить для складних, розподілених систем. Монолітна архітектура є простою у розробці, але може бути обмежена у масштабованості та розширенні.

Клієнт-серверна архітектура є однією з найпоширеніших архітектур для розробки програмного забезпечення (ПЗ). У цій архітектурі функції та обов'язки розподіляються між двома основними складовими: клієнтами і серверами. Клієнти — це користувачі або програми, які звертаються до сервера для отримання певних послуг або ресурсів. Сервери — це компоненти, які надають ці послуги або ресурси клієнтам. Саме цей тип архітектури було обрано для реалізації проекту, так як клієнт-серверна архітектура підходить для сценаріїв, де потрібно забезпечити централізований контроль, розподілення ресурсів та масштабованість серверів. Вона також ефективна, коли потрібна безпека, взаємодія з різними типами клієнтів та оптимальне використання мережевих ресурсів.

Детальний опис переваг клієнт-серверної архітектури:

- масштабованість: архітектура дозволяє гнучко масштабувати систему шляхом додавання або заміни серверів без впливу на клієнтську сторону

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

- розділення обов'язків: клієнтська і серверна сторони виконують різні функції, що дозволяє розподілити ресурси і забезпечити ефективне виконання задач
- забезпечення безпеки: застосування клієнт-серверної архітектури дозволяє реалізувати заходи безпеки, такі як автентифікація і авторизація, на рівні сервера
- інтеоперабельність: клієнт-серверна архітектура підтримує різноманітність клієнтів, таких як комп'ютери, мобільні пристрої або веб-браузери, що дає можливість розробляти універсальні рішення

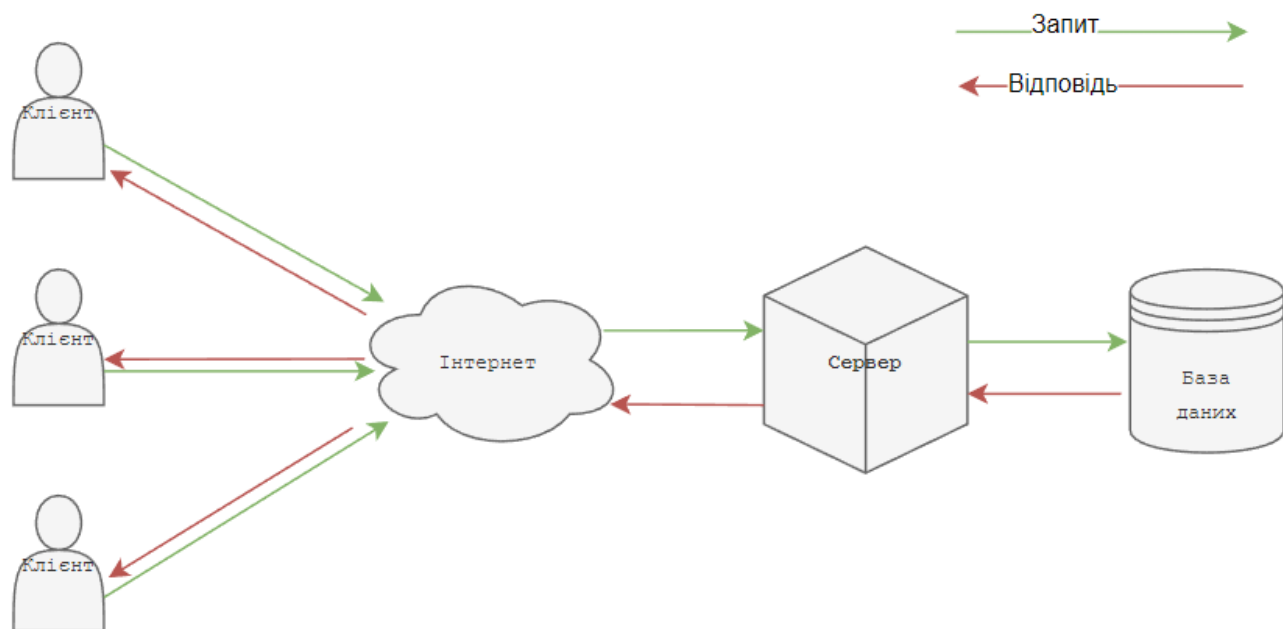


Рисунок 2.1.1 – схематичний рисунок принципу роботи клієнт-серверної архітектури

2.2 Опис структури даних та моделі бази даних

Вагома частина усіх веб-застосунків зберігають усі дані та потрібну інформацію у базі даних. Хоч веб-застосунки, що проектуються на дворівневій архітектурі, коли він складається лише з клієнта і сервера, існують, проте у

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

певних випадках, коли з сервером будуть неполадки, може статися так, що усі дані будуть втрачені. Саме з цієї причини використовуються бази даних. Такий тип архітектури є трирівневим. Бази даних надають можливість зберігати дані у безпеці. Саме тому важливим етапом є проектування логічної моделі бази даних.

Для того, щоб правильно спроектувати базу даних потрібно виконати певні етапи:

- Провести збір та аналіз даних;
- З'ясувати, які сутності будуть у БД;
- З'ясувати, які атрибути будуть у кожній сутності;
- З'ясувати, які обмеження будуть застосовані для сутностей;
- Провести нормалізацію БД;

Проаналізувавши усі зібрані дані можна визначити сутності БД:

- Users (міститиме усю інформацію про користувача);
- Products (міститиме усю інформацію про товар);
- Orders (міститиме усю інформацію про замовлення);
- Categories (міститиме усю інформацію про категорії);
- Roles (міститиме усю інформацію про ролі);
- OrderDetails (допоміжна сутність, що міститиме додаткову інформацію про замовлення);
- UserRoles (допоміжна сутність, що міститиме додаткову інформацію про роль користувача);

Далі варто провести опис атрибутів сутностей:

Users:

- ID – ідентифікатор користувача, унікальний номер (первинний ключ);
- FirstNme – ім'я користувача;
- LastName – прізвище користувача;
- EmailAdress – електронна скринька користувача;
- Password – пароль користувача;

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

Category:

- ID - ідентифікатор категорії;
- Name - назва категорії;
- Sorting - значення, що відповідає за нумерацію категорії в списку;

Products:

- ID – ідентифікатор товару;
- Name – назва товару;
- Description – опис товару;
- Price – ціна товару;
- CategoryId – ідентифікатор категорії товару;
- Image – зображення товару;

OrdersDetails:

- Id - унікальний ідентифікатор замовлення;
- OrderId - номер замовлення;
- UserID – унікальний ідентифікатор користувача;
- ProductId - унікальний ідентифікатор продукту;
- Quantity - кількість продукції;

Orders:

- OrderId – ідентифікатор замовлення;
- UserId – ідентифікатор користувача;
- CreatedAt – час створення замовлення;

Roles:

- Id – ідентифікатор ролі;
- Name – назва ролі;

UserRoles:

- UserId – ідентифікатор користувача;
- RoleId – ідентифікатор ролі;

Для деяких атрибутів застосовувалися певні обмеження, такі як первинний ключ (primary key) та зовнішній ключ (foreign key). Primary Key – (він же – первинний ключ) – надає посилання для зв'язку з іншими таблицями та створює

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		25

обмеження унікальності для стовпця, на якому задається. Foreign Key потрібен для того, щоб надати певне посилання для зв'язку між двома таблицями. \

Після визначення та опису сутностей та атрибутів іде етап нормалізації бази даних. Нормалізація схеми бази даних — покроковий процес розбиття одного відношення (на практиці: таблиці) відповідно до алгоритму нормалізації на декілька відношень на базі функціональних залежностей. Важливим етапом також є визначення зв'язків. Всього є три типи зв'язків:

- один до одного - коли кожен елемент у кожній таблиці відображається лише один раз;
- один до багатьох - коли один елемент в одній таблиці може мати зв'язок із кількома елементами в іншій таблиці;
- багато до багатьох - якщо один або кілька елементів в одній таблиці можуть мати зв'язок з одним або ким елементом в іншій таблиці;

Після з'ясування зв'язків між таблицями та приведення бази даних до третьої нормальної форми було створено ER-діаграму бази даних.

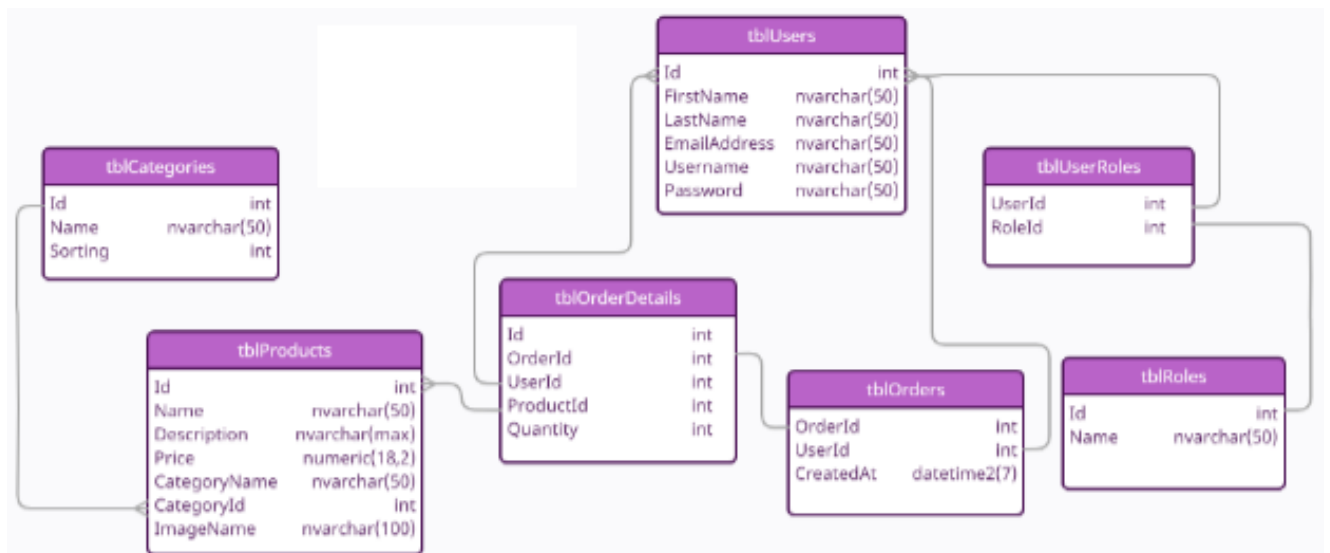


Рисунок 2.2.1 – ER-діаграма бази даних додатку

2.3 Проектування серверної частини веб-застосунку

Проектування серверної частини веб-застосунку з клієнт-серверною архітектурою включає розробку та налаштування серверних компонентів, які забезпечують обробку запитів від клієнтської частини та надання необхідної функціональності.

Основні кроки при проектуванні серверної частини веб-застосунку з клієнт-серверною архітектурою:

- визначення загальної архітектури серверної частини, аналіз та вибір шаблону;
- визначення API для взаємодії між клієнтською та серверною частиною;
- організація безпеки серверної частини;

При проектуванні архітектури серверної частини веб-застосунку з клієнт-серверною архітектурою важливо визначити, як будуть взаємодіяти різні компоненти додатка.

Один із поширених підходів - використання шаблону MVC (Model-View-Controller) дозволяє розділити логіку додатка на окремі компоненти:

- модель представляє бізнес-логіку додатку та дані, з якими вона працює. Вона може включати класи, що відповідають за доступ до бази даних, обробку даних та виконання бізнес-логіки. Модель не залежить від реалізації користувацького інтерфейсу і може використовуватись в різних частинах додатку;
- представлення відповідає за відображення даних користувачу і обробку його взаємодії з додатком. Воно представляє інформацію з моделі у вигляді, зрозумілому користувачу, наприклад, HTML-сторінки;
- контролер взаємодіє з користувачем та координує роботу моделі та представлення. Він приймає вхідні дані від користувача, обробляє їх та виконує необхідні дії, включаючи взаємодію з моделлю для отримання або оновлення даних та передачі результатів в представлення для відображення користувачу;

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

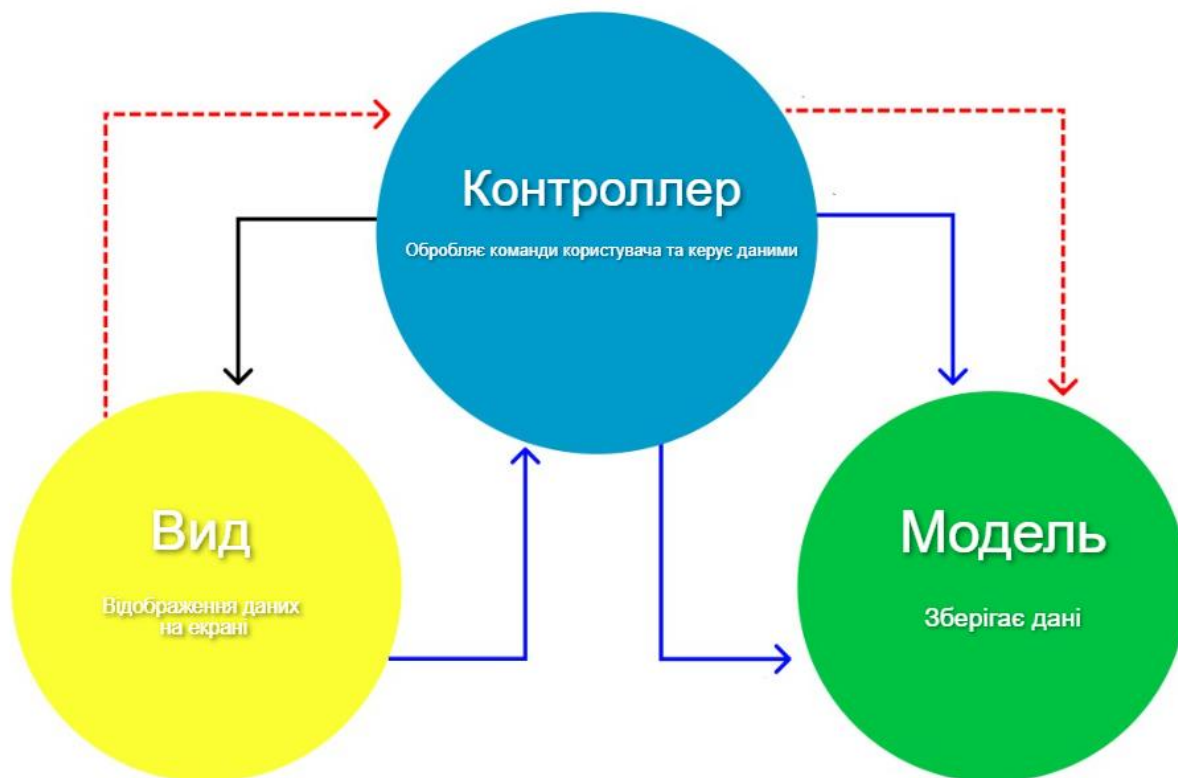


Рисунок 2.3.1 – схема роботи шаблону MVC

Використання шаблону MVC дозволяє забезпечити чистоту коду, та робить застосунок більш гнучким та масштабованим.

Однією з переваг шаблону MVC – це гнучкість та масштабованість, адже застосунок може легко розширюватися, у ньому може з'явитися новий функціонал і при цьому це все відбувається без значних змін у коді та структурі додатку. Не менш важливим є також і те, що кожен компонент виконує свою функцію, що означає те, що, наприклад, представлення може бути заміненим або модифікованим без впливу на модель чи контролер або, наприклад, те, що одна модель може використовуватися в різних частинах коду. В цілому, шаблон MVC є досить потужним інструментом для розробки веб-застосунків.

Наступний етап – це визначення API, що взаємодіють між клієнтською та серверною частиною додатку. Один з таких є REST API, що є архітектурним стилем, який дозволяє взаємодіяти з сервером шляхом передачі та отримання даних за допомогою HTTP-протоколу. REST API використовує розділення між

клієнтом (користувачем або клієнтською програмою) та сервером (системою, що надає послуги). Це дозволяє незалежно масштабувати та розвивати обидва компоненти. Він також використовує різні HTTP-методи для виконання різних операцій над ресурсами.

Найпоширеніші методи включають GET (отримання даних), POST (створення нового ресурсу), PUT (оновлення існуючого ресурсу) та DELETE (видалення ресурсу). REST API забезпечує гнучку та розширювану архітектуру для веб-застосунків, дозволяючи клієнтам ефективно взаємодіяти з серверними ресурсами через стандартні HTTP-протоколи.

Також, на етапі проектування архітектури серверної частини потрібно надати уваги реалізації її безпеки. Одні з основних аспектів реалізації безпеки – це елементарні авторизація та аутентифікація. Аутентифікація може включати в себе використання паролів, токенів, двофакторної аутентифікації, тощо. Для авторизації ж можна використовувати таку модель, в якій різним користувачам надаються різні ролі, що відповідно визначає права доступу до даних.

Для того, щоб схематично зобразити роботу серверної частини додатку можна використати різні діаграми. Діаграму послідовностей можна використати для представлення процесу обробки запитів, взаємодії клієнта та сервера, а також викликів між компонентами. Схематичний рисунок можна використовувати для представлення серверних компонентів, таких як база даних, бізнес-логіка, API та інші.

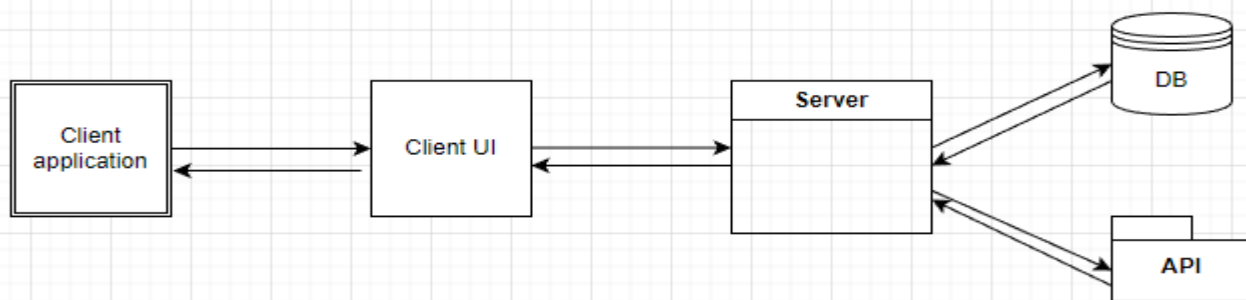


Рисунок 2.3.2 – схематичний рисунок для представлення серверних компонентів

На рисунку зображена діаграма, яка відображає зв'язок різних компонентів додатку.

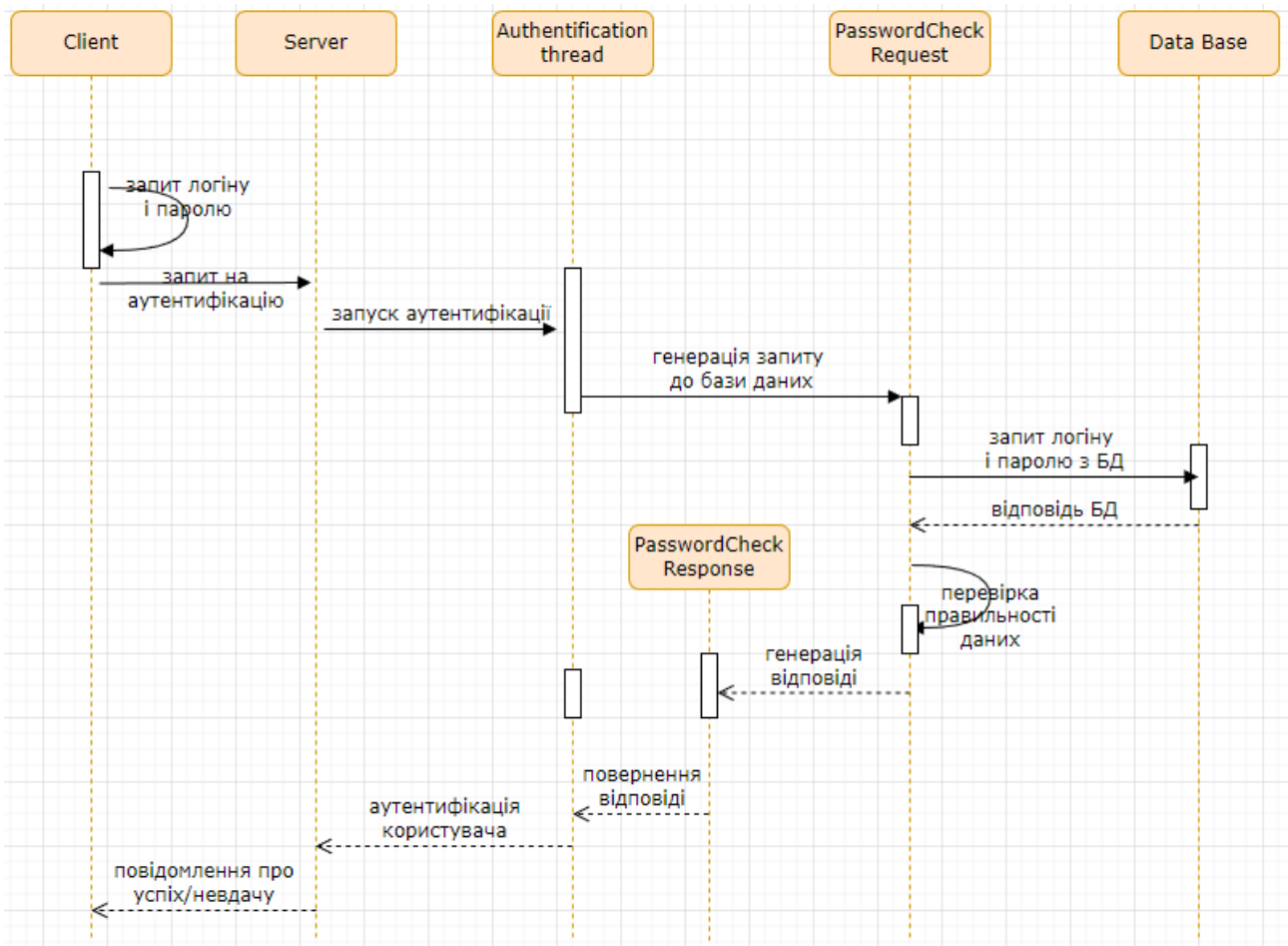


Рисунок 2.3.3 – діаграма послідовностей

На рисунку зображена діаграма послідовностей, яка відображає роботу додатку при спробі аутентифікації користувачем.

2.4 Проектування клієнтської частини веб-застосунку

Проектування клієнтської частини веб- застосунку є важливою складовою розробки веб- застосунків. Цей процес включає в себе створення користувацького інтерфейсу (UI) та розробку клієнтської логіки, які забезпечують взаємодію користувача з застосунком через веб-браузер. У процесі проектування клієнтської частини веб- застосунку важливо брати до уваги

потреби користувачів, забезпечувати зручний інтерфейс та ефективну роботу застосунку. Дотримання сучасних стандартів та найкращих практик веб-розробки допомагає створити високоякісний інтерфейс.

Одним із основних етапів проектування клієнтської частини веб-застосунку – це створення прототипу інтерфейсу, на якому відображаються головний функціонал та розміщення елементів. Для того, щоб зробити правильний прототип слід проаналізувати, з яких частин складається сторінка веб-застосунку. Зазвичай основні частини це: «header», «content» та «footer».

«Header» – частина сторінки, яка відображається у верхній частині сторінки, зазвичай містить у собі логотип фірми або назву сайту, кнопка переходу на головну сторінку, кнопки, які посилають користувача до реєстрації/авторизації, а також кошик товарів покупця.

«Footer» – частина сторінки, яка навпаки від header, знаходиться знизу сторінки і відображає у собі різні контактні дані, телефони, електронні скриньки та посилання на інші ресурси.

«Content» – основна частина сторінки, на якій відображається уся основна інформація. Більша частка наповнення сторінки відноситься саме до цієї частини. Тут зазвичай відображається уся інформація про товари чи послуги, що надаються на сайті. Також у цьому контейнері часто знаходяться додаткові блоки: «navigation» та «side». Блок «navigation» відповідає за навігацію по сайту, у ньому містяться посилання на різні сторінки. Блок «side» може застосовуватися про різному, але в основних випадках це або блок для фільтрів, або блок для додаткової навігації.

На рисунку 2.4.1 зображений макет головної сторінки додатку. Такий вигляд він має для звичайного користувача.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		31

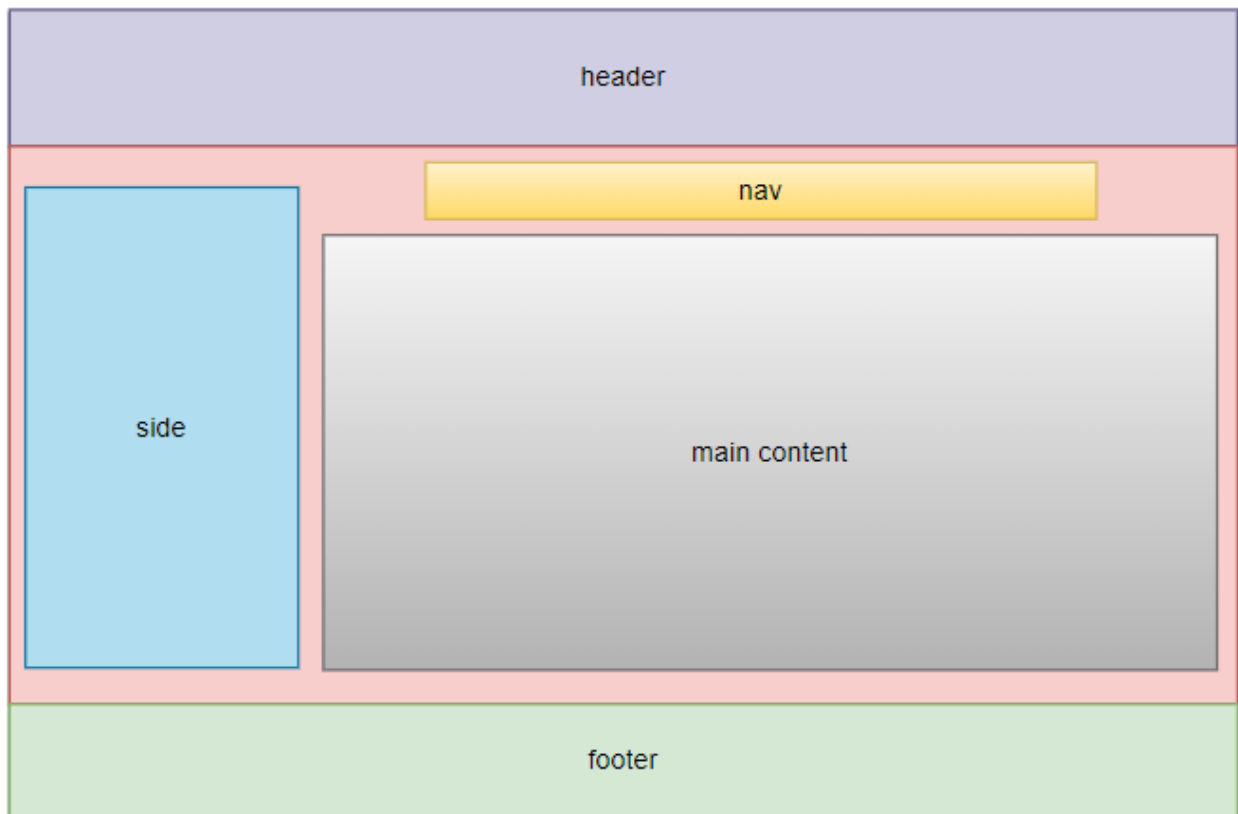


Рисунок 2.4.1 – макет головної сторінки сайту

Для адміністратора головна сторінка буде дещо відрізнятися. Так як у адміністратора є функціонал, для керування сайтом, який не доступний звичайному користувачеві, на макеті видно, що у адміністратора є різні контейнери розділені між собою, які відповідають за різний функціонал. Окрім відображення наповнення сайту, в адміністратора з'являється інша інформація на екрані(кнопки для редагування сайту, кнопки переходу до вкладок адміністрування та перегляду інформації про замовлення)

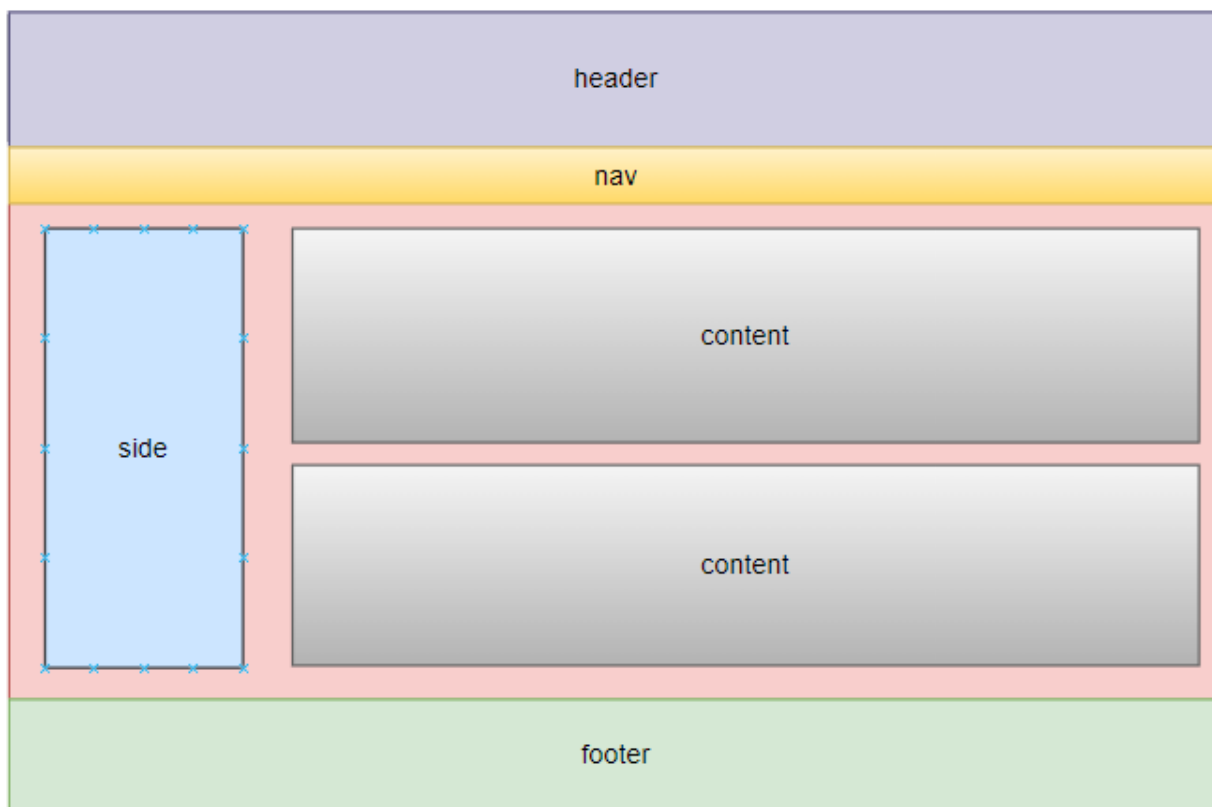


Рисунок 2.4.2 – макет головної сторінки для адміністратора

Залежно від прототипу, розробляється дизайн інтерфейсу. Це включає в себе вибір кольорової схеми, шрифтів, графічних елементів та інших важливих аспектів, що впливають на зовнішній вигляд додатку. Зробивши аналіз актуальних рішень та вимог до розроблюваного додатку була підібрана кольорова палітра(рисунок 2.4.3) та шрифт(рисунок 2.4.4). Кольори були підібрані у строгому стилі, тому чорний і білий є досить гарним поєднанням.

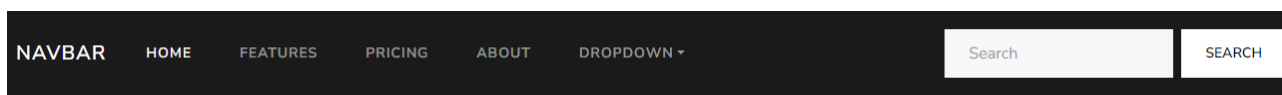


Рисунок 2.4.3 – вибір кольорів для дизайну сайту

Шрифт, який підбирався для сайту був також досить простим, але при цьому його можна легко читати, він не має лишніх елементів і підходить до загального стилю сайту.

TYPOGRAPHY

HEADING 1

HEADING 2

HEADING 3

HEADING 4

HEADING 5

HEADING 6

HEADING WITH MUTED TEXT

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.

Рисунок 2.4.4 – шрифт сайту

2.5 Аналіз та вибір технологій і методів реалізації веб- застосунку

На сьогоднішній день розробка веб-застосунків займає одну з основних ніш в програмуванні в цілому. Існує велика кількість мов програмування, фреймворків та інших технологій, що застосовуються в веб-розробці і з кожним роком кількість різних технологій зростає. На зараз основними мовами програмування, що використовують для веб розробки являється С#, PHP та JavaScript. Якщо звернутися до статистики за минулий 2022 рік, усі ці мови програмування входять у топ-5 мов програмування, що застосовуються у всіх видах розробки.



Рисунок 2.5.1 – статистка використання мов програмування

Усі ці мови програмування мають свої переваги та недоліки при застосуванні і обираються для розробки у відповідності до вимог до веб-застосунку.

Мова програмування JavaScript є однією з найбільш популярних мов програмування, що застосовуються для веб розробки. JS є сильним інструментом

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		34

веб-розробників, оскільки він дозволяє створювати інтерактивні та динамічні веб-сторінки, взаємодіяти з сервером та інтегруватись з різноманітними сервісами. Його широкий набір функціональностей робить його важливим інструментом в сучасній веб-розробці. JS використовується в розробці різних веб-фреймворків, таких як React, Angular і Vue.js. Ці фреймворки дозволяють розробникам будувати складні веб-застосунки, використовуючи компонентний підхід та інші функціональності JS. JS має свої переваги та недоліки:

До переваг можна віднести:

- клієнтська взаємодія(дозволяє створювати динамічні та інтерактивні веб-сторінки, які реагують на дії користувача);
- широке застосування(дозволяє використовувати одну мову для розробки як клієнтських, так і серверних додатків);
- велика спільнота розробників(багато ресурсів, бібліотек, фреймворків та підтримки, що полегшує розробку та вирішення проблем);
- швидкість виконання на браузері(виконується безпосередньо в браузері, що дозволяє отримувати швидкість виконання);

До недоліків можна віднести:

- сумісність браузерів(різні браузери можуть по-різному інтерпретувати та виконувати JS, що може призводити до проблем з сумісністю);
- обмежені можливості доступу до системних ресурсів(JS, виконуваний у браузері, має обмежені можливості доступу до системних ресурсів комп'ютера, таких як файлова система або операційна система);

Загалом, JavaScript є потужним і важливим інструментом для веб-розробки, але він також має свої обмеження та недоліки. Варто враховувати ці фактори проектуванні та розробці веб-додатків.

Так само як і JS, мова програмування PHP має широке застосування у веб-розробці. PHP є однією з найбільш використовуваних мов програмування для розробки серверної частини веб-застосунків. Також PHP надає можливість взаємодіяти з різними системами управління баз даних, а також забезпечує зручний API для підключення до бази даних.

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

Переваги PHP:

- простота застосування(синтаксис PHP легкий для сприйняття і дозволяє швидко створювати функціональні веб-додатки);
- велика кількість фреймворків та бібліотек(ці інструменти спрощують розробку, забезпечують структуру та підтримують повторне використання коду);
- взаємодія з базами даних(має вбудовану підтримку для багатьох систем управління базами даних);

Недоліки PHP:

- швидкодія(у порівнянні з деякими іншими мовами програмування PHP може бути менш продуктивним в плані швидкодії);
- управління типами даних(PHP є слабкотипізованою мовою, що може викликати проблеми при роботі з типами даних);

PHP залишається популярним вибором для багатьох веб-розробників завдяки своїй простоті, широкій спільноті та доступності.

Мова програмування C# широко використовується в веб-розробці з використанням фреймворків і технологій, які побудовані на її основі. C# надає змогу використовувати багатофункціональну платформу .NET для розробки веб-застосунків. Велика кількість програм та сервісів було написано за допомогою технологій C#, одні з них це відомий серед розробників сайт stackoverflow.com або, наприклад, сайт компанії Microsoft. Одним із плюсів C# також являється її строга типізація та потужність, це допомагає забезпечити високий рівень стабільності та надійності веб-застосунків та допомагає виявити помилки під час компіляції. C# використовується для розробки різних типів додатків, тому це суттєво полегшує переносимість коду. Найважливішою перевагою C# можна назвати велику кількість фреймворків, бібліотек та інструментів, що полегшують розробку веб-застосунку. Один з них - ASP.NET, який надає багато готових компонентів та рішень для розробки веб-застосунків. Однією з основних віх цього фреймворку є ASP.NET MVC. ASP.NET MVC - це популярний фреймворк для розробки веб-застосунків, який базується на патерні проектування MVC.

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

До основних переваг ASP.NET MVC можна віднести:

- розділення логіки додатку на модель, представлення та контролер, що полегшує розподіл обов'язків між різними компонентами;
- можливість тестування кожного компонента окремо, що сприяє покращенню якості коду;
- підтримка шаблонів і розширюваності для створення ефективних та гнучких додатків;
- широкий набір вбудованих можливостей, таких як маршрутизація, обробка помилок, аутентифікація та авторизація, що дозволяють прискорити процес розробки;

Підводячи підсумки до написаного вище, можна сказати, що вибір було зроблено саме на користь використання мови програмування C# разом із різними фреймворками для платформи .NET, а саме ASP.NET MVC та Entity Framework.

Наступним етапом аналізу та вибору технологій розробки є аналіз та вибір типу бази даних та системи керування базою даних. Основні два типи баз даних – реляційні та нереляційні бази даних.

Реляційна база даних є одним з найпоширеніших типів баз даних. Вона зберігає дані у вигляді таблиць зі зв'язками між ними. Реляційні БД, такі як MySQL, PostgreSQL або Microsoft SQL Server, забезпечують стандартизовану мову запитів (SQL) та гарантують цілісність даних.

Нереляційна база даних (NoSQL): Нереляційні бази даних, такі як MongoDB або Cassandra, використовують інші моделі для зберігання та організації даних. Вони часто застосовуються, коли необхідно обробляти великі обсяги неструктурованих даних або коли вимоги до швидкодії є критичними.

Таблиця 2.5.1 – порівняння типів баз даних

Характеристика	Реляційна БД	Нереляційні БД
Структура даних	Таблиці із зв'язками	Різні моделі даних

Продовження таблиці 2.5.1

Масштабованість	Горизонтальна та вертикальна	Горизонтальна масштабованість
Запити	SQL-запити	Різні механізми запитів
Гнучкість	Статична схема даних	Динамічна схема даних
Загальне використання	Універсальні, широко використовуються в різних сценаріях	Використовуються в основному для великих обсягів даних

Як вже було написано вище, мова програмування С# та фреймворк ASP.Net MVC сумісні також з фреймворком Entity Framework. Entity Framework (EF) є фреймворком для розробки програмного забезпечення, який надає об'єктно-орієнтований підхід до роботи з базами даних у середовищі .NET. Він є ORM (Object-Relational Mapping), що дозволяє розробникам працювати з базами даних, використовуючи об'єктну модель даних. Entity Framework володіє низкою переваг, такі як:

- спрощений доступ до бази даних (EF дозволяє розробникам працювати з базою даних, використовуючи об'єктну модель, що полегшує взаємодію з даними та запитамі);
- автоматичне створення SQL-запитів (EF автоматично генерує SQL-запити на основі запитів LINQ що зменшує необхідність вручну писати складні SQL-запити);
- підтримка різних баз даних;

Підбиваючи підсумки написаного вище, можна зробити висновки, що застосування Entity Framework надає велику кількість переваг, тому його застосування значно полегшить роботу з базою даних. База даних, яка найбільше підходить відносно вимог, які були описані раніше і при цьому має чудову сумісність з Entity Framework є Microsoft SQL Server. Саме на користь Microsoft SQL Server був зроблений вибір.

Також не менш важливими технологіями, що застосовуються для розробки веб-застосунку є мова розмітки HTML та каскадні таблиці стилів CSS.

HTML використовується для створення структури веб-сторінки. Він включає різноманітні теги, які визначають різні елементи на сторінці, такі як заголовки, абзаци, списки, таблиці, форми та інші. З допомогою HTML розробники створюють структуровану основу веб-сторінок.

CSS використовується для задання зовнішнього вигляду і стилю веб-сторінки. Він дозволяє змінювати кольори, шрифти, розміри, межі, фони та інші властивості елементів HTML. CSS забезпечує можливість створювати адаптивний дизайн, який змінюється в залежності від розміру екрану пристрою.

2.6. Висновки до розділу 2

Отже, виконуючи роботу над розділом 2 було зроблено аналіз та вибір архітектури веб-застосунку, розглянуто схему БД, проведено нормалізацію БД та візуалізовано її у вигляді ER-діаграми. Наступним кроком спроектовано серверну частину веб-застосунку та клієнтську частину веб-застосунку.

Побудовано різні діаграми та схеми для демонстрації роботи серверної та клієнтської частини веб-застосунку. Заключним етапом стало проведення аналізу та вибір методів та технологій, за допомогою яких відбудуватиметься реалізація веб-застосунку.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка бази даних

У попередніх розділах було визначено усі сутності та атрибути, спроектовано логічну модель та обрано систему керування базою даних. Наступним кроком після цього відбувається створення фізичної бази даних. На основі логічної моделі створюються таблиці та відношення між ними. Важливим етапом також є встановлення обмежень, такі як: первинні ключі, зовнішні ключі та інші обмеження (тип та розмір даних). У таблицях 3.1.1-3.1.6 відображаються тип та розмір даних, а також обмеження для кожної таблиці фізичної бази даних.

Таблиця 3.1.1 – таблиця Users

Поле	Тип та розмір даних	Обмеження
UserID	int	primary key, unique key, not null
FirstName	nvarchar(50)	not null
LastName	nvarchar(50)	not null
EmailAddress	nvarchar(50)	not null
Username	nvarchar(50)	not null
Password	nvarchar(50)	not null

Таблиця Users використовується для зберігання інформації про зареєстрованих користувачів. У ній немає ніяких зовнішніх ключів, лише один первинний ключ, який зберігатиме унікальне значення для кожного користувача.

Таблиця 3.1.2 – таблиця Categories

Поле	Тип та розмір даних	Обмеження
CategoryID	int	primary key, unique key, not null
Name	nvarchar(50)	not null
Sorting	int	not null

Таблиця Categories містить у собі дані про категорії товарів. Маленька таблиця з одним первинним ключем.

Таблиця 3.1.3 – таблиця Products

Поле	Тип та розмір даних	Обмеження
ProductID	int	primary key, unique key, not null
Name	nvarchar(50)	not null
Description	nvarchar(max)	not null
Price	numeric(18,2)	not null
CategoryName	nvarchar(50)	not null
CategoryId	int	foreign key, not null
ProductImage	nvarbinary(max)	not null

Таблиця Products зберігає дані про товари. У таблиці один первинний ключ, а також один зовнішній ключ, який зв'язує цю таблицю з таблицею categories.

Таблиця 3.1.4 – таблиця OrdersDetails

Поле	Тип та розмір даних	Обмеження
ODID	int	primary key, unique key, not null
OrderID	int	foreign key, not null
UserID	int	foreign key, not null
ProductID	int	foreign key, not null
Quantity	int	not null

Таблиця OrdersDetails містить у собі деталі про кожне замовлення, за допомогою зовнішніх ключів зв'язується з таблицями orders, users та products.

Таблиця 3.1.5 – таблиця Orders

Поле	Тип та розмір даних	Обмеження
OrderID	Int	foreign key, not null
UserID	Int	foreign key, not null
CreatedAt	datetime	not null

У таблиці orders зберігається загальна інформація про замовлення. Має два обмеження у вигляді зовнішніх ключів.

Таблиця 3.1.6 – таблиця UsersRoles

Поле	Тип та розмір даних	Обмеження
UserID	int	foreign key, not null
RoleID	int	foreign key, not null

Таблиця UsersRoles допоміжна таблиця для прив'язки ролі до користувача, тому не має власного первинного ключа, а лише два зовнішніх ключі.

Таблиця 3.1.7 – таблиця Roles

Поле	Тип та розмір даних	Обмеження
UserID	int	primary key, unique key, not null
Name	nvarchar(50)	not null

У таблиці Roles зберігається загальна інформація про ролі.

Створена фізична база даних має структуру, яка відображається на рисунку 3.1.1., на якому видно, які таблиці містяться у базі даних.

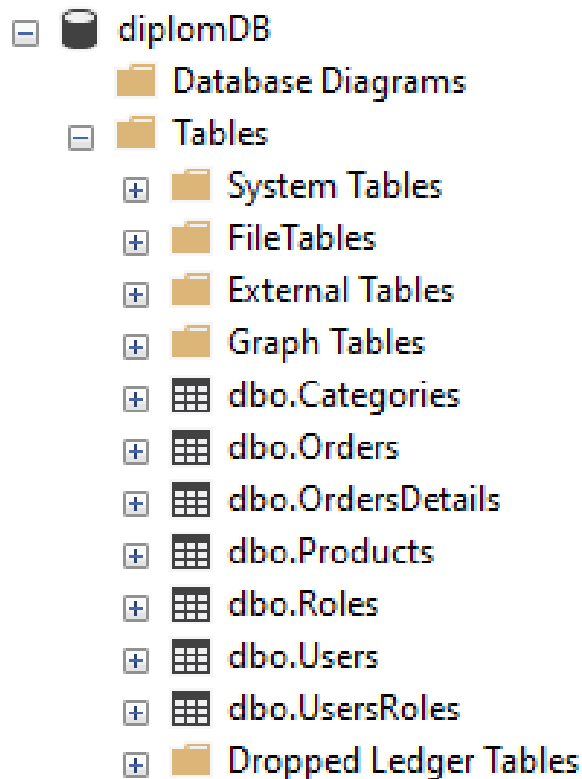


Рисунок 3.1.1 – структура бази даних

3.2 Розробка програмних модулів

У цьому розділі описуються розроблені програмні модулі, демонструється результати їх роботи, а також демонструються уривки коду програми, що відповідаються кожному модулю програми.

Для початку слід написати про форму реєстрації та форму авторизації. Так як після реєстрації або авторизації користувач отримує доступ до більшої частини функціоналу сайту, справедливо буде назвати ці програмні модулі, як одні з основних. На рисунку 3.2.1. зображено код форми для реєстрації користувача.

```

public InputModel Input { get; set; }

3 references
public string returnUrl { get; set; }

3 references
public IList<AuthenticationScheme> ExternalLogins { get; set; }

1 reference
public class InputModel
{
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]
    6 references
    public string Email { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "The {0} must be at least {2} and at max {1} characters long.", MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    3 references
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
    2 references
    public string ConfirmPassword { get; set; }
    [Required]
    3 references
    public string FullName { get; set; }
    [Required]
    3 references
    public string PhoneNumber { get; set; }
}

```

Рисунок 3.2.1 – уривок коду модулю реєстрації

Як вже було написано вище, авторизація відкриває користувачу доступ до більшої частини функціоналу сайту, тому цей модуль також важливо правильно реалізувати. На рисунку 3.2.3. можна побачити уривок коду, який відповідає за авторизацію користувача і перевірку введення даних.

```

43
44
45
46
47
48
49
50
51
52
53
54
55
public class InputModel
{
    [Required]
    [EmailAddress]
    2 references
    public string Email { get; set; }

    [Required]
    [DataType(DataType.Password)]
    2 references
    public string Password { get; set; }

    [Display(Name = "Remember me?")]
    2 references
    public bool RememberMe { get; set; }
}

```

Рисунок 3.2.2 – уривок коду модуля авторизації

Для сайту, який класифікується як інтернет-магазин, розміщення товарів основна річ, тому створення нового запису про товар є одним із найважливіших

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

модулів програми. На рисунку 3.2.3 зображено уривок коду, що відповідає саме за збереження нового товару.

```
97         string upload = webRootPath + WC.ImagePath;
98         string fileName = Guid.NewGuid().ToString();
99         string extension = Path.GetExtension(files[0].FileName);
100
101         using (var fileStream = new FileStream(Path.Combine(upload, fileName + extension), FileMode.Create))
102         {
103             files[0].CopyTo(fileStream);
104         }
105
106         productVM.Product.Image = fileName + extension;
107
108         _prodRepo.Add(productVM.Product);
109     }
```

Рисунок 3.2.3 – уривок коду модуля створення запису

Зміна характеристик, опису або ціни товару відбувається досить часто, тому справедливо буде відмітити модуль, який відповідає за оновлення запису товару. Це уривок коду відображається на рисунку 3.2.4

```
113         var objFromDb = _prodRepo.FirstOrDefault(u => u.Id == productVM.Product.Id, isTracking:false);
114
115         if (files.Count > 0)
116         {
117             string upload = webRootPath + WC.ImagePath;
118             string fileName = Guid.NewGuid().ToString();
119             string extension = Path.GetExtension(files[0].FileName);
120
121             var oldFile = Path.Combine(upload, objFromDb.Image);
122
123             if (System.IO.File.Exists(oldFile))
124             {
125                 System.IO.File.Delete(oldFile);
126             }
127
128             using (var fileStream = new FileStream(Path.Combine(upload, fileName + extension), FileMode.Create))
129             {
130                 files[0].CopyTo(fileStream);
131             }
132
133             productVM.Product.Image = fileName + extension;
134         }
135         else
136         {
137             productVM.Product.Image = objFromDb.Image;
138         }
139         _prodRepo.Update(productVM.Product);
140     }
```

Рисунок 3.2.4 – уривок коду модуля редагування запису

Для завершення покупки клієнт має скористуватися кошиком, у якому він може відредагувати вміст свого замовлення та розпочати його оформлення та покупку. На рисунку 3.2.5 зображено фрагмент коду, що відповідає за кошик клієнта, а саме відображення товарів у кошику та відповідає за інший функціонал кошику(редагування кількості товару або видалення його з кошику)

```

71 public IActionResult DetailsPost(int id, DetailsVM detailsVM)
72 {
73     List<ShoppingCart> shoppingCartList = new List<ShoppingCart>();
74     if(HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart)!=null
75         && HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart).Count() > 0)
76     {
77         shoppingCartList = HttpContext.Session.Get<List<ShoppingCart>>(WC.SessionCart);
78     }
79     shoppingCartList.Add(new ShoppingCart { ProductId = id});
80     HttpContext.Session.Set(WC.SessionCart, shoppingCartList);
81     TempData[WC.Success] = "Item add to cart successfully";
82     return RedirectToAction(nameof(Index));
83 }
84
85 D references
86 public IActionResult RemoveFromCart(int id)
87 {
88     List<ShoppingCart> shoppingCartList = new List<ShoppingCart>();
89     if (HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart) != null
90         && HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart).Count() > 0)
91     {
92         shoppingCartList = HttpContext.Session.Get<List<ShoppingCart>>(WC.SessionCart);
93     }
94     var itemToRemove = shoppingCartList.SingleOrDefault(r => r.ProductId == id);
95     if (itemToRemove != null)
96     {
97         shoppingCartList.Remove(itemToRemove);
98     }
99
100     HttpContext.Session.Set(WC.SessionCart, shoppingCartList);
101     TempData[WC.Success] = "Item removed from cart successfully";
102     return RedirectToAction(nameof(Index));
103 }

```

Рисунок 3.2.5 – уривок коду модуля кошику

Одним із основних модулів програми, який відноситься до адміністративної частини – це стеження за замовленнями. На рисунку 3.2.6 зображено фрагмент коду, що відповідає за меню відстежування замовлення. За допомогою цього модуля адміністратор зможе керувати замовленнями, слідкувати за їх статусом та відмічати успішно доставлені замовлення.

```

68 public IActionResult Details(int id)
69 {
70     OrderVM = new OrderVM()
71     {
72         OrderHeader = _orderHRepo.FirstOrDefault(u => u.Id == id),
73         OrderDetail = _orderDRepo.GetAll(o => o.OrderHeaderId == id, includeProperties: "Product")
74     };
75
76     return View(OrderVM);
77 }
78
79 [HttpPost]
80 D references
81 public IActionResult StartProcessing()
82 {
83     OrderHeader orderHeader = _orderHRepo.FirstOrDefault(u => u.Id == OrderVM.OrderHeader.Id);
84     orderHeader.OrderStatus = WC.StatusInProgress;
85     _orderHRepo.Save();
86     TempData[WC.Success] = "Order is In Process";
87     return RedirectToAction(nameof(Index));
88 }
89
90 [HttpPost]
91 D references
92 public IActionResult ShipOrder()
93 {
94     OrderHeader orderHeader = _orderHRepo.FirstOrDefault(u => u.Id == OrderVM.OrderHeader.Id);
95     orderHeader.OrderStatus = WC.StatusShipped;
96     orderHeader.ShippingDate = DateTime.Now;
97     _orderHRepo.Save();
98     TempData[WC.Success] = "Order Shipped Successfully";
99     return RedirectToAction(nameof(Index));
100 }

```

Рисунок 3.2.6 – уривок коду модуля перегляду замовлення

									Арк.
									46
Зм.	Арк	№ докум.	Підпис	Дата					

3.3 Керівництво користувача

У цьому розділі здійснюється покорковий опис користування веб-застосунком для всіх категорій користувачів. Розпочати слід з неавторизованого користувача, тобто людини, яка щойно відкрила сайт у себе в браузері. Перша сторінка (рисунки 3.3.1), яка відкривається перед користувачем – це головна сторінка з товарами.

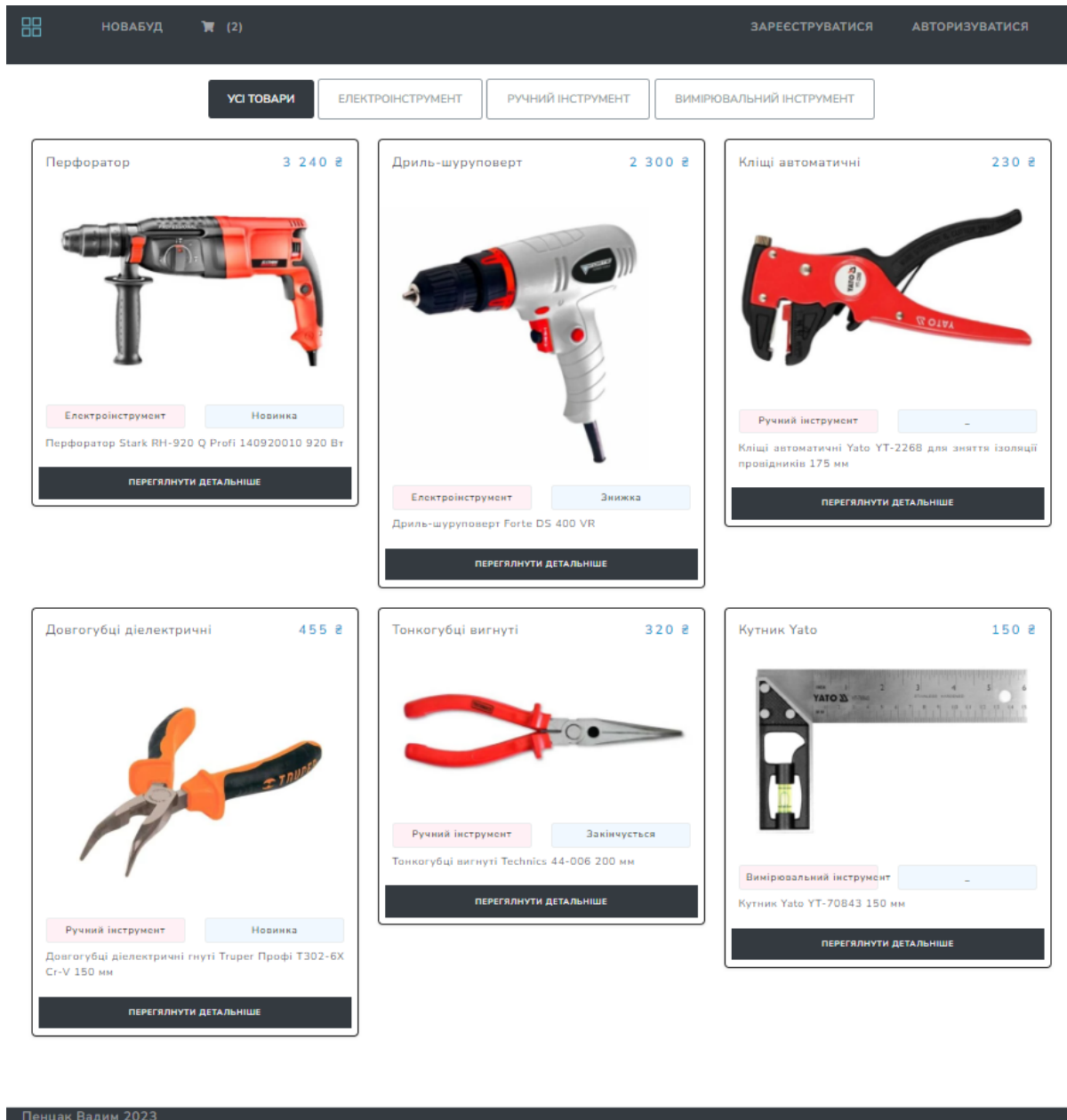


Рисунок 3.3.1 – головний екран

По центру екрана користувач бачить каталог з усіма представленими товарами і може вільно переглядати їх. Вище над товарами розташовується блок фільтрів, яким користувач може скористатися для спрощеного пошуку товарів.

З самого верху екрану розташована «шапка» сайту, де розміщуються кнопки переходу на головну сторінку, кнопки реєстрації та авторизації користувача. Неавторизований користувач також може відкрити кошик, якщо він щось туди додав, але інформація про обраний товар збережеться там тільки у випадку, якщо користувач далі авторизується чи зареєструється, у іншому випадку після закриття сайту весь вміст кошику очиститься.

Для здійснення покупок неавторизований користувач має увійти в систему, тому у випадку, коли він не має свого особистого профілю, він переходить до вікна реєстрації(рисунок 3.3.2). По центру екрана користувач бачить форму реєстрації, де він має ввести інформацію про себе(прізвище та ім'я, електронну скриньку та телефонний номер), а також ввести пароль, за допомогою якого він зможе увійти в систему.

РЕЄСТРАЦІЯ

Логін

Електронна пошта

+380 Номер телефону

Пароль Підтвердження паролю

РЕЄСТРАЦІЯ

2023 КвР

Рисунок 3.3.2 – вікно реєстрації

Для того, щоб уникнути встановлення неправильного паролю на формі розташовується комірка для введення паролю повторно. Коли користувач введе

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		48

всі дані, він може натиснути кнопку «Реєстрація» після чого його особистий профіль створиться, користувач зможе авторизуватися у системі і розпочати покупки.

Якщо користувач вже має свій особистий профіль, він може просто авторизуватися в системі. Для цього він може перейти до форми авторизації, натиснувши кнопку переходу у верхній частині сайту. Форма авторизації виглядає доволі просто, всього дві комірки, де потрібно ввести особисту електронну пошту та пароль, які він вказував при реєстрації. Нижче під комірками вводу даних розташовується кнопка для виконання авторизації. Також на формі розміщена спеціальна кнопка, за допомогою якої користувач може відновити свій пароль, у випадку, якщо він втратив доступ до профілю або забув пароль.

Рисунок 3.3.3 – вікно авторизації

Після авторизації користувача повертає назад до головної сторінки. Те, що користувач здійснив вхід в систему можна зрозуміти по характерному надпису у верхній частину профілю(рисунок). Тепер користувач може користуватися всім функціоналом сайту.

На рисунку 3.3.4 також відображається застосування фільтрів. Як видно з рисунку над усіма товарами розташована секція, де можна відфільтрувати товар. Користувач може як переглядати весь товар, так і вибрати якусь конкретну категорію, для того, щоб спростити собі пошук товару. На рисунку

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

продемонстрована робота фільтру, який показує лише одну обрану категорію товарів, це видно якщо порівняти рисунок 3.3.1 та рисунок 3.3.4

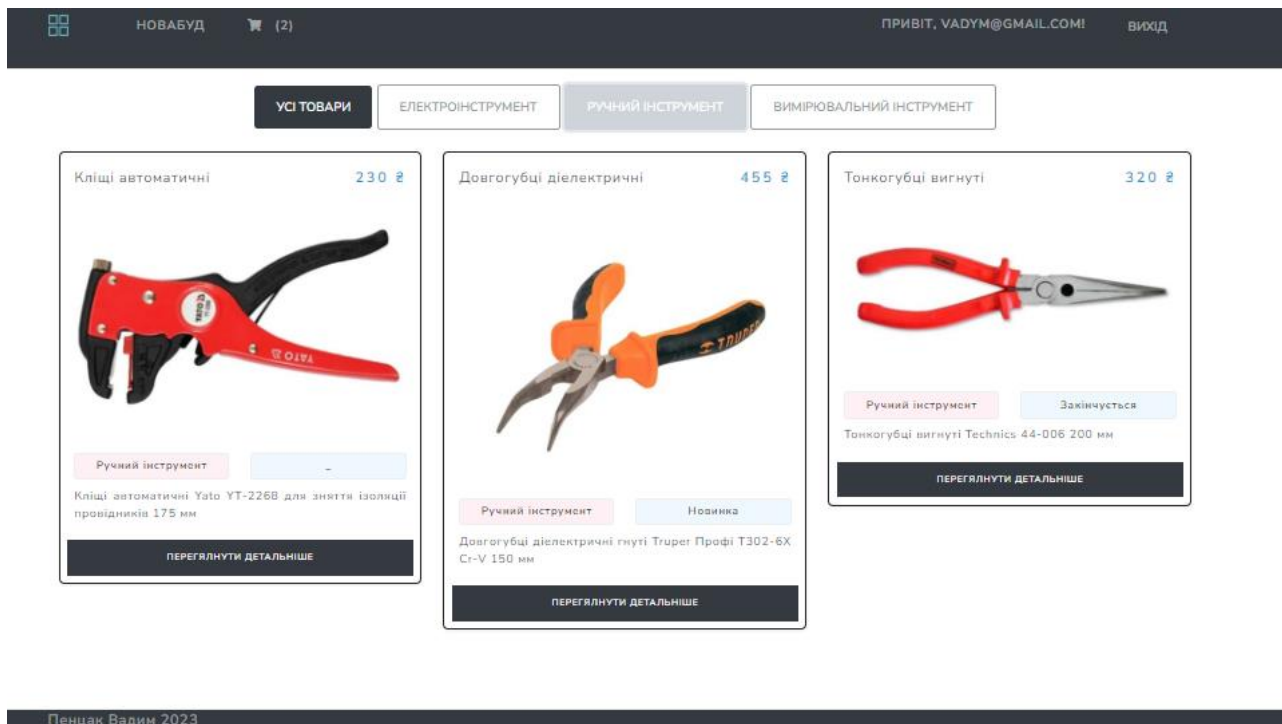


Рисунок 3.3.4 – каталог з застосуванням фільтру

Якщо користувач хоче переглянути детальну інформацію про товар, він може натиснути кнопку «переглянути детальніше» під товаром, який його цікавить, після чого на екрані відкриється вікно з детальною інформацією про товар.

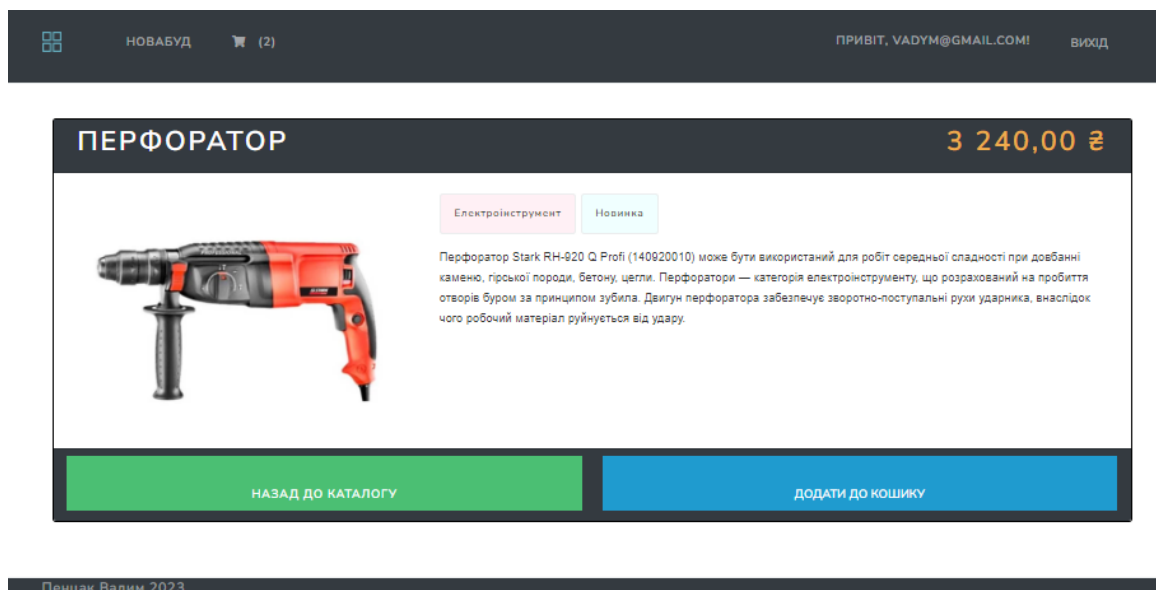


Рисунок 3.3.5 – детальний перегляд товару

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		50

Після того, як товар додано до кошику, вміст кошику можна переглянути. Для цього потрібно натиснути на спеціальну іконку кошика у верхній частині сайту. На рисунку 3.3.6 показано вигляд кошику. Як видно з рисунку, користувач додав до кошику два товари. Далі він може або відредагувати вміст кошику, або перейти до оформлення покупки. Якщо користувач хоче повністю очистити вміст кошику, то він може скористуватися спеціальною кнопкою, що відображається над товарами. Поруч з нею знаходиться кнопка, яка повертає користувача до каталогу.

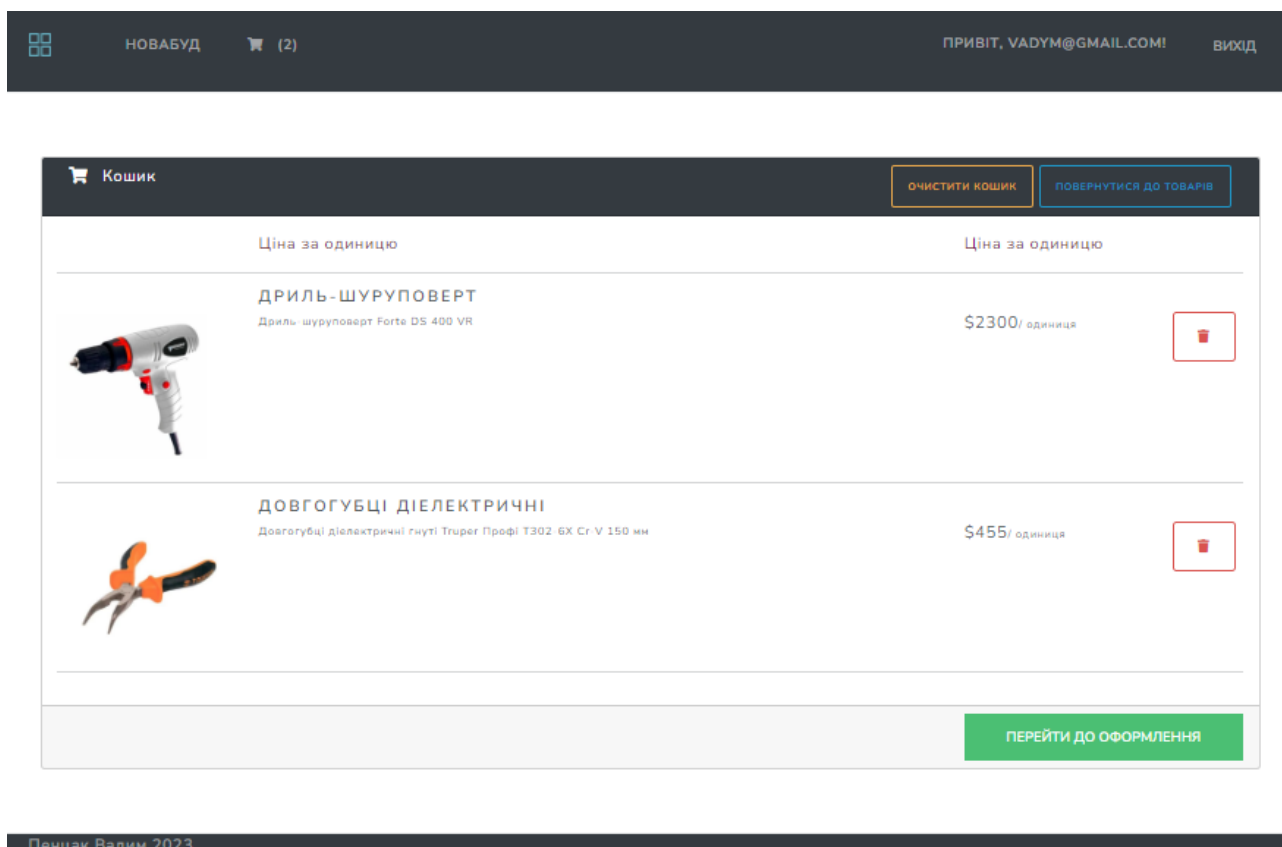


Рисунок 3.3.6 – перегляд каталогу

На рисунку 3.3.7. можна побачити, як виглядають налаштування особистого кабінету користувача. На першій вкладці, яка відкривається на екрані відображається загальна інформація про користувача, його контактні дані. Якщо електронна скринька або номер телефону користувача зміниться, він може відредагувати цю інформацію в профілі, просто ввівши нові дані у відповідні клітинки і натиснувши кнопку «Зберегти», яка розташована внизу під відображеними даними.

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		51

Рисунок 3.3.7 – налаштування профілю

Окрім вкладки з загальними даними користувача у налаштуваннях профілю також є вкладки зміни паролю користувача(рисунок 3.3.8). Тут застосовується такий самий принцип як і при реєстрації, де користувач має ввести пароль двічі, щоб уникнути помилки при введенні, а також він має ввести старий пароль, який застосовується на момент заміни.

Рисунок 3.3.8 – зміна паролю

З загальними принципами користування сайтом для неавторизованого та авторизованого користувача ознайомлено. Наступний крок – керівництво користування для адміністратора. Адміністратор може авторизуватися так само, як і звичайний користувач, змін у цьому не відбувається.

Авторизований адміністратор на головній сторінці бачить приблизно те саме, що і звичайний користувач. Основні відмінності криються у «шапці» сайту. Як можна побачити з рисунку 3.3.9, по характерному напису, що авторизація відбулася саме з профіля адміністратора. Також у тій самій частині сайту видно відмінність між звичайним користувачем та адміністратором, адже тепер там доступні нові функції для керування сайтом та переглядом замовлень.

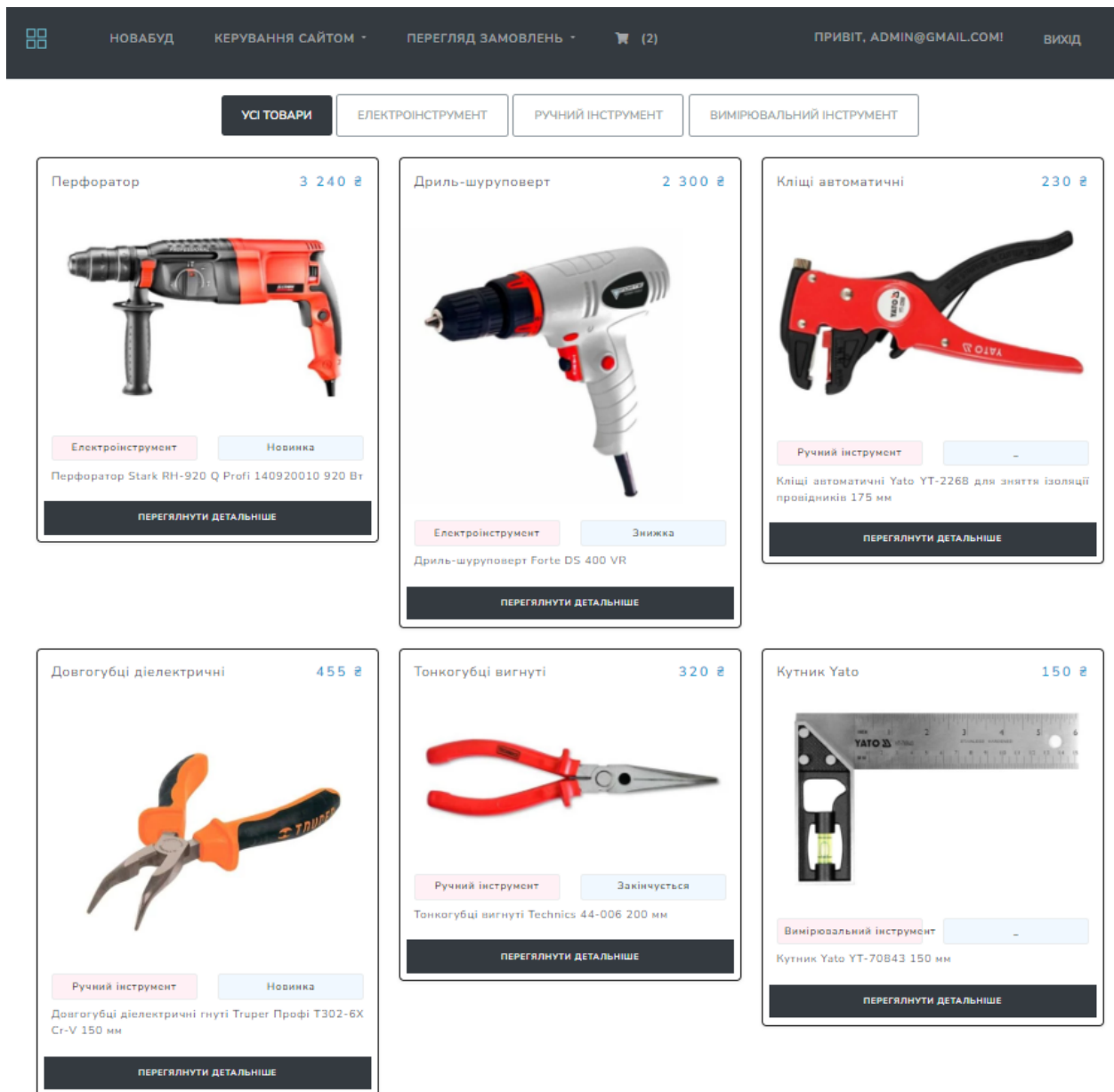


Рисунок 3.3.9. – головний екран від лиця адміністратора

Так як на сайті є можливість створення замовлень адміністратором, у випадку, коли покупець прийшов у фізичний магазин, але потрібний товар був відсутній, то каталог і кошик має такий самий функціонал, як в звичайного

користувача так і в адміністратора. При оформленні адміністратор вводить дані покупця і спрямовує доставку прямо до нього.

Два нових меню, що з'явилися у «шапці» сайту представляють собою випадуючі списки(рисунок 3.3.10 та рисунок 3.3.11). Натискаючи на якусь кнопку із цього списку, який відображається на рисунку 3.3.10, відкривається меню керування категоріями, пропозиціями або товаром. Детальний опис цих функцій буде нижче. Також у цьому списку є спеціальна реєстрація адміністратора, коли треба створити це один особистий профіль, що володітиме усіма функціями адміністратора. Реєстрація такого профілю відбувається так само, як і реєстрація звичайного користувача, єдина відмінність це те, що до форми реєстрації можна перейти лише з спеціального меню адміністратора.

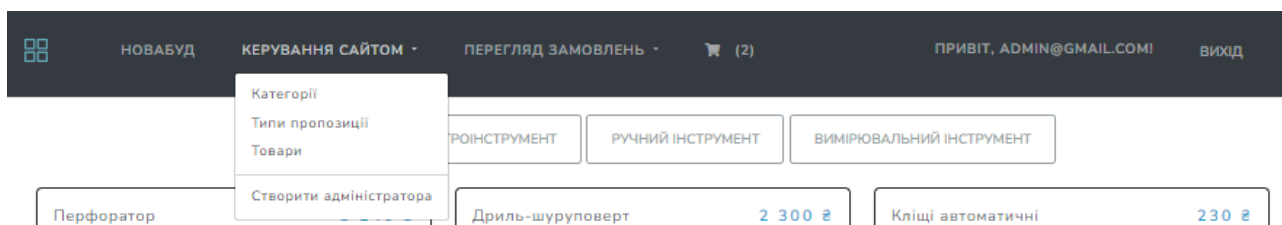


Рисунок 3.3.10 – меню керування сайтом для адміністратора

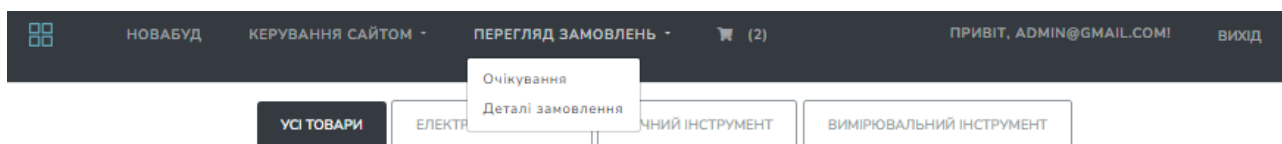








Рисунок 3.3.11 – меню перегляду замовлень для адміністратора

Відкривши меню «Категорії»(рисунок 3.3.12) адміністратор може переглянути їх список. На цій вкладці також присутні по дві кнопки навпроти кожної категорії. Вони відповідають за редагування та видалення категорії. Якщо адміністратор хоче якось змінити назву категорії або змінити її порядок відображення у каталозі він може зробити це за допомогою цього інструменту. Так само, у випадку, коли категорію потрібно взагалі видалити, це робиться за допомогою призначеної для цього кнопки. Також з цього меню можна перейти до меню створення нової категорії. Відповідна кнопка розташована трішки вище над списком уже наявних категорій.

СПИСОК КАТЕГОРІЙТ



[+ СТВОРИТИ НОВУ КАТЕГОРІЮ](#)

НАЗВА КАТЕГОРІЇ	ПОРЯДОК ВІДОБРАЖЕННЯ	
Електроінструмент	1	 
Ручний інструмент	2	 
Вимірвальний інструмент	3	 

Пенцак Вадим 2023

Рисунок 3.3.12 – перегляд списку категорій

Меню створення нової категорії(рисунок 3.3.13) інтуєтивно-зрозуміле і просте. У спеціальні комірки потрібно ввести назву, а також вибрати порядок відображення у списку.


 НОВАБУД КЕРУВАННЯ САЙТОМ ПЕРЕГЛЯД ЗАМОВЛЕНЬ  (2) ПРИВІТ, ADMIN@GMAIL.COM! ВИХІД

ДОДАТИ КАТЕГОРІЮ

Назва категорії

Порядок відображення

Пенцак Вадим 2023

Рисунок 3.3.13 – створення категорії

На сайті присутні товари, що продаються за спеціальною пропозицією(акційні товари, новинки, тощо). Для того, щоб адміністратор міг позначити товари цими відмітками у нього є функціонал, що допомагає створювати такі пропозиції(рисунок 3.3.14). Меню для пропозицій має дуже схожий функціонал до попереднього меню категорій. Тут так само можна

редагувати записи, видаляти або створювати нові. Вікно створення пропозиції ідентичне до меню створення категорії.

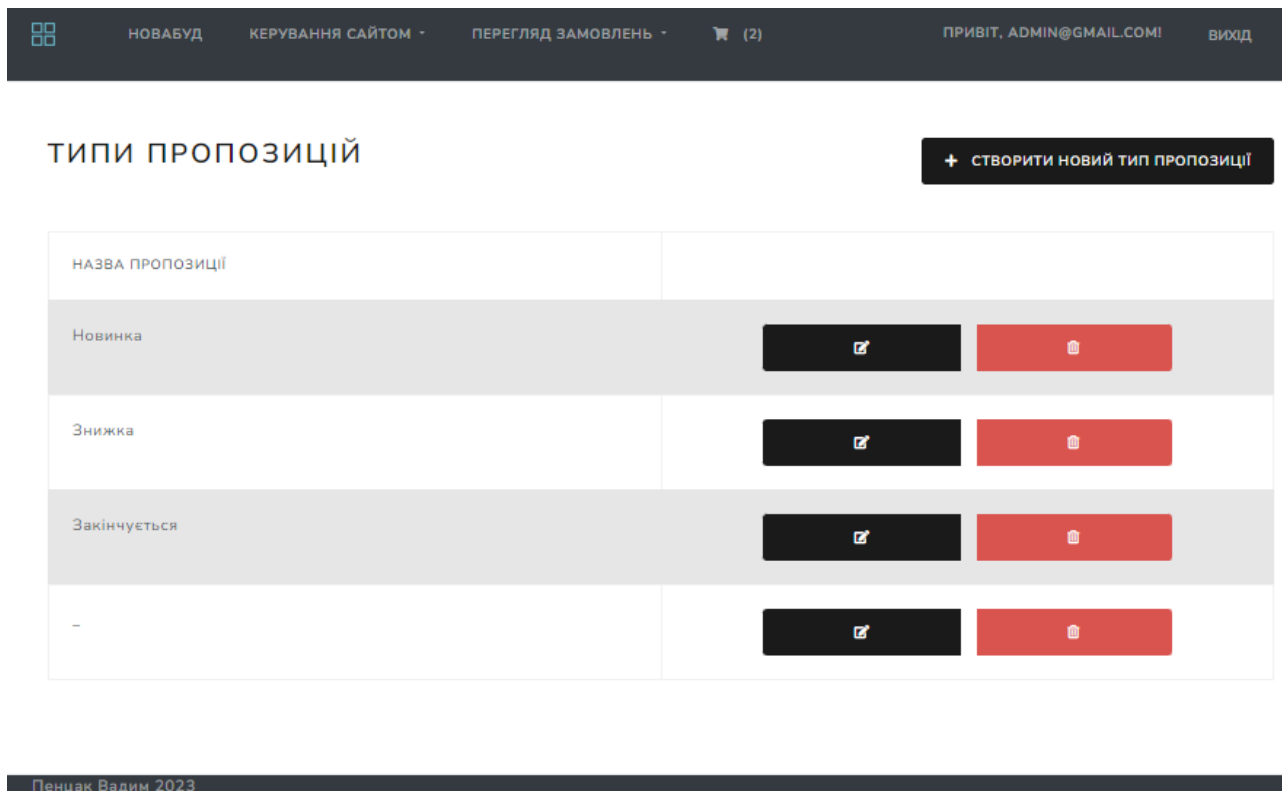


Рисунок 3.3.14 – перегляд пропозицій

Основним меню адміністратора у керуванні сайтом є меню списку товарів(рисунок 3.3.15.). У цьому меню відображаються усі товари, які є на сайті. Для адміністратора відображається уся інформація про товари, також він може відредагувати товари, коли, наприклад, змінюється його ціна або коли товар переходить з акційної пропозиції до звичайної пропозиції або пропозиції закінчення товару. Функціонал цього меню достатньо схожий на попередні меню, тут є ті самі кнопки редагування, видалення та створення нового запису.



СПИСОК ТОВАРІВ

+ СТВОРИТИ НОВИЙ ТОВАР










НАЗВА ТОВАРУ	ЦІНА	КАТЕГОРІЯ	ТИП ПРОПОЗИЦІЇ	
Перфоратор	3240	Електроінструмент	Новинка	 
Дриль-шурупверт	2300	Електроінструмент	Знижка	 
Кліщі автоматичні	230	Ручний інструмент	-	 
Довгогубці діелектричні	455	Ручний інструмент	Новинка	 
Тонкогубці вигнуті	320	Ручний інструмент	Закінчується	 
Кутник Yato	150	Вимірювальний інструмент	-	 

Рисунок 3.3.15 – перегляд списку товарів

Проте меню редагування товару, на відміну від попередніх меню редагування категорій та пропозицій, досить відрізняється. Як видно на рисунку 3.3.16 інформація про товар має дуже багато значень, починаючи від повної назви, закінчуючи ціною. Також для товарів є функція показу зображення, тому так само присутня і функція зміни цього фото. Комірка для вводу детального опису товару має великий обсяг і включає велику кількість різних налаштувань. Адміністратор може скористуватися кнопкою «Редагувати» для збереження змін або кнопкою «Назад», у разі відсутності потреби внесення змін.



РЕДАГУВАННЯ ТОВАРУ

Назва	Перфоратор
Price	3240
Короткий опис	Перфоратор Stark RH-920 Q Profi 140920010 920 Вт
Повний опис	<div style="border: 1px solid #ccc; padding: 5px;"> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;"> 🔍 B U NUNITO SANS A </div> <div style="padding: 5px;"> <p>Перфоратор Stark RH-920 Q Profi (140920010) може бути використаний для робіт середньої складності при добуванні каменю, гірської породи, бетону, цегли. Перфоратори — категорія електрострументу, що розрахований на пробиття отворів буром за принципом зубила. Двигун перфоратора забезпечує зворотно-поступальні рухи ударника, внаслідок чого робочий матеріал руйнується від удару.</p> </div> </div>
Завантажити зображення	<div style="border: 1px solid #ccc; padding: 5px;"> Вибрати файли Файл не вибрано </div>
Тип категорії	Електрострумент ▼
Тип пропозиції	Нюанска ▼
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> РЕДАГУВАТИ ➔ НАЗАД </div>	



Рисунок 3.3.16. – редагування товару

Як вже було сказано вище, товар має велику кількість різних деталей у його описі, тому меню створення нового товару(рисунок 3.3.17) має такий самий широкий обсяг даних, як і меню редагування товару. Перша клітинка призначена для введення короткої назви товару. Наступна після неї йде клітинка вводу ціни. Варто зауважити, що в цю комірку можна вводити лише числові значення. У комірку «короткий опис» вводиться інформація про товар, яка буде відображатися під ним у каталозі, зазвичай це короткий його опис або повна назва з певними характеристиками товару. «Повний опис» - це, відповідно, поле для вводу усієї детальної інформації про товар. Ці дані будуть відображатися лише при перегляді конкретного товару. За допомогою кнопки «завантажити зображення» адміністратор може додати фото до інструменту, якщо воно є. Останні два пункти – це випадючі списки з вибором категорії товару та типом його пропозиції. Варто зазначити, що усі клітинки мають містити якусь

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		58

інформацію, у випадку, коли якийсь поле буде пусте користувач отримає повідомлення про це. Для того, щоб зберегти запис, потрібно натиснути кнопку «Створити»

The screenshot shows a web application interface for creating a product. At the top, there is a dark navigation bar with a logo, menu items like 'НОВАБУД', 'КЕРУВАННЯ САЙТОМ', 'ПЕРЕГЛЯД ЗАМОВЛЕНЬ', a shopping cart icon with '(2)', and user information 'ПРИВІТ, ADMIN@GMAIL.COM!' and 'ВИХІД'. The main content area is titled 'СТВОРИТИ ТОВАР'. It contains several input fields: 'Name', 'Ціна' (Price) with the value '0', 'Короткий опис' (Short description), and 'Повний опис' (Full description) which includes a rich text editor with a toolbar. Below these are file upload options: 'Завантажити зображення' (Upload images) with 'Вибрати файли' (Select files) and 'Файл не вибрано' (File not selected). There are also dropdown menus for 'Тип категорії' (Category type) and 'Тип пропозиції' (Proposal type). At the bottom of the form are two buttons: a black 'СТВОРИТИ' (Create) button and a green '← НАЗАД' (Back) button.

Пенцак Вадим 2023

Рисунок 3.3.17 – створення товару

3.4 Технічні характеристики веб- застосунку

Для того, щоб веб-застосунок правильно функціонував з клієнтської сторони потрібно забезпечення певними технічними вимогами:

- Операційна система Windows 7 та вище;
- Інтернет-браузер;
- Стабільне підключення до мережі Інтернет;
- 2-ядерний процесор ;

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		59

- 1 ГБ оперативної пам'яті;

Для правильного функціонування серверної сторони веб-застосунку потрібне таке забезпечення технічними вимогами:

- Операційна система Windows 7 та вище;
- Встановлений ASP.NET та всі потрібні фреймворки(ASP.NET MVC та Entity Framework);
- SQL Server Management Studio;
- SQL Server 8.0 та вище;
- Підключення до мережі Інтернет зі швидкістю не менше 15Мбіт/с;
- 4-ядерний процесор;
- 4ГБ оперативної пам'яті;
- 10ГБ внутрішньої пам'яті;

Усі вказані вище вимоги для нормального функціонування веб-застосунку є мінімальними.

3.5. Аналіз методів тестування веб- застосунку та розробка тестів

Існує кілька різних типів тестування веб- застосунків, за допомогою яких можна переконатися в якості та правильності його роботи. Деякі з найпоширеніших типів тестування веб- застосунків включають:

- функціональне тестування;
- тестування інтерфейсу;
- тестування продуктивності;
- тестування сумісності;
- тестування навантаження;

Функціональне тестування: це тип тестування, який перевіряє, чи працює функціонал програми належним чином. Тут виконуються вхідні та вихідні тести для перевірки правильності відповідей та обробки помилок.

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		60

Тестування інтерфейсу користувача (UI): цей тип тестування перевіряє, чи інтерфейс користувача веб-застосунку відповідає визначеним вимогам. Тестування передбачає перевірку зовнішнього вигляду, розташування та поведінки елементів інтерфейсу.

Тестування продуктивності: це тести, призначені для перевірки продуктивності веб-застосунку, наприклад часу завантаження сторінки, швидкості відгуку сервера, оптимізації використання ресурсів і обробки навантаження.

Тестування сумісності: цей тип тестування перевіряє, чи веб-застосунок працює на різних платформах, операційних системах і браузерах. Вони стежать за тим, щоб застосунок правильно відображався та працював на різних пристроях.

Тестування навантаження: це тестування виконується для визначення обмежень продуктивності веб-застосунку шляхом посилення навантаження на сервер і спостереження за тим, як він реагує на це навантаження.

Для перевірки роботи сайту було виконано функціональне тестування, який призначений перевірити чи працює програма належним чином. У таблиці 3.5.1 показані тестові сценарії

Таблиця 3.5.1 – тестовий сценарій №1

Дія	Очікуваний результат	Отриманий результат
Запуск веб-застосунку	Відкриття головної сторінки веб-застосунку	Головна сторінка відкрилася, усе наповнення відобразилося
Натиск на кнопку «Авторизація»	Відкриття сторінки веб-застосунку з формою авторизації	Сторінка відкрилася, усі елементи форми авторизації відобразилися

Продовження таблиці 3.5.1

Авторизація користувача	Після введення усіх даних користувача відкриється головна сторінка з функціоналом авторизованого користувача	Авторизація підтвердилася, відкрилася головна сторінка, з'явилися вкладка «Кошик» та «Налаштування профілю»
Перегляд детальної інформації про товар	Після натиску на кнопку «Переглянути деталі» під товаром відкриється сторінка з детальним описом товару та його характеристиками	Сторінка товару відкрилася, уся інформація про товар відобразилася на екрані. Кнопки «Додати у кошик» та «Назад до каталогу» з'явилися знизу сторінки
Натиск на кнопку «Назад до каталогу»	Відкриється головна сторінка сайту, відображається функціонал авторизованого користувача	Перехід назад на сторінку з каталогом, користувач залишився авторизованим
Вихід з особистого профілю	Після натиску на кнопку «Вихід», головна сторінка залишиться відкритою, але уже з виглядом для неавторизованого користувача	Вихід з облікового запису відбувся, головна сторінка залишилася відкритою, інформація відображається коректно

Таблиця 3.5.2 – тестовий сценарій №2

Дія	Очікуваний результат	Отриманий результат
Запуск веб-застосунку	Відкриття головної сторінки веб-застосунку	Головна сторінка відкрилася, усе наповнення відобразилося

Продовження таблиці 3.5.2

Натиск на кнопку «Реєстрація»	Відкриється вікно з формою реєстрації	Відкриття сторінки, виведення усіх елементів форми на екран
Реєстрація	Користувач введе дані, пройде перевірку правильності введення паролю, з'явиться повідомлення про реєстрацію	Реєстрація підтвердилася, усі дані були введені правильно, відкрилося повідомлення про успішну реєстрацію
Натиск на кнопку «Авторизація»	Відкриття сторінки веб-додатку з формою авторизації	Сторінка відкрилася, усі елементи форми авторизації відобразилися
Авторизація користувача	Після введення усіх даних користувача відкриється головна сторінка з функціоналом авторизованого користувача	Авторизація підтвердилася, відкрилася головна сторінка, з'явилися вкладка «Кошик» та «Налаштування профілю»
Натиск на одну з кнопок фільтрів	На екрані відобразатимуться лише товари обраної категорії	Частина записів зникла з екрану, на екрані залишилися лише товари обраної категорії
Натиск на кнопку «Додати у кошик»	З'явиться повідомлення про успішне додавання товару у кошик, у «шапці» сайту біля символу кошика з'явиться цифра	Після натиску з'явилося повідомлення, біля символу кошика з'явилася цифра

Кінець таблиці 3.5.2

Натиск на кнопку «Кошик»	Перехід на сторінку кошика	Відбувся перехід на сторінку з вмістом кошику, усі записи відобразилися
Натиснути кнопку «Видалити замовлення»	Один із записів зникне	Товар, біля його натиснули кнопку видалився і запис зник
Натиснути кнопку «Оформити замовлення»	Відбудеться перехід на нову сторінка з формою оформлення замовлення	Перехід відбувся, усі поля форми відобразилися
Натиснути кнопку «Підтвердити замовлення»	Після перевірки правильності введених даних відбудеться перехід на сторінку оплати	Відкрилася сторінка оплати
Вихід з особистого профілю	Після натиску на кнопку «Вихід», головна сторінка залишиться відкритою, але уже з виглядом для неавторизованого користувача	Вихід з облікового запису відбувся, головна сторінка залишилася відкритою, інформація відображається коректно

Після проведення тестування можна зазначити, що увесь функціонал відповідає усім вимогам до програмного забезпечення. Усі модулі застосунку працюють безперешкодно, усі випадки в яких застосунок працював некоректно з'ясовано та приведено до ладу.

3.6 Висновки до розділу 3

Під час виконання роботи над розділом 3 було виконано декілька етапів розробки застосунку. Спершу було побудовано фізичну базу даних, а також підключено її до модулів застосунку. Наступним кроком було реалізовано різні модулі застосунку, а також описано їх роботу. Після завершення розробки застосунку було написано керівництво користувача для усіх категорій користувачів(неавторизовані, авторизовані, адміністратор) з поясненням роботи усіх модулів веб-застосунку і супроводом рисунків. Останнім кроком стало визначення та виконання тестів для перевірки роботи веб-застосунку. Розробка веб-застосунку була успішною, всі тестові випадки пройдено з відмінним результатом.

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		65

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проаналізовано предметну область, з'ясована актуальність розробки веб-застосунку. Побудовано IDEF0-діаграми декомпозиції. Також було проведено аналіз існуючих рішень, на основі яких було побудовано таблицю порівняння для визначення основних переваг, які треба реалізувати у веб-додатку. Також опираючись на проведений аналіз було визначено усі функціональні та нефункціональні вимоги до розроблюваного веб-застосунку та визначені основні актори програмного забезпечення. Побудовано UML діаграму використання. На основі проаналізованої інформації сформоване технічне завдання.

На етапі проектування веб-застосунку було досліджено різні типи архітектури програмного забезпечення. В результаті аналізу було обрано клієнт-серверну архітектуру. Також на етапі проектування було побудовано логічну модель бази даних. Перед цим були з'ясовані усі сутності та атрибути, а також проведено нормалізацію бази даних і визначено зв'язки.

Спроектовано серверну частину веб-застосунку, описано роботу серверної частини, а також обрано шаблон MVC, за яким проектувалася серверна частина додатку. Побудовано UML діаграму послідовностей для демонстрації роботи одного із модулів. Також побудовано макети для клієнтської частини додатку, визначено дизайн веб-застосунку.

На основі проектування сайту були визначені засоби та технології, за допомогою яких буде спроектовано веб-застосунок. Вибір опирався на аналіз різних мов програмування, виборі різних фреймворків та аналізі підходящого типу бази даних та системи керування базою даних. Основними засобами і технологіями стали: платформа ASP.NET, мова програмування C#, фреймворки ASP.NET MVC та Entity Framework, СКБД Microsoft SQL Server та мова розміти HTML і каскадні таблиці стилів CSS.

					КвРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		66

У результаті реалізації додатку було створено фізичну базу даних опираючись на спроектовану раніше логічну модель бази даних. Наступним кроком стала реалізація основних модулів серверної частини веб-застосунку та реалізація клієнтської частини інтерфейсу користувача.

Останнім етапом стало тестування системи. Під час тестування програмного забезпечення було розроблено різні кейс-тести для перевірки справності роботи різних модулів програми. На основі проведеного тестування усі виявлені недоліки було вирішено.

У результаті виконання кваліфікаційної роботи поставлене завдання було виконано. Розроблений веб-застосунок для продажу будівельних інструментів відповідає усім вимогам, пройшов усі тестування і справно працює. Під час виконання кваліфікаційної роботи отримано нові навички у проектуванні та розробці веб-застосунків, а також засвоєні уже наявні знання.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		67

ПЕРЕЛІК ДЖЕРЕЛ

1. Бондарєв, В.М. "Об'єктно-орієнтовне програмування на С#": навч. посібник / В.М. Бондарєв. - Харків: Компанія СМІТ, 2009. - 224 с.
2. Павловський, В.В., Журавльова, І.С. "Веб-технології: HTML, CSS, JavaScript, PHP, AJAX, MySQL" / В.В. Павловський, І.С. Журавльова. - Київ: КНЕУ, 2016. - 375 с.
3. Литвин, А. "Розробка веб-додатків на PHP. Об'єктно-орієнтоване програмування" / А. Литвин. - Київ: ДМК Прес, 2017. - 384 с.
4. Запорожець, О.В. "Системи управління базами даних та знань в інтернет-технологіях" / О.В. Запорожець. - Київ: НУХТ, 2013. - 192 с.
5. Батинцев, В.А. "Основи мережевого адміністрування" / В.А. Батинцев. - Київ: ТОВ "Техніка", 2015. - 224 с.
6. Дуб, О.О., Луценко, Д.В. "ASP.NET: Розробка веб-додатків" / О.О. Дуб, Д.В. Луценко. - Київ: Видавничий дім "Ін Юре", 2018. - 336 с.
7. Бабенко, О.В., Гриценко, І.М., Сирота, Є.М. "Веб-програмування: HTML, CSS, JavaScript, PHP, MySQL" / О.В. Бабенко, І.М. Гриценко, Є.М. Сирота. - Київ: ННЦ "ІНФРА-М", 2017. - 288 с.
8. Литвин, А. "Основи роботи з ASP.NET: Практикум" / А. Литвин. - Київ: ДМК Прес, 2012. - 280 с.
9. Штайнберг, А. "ASP.NET MVC 5: Розробка додатків" / А. Штайнберг. - Київ: БХВ-Петербург, 2014. - 544 с.
10. Шевченко, О.В. "Програмування на С# для початківців" / О.В. Шевченко. - Київ: Навчальна книга – Богдан, 2019. - 400 с.
11. Стахів, І.М. "Веб-програмування: Підручник" / І.М. Стахів. - Львів: Світ, 2016. - 336 с.
12. Корнійчук, М.М. "Розробка веб-додатків на платформі ASP.NET" / М.М. Корнійчук. - Київ: Центр учбової літератури, 2012. - 320 с.
13. Степаненко, І.В., Мазур, В.В. "Основи веб-програмування" / І.В. Степаненко, В.В. Мазур. - Харків: ФОП Жуков Є.В., 2019. - 320 с.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

14. Ігнатенко, С.А. "Програмування мовою С#: Навчальний посібник" / С.А. Ігнатенко. - Київ: Видавничий дім "Ін Юре", 2018. - 352 с.
15. Гордієнко, О.В., Гордієнко, О.О. "ASP.NET: Використання фреймворка" / О.В. Гордієнко, О.О. Гордієнко. - Київ: Видавничий дім "Ін Юре", 2019. - 352 с.
16. Барановський, І.М., Кобельський, А.А. "Основи веб-розробки" / І.М. Барановський, А.А. Кобельський. - Київ: Видавничо-поліграфічний центр "Київський університет", 2015. - 344 с.
17. Іванов, В.В., Кузьмін, М.М. "Веб-програмування з використанням технологій ASP.NET та Silverlight" / В.В. Іванов, М.М. Кузьмін. - Київ: Видавництво "Алерта", 2010. - 416 с.
18. Гарбуз, С.В. "Основи інтернет-технологій" / С.В. Гарбуз. - Київ: Центр учбової літератури, 2014. - 320 с.
19. Карпенко, С.М., Герасимов, Ю.В. "ASP.NET MVC 4: Розробка веб-приложений" / С.М. Карпенко, Ю.В. Герасимов. - Київ: Диалектика, 2013. - 288 с.
20. Халап, І.В. "Основи програмування в середовищі .NET" / І.В. Халап. - Київ: Навчальна книга – Богдан, 2019. - 320 с.
21. Шарун, А.В. "Програмування на мові С#" / А.В. Шарун. - Київ: Кондор, 2013. - 352 с.
22. Курдюмов, А.Г., Курдюмова, О.А. "Програмування на мові С#. Розробка ігор" / А.Г. Курдюмов, О.А. Курдюмова. - Київ: ТОВ "Видавничий дім "Слово", 2017. - 416 с.
23. Шварц, О.М., Телепін, А.В. "ASP.NET: Розробка веб-додатків" / О.М. Шварц, А.В. Телепін. - Київ: ДУХ І ЛІТЕРА, 2012. - 460 с.
24. Жарков, І.М., Мясоєдов, О.М. "ASP.NET MVC: Створення веб-додатків" / І.М. Жарков, О.М. Мясоєдов. - Київ: Видавництво "Алерта", 2012. - 336 с.
25. Мирзоян, А.А., Матяш, А.В. "ASP.NET. Быстрый старт" / А.А. Мирзоян, А.В. Матяш. - Київ: Диалектика, 2014. - 288 с.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		69

26. Дубінін, Є.В., Жукова, В.М. "ASP.NET MVC: Професійний підхід" / Є.В. Дубінін, В.М. Жукова. - Київ: Диалектика, 2015. - 384 с.
27. Харченко, О.В. "Основи програмування на мові С#" / О.В. Харченко. - Київ: Видавничий дім "Ін Юре", 2016. - 352 с.
28. Гаврилюк, В.М., Довгий, В.І. "ASP.NET 4.5 і чистий код" / В.М. Гаврилюк, В.І. Довгий. - Київ: Київський університет, 2014. - 288 с.
29. Шевченко, А.В., Клочко, І.О. "ASP.NET 4.5 з використанням С#" / А.В. Шевченко, І.О. Клочко. - Київ: Центр учбової літератури, 2013. - 432 с.
30. Бондаренко, Є.А., Лещинський, І.М. "ASP.NET 4.5: Створення веб-додатків" / Є.А. Бондаренко, І.М. Лещинський. - Київ: Видавництво "Слово", 2014. - 520 с.
31. Васильєв, І.С., Черевко, А.А. "ASP.NET. Технології AJAX і DHTML" / І.С. Васильєв, А.А. Черевко. - Київ: Видавничо-поліграфічний центр "Київський університет", 2016. - 344 с.
32. Синявський, В.В., Кучеренко, А.А. "ASP.NET 3.5" / В.В. Синявський, А.А. Кучеренко. - Київ: Диалектика, 2009. - 416 с.
33. Котляр, І.В. "Основи програмування в середовищі Visual Studio 2019" / І.В. Котляр. - Київ: Видавничий дім "Ін Юре", 2019. - 400 с.
34. Бобер, В.В. "ASP.NET 4.0 для професіоналів" / В.В. Бобер. - Київ: Диалектика, 2010. - 960 с.
35. Лабунець, О.П., Мальцев, А.В. "Основи програмування на мові С#. Навчальний посібник" / О.П. Лабунець, А.В. Мальцев. - Київ: НУХТ, 2013. - 220 с.
36. Іванчук, О.Є., Боднарчук, Ю.С. "ASP.NET: Технології створення веб-додатків" / О.Є. Іванчук, Ю.С. Боднарчук. - Київ: Київський університет, 2009. - 424 с.
37. Молчанов, А.В. "ASP.NET 3.5. Професійне програмування" / А.В. Молчанов. - Київ: Диалектика, 2008. - 1088 с.
38. Біловодський, В.І., Поліщук, О.О. "ASP.NET. Практичний курс" / В.І. Біловодський, О.О. Поліщук. - Київ: Диалектика, 2010. - 736 с.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

39. Васильєв, В.В., Черевко, А.А. "ASP.NET. Базовий курс" / В.В. Васильєв, А.А. Черевко. - Київ: Видавничий дім "Ін Юре", 2013. - 480 с.

40. Шумило, М.В., Шарун, А.В. "ASP.NET: Швидкий старт" / М.В. Шумило, А.В. Шарун. - Київ: Діалектика, 2015. - 272 с.

					КВРІПЗ. 190138.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		71

ДОДАТОК А

Технічне завдання

Введення

Робота виконується в рамках проекту розробки веб-застосунку для продажу будівельних інструментів, у вигляді сайту, що дозволяє здійснювати онлайн покупки.

1 Підстава для розробки

Підставою для розробки є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри інженерії програмного забезпечення. Найменування розробки: Веб-застосунок для продажу будівельних інструментів.

2 Призначення розробки

Призначення веб-застосунку для продажу будівельних інструментів полягає у представленні каталогу товарів, інформації про товар(опис та характеристики), здійснення покупок товарів та оформлення доставки придбаних товарів. Користувачами веб-застосунку є користувач мережі Інтернет

Функціональне призначення полягає у: додаванні товарів до каталогу, редагування інформації про товар, реєстрації та авторизації користувачів, оформленні замовлення та оплати замовлення.

Редагування наповнення та будь-якої інформації на сайті може відбуватися адміністратором відповідно до вимог. Для використання ПЗ не потрібно ніякого додаткового налаштування і воно працює на усіх комп'ютерах, які відповідають технічним вимогам.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Веб-застосунок для продажу будівельних інструментів має забезпечувати наступні функції:

- додавання товарів до каталогу;

- реєстрація та авторизація у системі;
- налаштування особистого профілю(редагування даних, зміна паролю);
- відображення товарів та інформації про них;
- фільтрація для пошуку товарів;
- додавання товарів до кошику;
- оформлення замовлення;

3.2 Вимоги до надійності

Система має виконувати наступні вимоги до надійності:

- розмежування прав користувачів у системі;
- валідація на стороні сервера для контролю інформації, що вводиться та виключення введення помилкових даних користувачем;
- підтримання захисту від несанкціонованого доступу;
- розмежування прав користувачів у системі;

3.3 Умови експлуатації

Умови експлуатації мають відповідати санітарним і технічним нормам експлуатації персонального комп'ютера, при температурі та відносній вологості навколишнього середовища, визначених для персональної обчислювальної техніки згідно з ГОСТ 15150-69 [7].

Для обслуговування веб-застосунку допускаються тільки спеціально навчені адміністратори або розробники. До користування веб-застосунком допускаються користувачі мережі Інтернет.

3.4 Вимоги до складу та параметрів технічних засобів

- ОС Windows 7 та вище;
- Веб-браузер;
- Дискретний або інтегрований відеоадаптер;
- Оперативна пам'ять 1ГБ;
- Жорсткий диск 1ГБ;

- Девайси(миша, клавіатура);
- Монітор;
- Під'єднання до мережі Інтернет;

3.5 Вимоги до інформаційної та програмної сумісності

Для створення клієнтської частини веб-застосунку використовувалася мова розмітки HTML, каскадні таблиці стилів CSS та мова програмування JS

Для створення серверної частини веб-застосунку використовувалися технології платформи ASP.NET, мова програмування C# та фреймворки ASP.NET MVC і Entity Framework

3.6 Спеціальні вимоги

Інтуєтивно-зрозумілий інтерфейс, простий дизайн та легкий у користуванні функціонал застосунку.

4 Вимоги до програмної документації

Набір документації, що надається у момент здачі проекту:

- Опис модулів програми з демонстрацією їх роботи
- Текст програми з коментарями
- Керівництво користувача для звичайних користувачів та адміністратора
- Довідкова інформація
- Керівництво для розробника

5 Стадії та етапи розробки

Стадії та етапи розробки веб-застосунку для продажу будівельних інструментів продемонстровано у таблиці 1

Таблиця 1 - стадії та етапи розробки

Стадія розробки	Етапи робіт	Зміст робіт
Технічне завдання 02.01 – 31.01.2023	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання
Ескізний проект 01.02 – 14.02 2023	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися
Технічний проект 15.02 – 28.02 2023	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів
Робочий проект 01.03 – 10.04.2023	Розробка програмного забезпечення	Реалізація програмного забезпечення; відладка; проведення попереднього тестування
Розробка програмної документації 11.04 – 20.04.2023	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням
Тестування системи 21.04 – 30.04.2023	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Підготовка і передача програмного забезпечення; навчання персоналу використовуванню програмного забезпечення; внесення коректувань в програмне забезпечення і документацію

6 Порядок контролю та приймання

Контроль здійснюється користувачами застосунку підключеними на етапі тестування. Прийом програми здійснюється після повного її розгортання, тестування і налаштування для справного функціонування

ДОДАТОК Б

Код(лістинг)програми:

ProductController.cs:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;

namespace Controllers
{
    [Authorize(Roles = WC.AdminRole)]
    public class ProductController : Controller
    {
        private readonly IProductRepository _prodRepo;
        private readonly IWebHostEnvironment _webHostEnvironment;

        public ProductController(IProductRepository prodRepo, IWebHostEnvironment webHostEnvironment)
        {
            _prodRepo = prodRepo;
            _webHostEnvironment = webHostEnvironment;
        }

        public IActionResult Index()
        {
            IEnumerable<Product> objList = _prodRepo.GetAll(includeProperties:
"Category,ApplicationType");
            return View(objList);
        }

        [HttpGet]
```

```

public IActionResult Upsert(int? id)
{
    ProductVM = new ProductVM()
    {
        Product = new Product(),
        CategorySelectList = _prodRepo.GetAllDropDownList(WC.CategoryName),
        ApplicationTypeSelectList =
        _prodRepo.GetAllDropDownList(WC.ApplicationTypeName),
    };
    if (id != null)
    {
        productVM.Product = _prodRepo.Find(id.GetValueOrDefault());
        if (productVM.Product == null)
        {
            return NotFound();
        }
    }
    return View(productVM);
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Upsert(ProductVM productVM)
{
    if (ModelState.IsValid)
    {
        var files = HttpContext.Request.Form.Files;
        string webRootPath = _webHostEnvironment.WebRootPath;

        if (productVM.Product.Id == 0)
        {
            // Creating
            string uniqueFileName = ProcessUploadedFile(files, webRootPath);
            productVM.Product.Image = uniqueFileName;
            _prodRepo.Add(productVM.Product);
        }
    }
}

```

```

    }
    else
    {
        // Updating
        var objFromDb = _prodRepo.FirstOrDefault(u => u.Id == productVM.Product.Id,
isTracking: false);
        if (files.Count > 0)
        {
            DeleteOldFile(objFromDb.Image, webRootPath);
            string uniqueFileName = ProcessUploadedFile(files, webRootPath);
            productVM.Product.Image = uniqueFileName;
        }
        else
        {
            productVM.Product.Image = objFromDb.Image;
        }
        _prodRepo.Update(productVM.Product);
    }
    TempData[WC.Success] = "Action completed successfully";
    _prodRepo.Save();
    return RedirectToAction("Index");
}

productVM.CategorySelectList = _prodRepo.GetAllDropDownList(WC.CategoryName);
productVM.ApplicationTypeSelectList =
_prodRepo.GetAllDropDownList(WC.ApplicationTypeName);

return View(productVM);
}

private string ProcessUploadedFile(IFormFileCollection files, string webRootPath)
{
    string uniqueFileName = null;

    foreach (var file in files)
    {

```

```

    if (file.Length > 0)
    {
        string uploadsFolder = Path.Combine(webRootPath, WC.ImagePath);
        uniqueFileName = Guid.NewGuid().ToString() + "_" + file.FileName;
        string filePath = Path.Combine(uploadsFolder, uniqueFileName);
        using (var fileStream = new FileStream(filePath, FileMode.Create))
        {
            file.CopyTo(fileStream);
        }
    }
}
return uniqueFileName;
}
private void DeleteOldFile(string fileName, string webRootPath)
{
    string filePath = Path.Combine(webRootPath, WC.ImagePath, fileName);
    if (System.IO.File.Exists(filePath))
    {
        System.IO.File.Delete(filePath);
    }
}
[HttpGet]
public IActionResult Delete(int? id)
{
    if (id == null || id == 0)
    {
        return NotFound();
    }
    Product = _prodRepo.FirstOrDefault(u => u.Id == id, includeProperties:
"Category,ApplicationType");
    if (product == null)
    {
        return NotFound();
    }
}

```

```

        return View(product);
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    public IActionResult Delete(int id)
    {
        var obj = _prodRepo.Find(id);
        if (obj == null)
        {
            return NotFound();
        }
        DeleteOldFile(obj.Image, _webHostEnvironment.WebRootPath);
        _prodRepo.Remove(obj);
        _prodRepo.Save();
        TempData[WC.Success] = "Action completed successfully";
        return RedirectToAction("Index");
    }
}
}

```

CardController:

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Claims;
using System.Text;
using System.Threading.Tasks;
using Braintree;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Identity.UI.Services;

```

```

using Microsoft.AspNetCore.Mvc;
{
    [Authorize]
    public class CartController : Controller
    {
        private readonly IWebHostEnvironment _webHostEnvironment;
        private readonly IEmailSender _emailSender;
        private readonly IApplicationUserRepository _userRepo;
        private readonly IProductRepository _prodRepo;
        private readonly IIquiryHeaderRepository _inqHRepo;
        private readonly IIquiryDetailRepository _inqDRepo;
        private readonly IOrderHeaderRepository _orderHRepo;
        private readonly IOrderDetailRepository _orderDRepo;
        private readonly IBrainTreeGate _brain;
        [BindProperty]
        public ProductUserVM { get; set; }
        public CartController(IWebHostEnvironment webHostEnvironment, IEmailSender
emailSender,
            IApplicationUserRepository userRepo, IProductRepository prodRepo,
            IIquiryHeaderRepository inqHRepo, IIquiryDetailRepository inqDRepo,
            IOrderHeaderRepository orderHRepo, IOrderDetailRepository orderDRepo, IBrainTreeGate
brain)
        {
            _webHostEnvironment = webHostEnvironment;
            _emailSender = emailSender;
            _brain = brain;
            _userRepo = userRepo;
            _prodRepo = prodRepo;
            _inqDRepo = inqDRepo;
            _inqHRepo = inqHRepo;
            _orderDRepo = orderDRepo;
            _orderHRepo = orderHRepo;
        }
        public IActionResult Index()
        {

```

```

        List<ShoppingCart> shoppingCartList = new List<ShoppingCart>();
if(HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart)!=null
    && HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart).Count()
> 0)
    {
        //session exists
        shoppingCartList = HttpContext.Session.Get<List<ShoppingCart>>(WC.SessionCart);
    }
    List<int> prodInCart = shoppingCartList.Select(i => i.ProductId).ToList();
    IEnumerable<Product> prodListTemp = _prodRepo.GetAll(u =>
prodInCart.Contains(u.Id));
    IList<Product> prodList = new List<Product>();
    foreach (var cartObj in shoppingCartList)
    {
        Product prodTemp = prodListTemp.FirstOrDefault(u => u.Id == cartObj.ProductId);
        prodTemp.TempSqFt = cartObj.SqFt;
        prodList.Add(prodTemp);
    }
    return View(prodList);
}
[HttpPost]
[ValidateAntiForgeryToken]
[ActionName("Index")]
public IActionResult IndexPost(IEnumerable<Product> ProdList)
{
    List<ShoppingCart> shoppingCartList = new List<ShoppingCart>();
    foreach (Product prod in ProdList)
    {
        shoppingCartList.Add(new ShoppingCart { ProductId = prod.Id, SqFt = prod.TempSqFt
});
    }

    HttpContext.Session.Set(WC.SessionCart, shoppingCartList);
    return RedirectToAction(nameof(Summary));
}

```

```

public IActionResult Summary()
{
    ApplicationUser;

    if(User.IsInRole(WC.AdminRole))
    {
        if (HttpContext.Session.Get<int>(WC.SessionInquiryId) != 0)
        {
            //cart has been loaded using an inquiry
            InquiryHeader = _inqHRepo.FirstOrDefault(u => u.Id ==
HttpContext.Session.Get<int>(WC.SessionInquiryId));
            applicationUser = new ApplicationUser()
            {
                Email = inquiryHeader.Email,
                FullName = inquiryHeader.FullName,
                PhoneNumber = inquiryHeader.PhoneNumber
            };
        }
        else
        {
            applicationUser = new ApplicationUser();
        }
        var gateway = _brain.GetGateway();
        var clientToken = gateway.ClientToken.Generate();
        ViewBag.ClientToken = clientToken;
    }
    else
    {
        var claimsIdentity = (ClaimsIdentity)User.Identity;
        var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);
        //var userId = User.FindFirstValue(ClaimTypes.Name);
        applicationUser = _userRepo.FirstOrDefault(u => u.Id == claim.Value);
    }
}

```

```

    }
    List<ShoppingCart> shoppingCartList = new List<ShoppingCart>();
    if (HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart) != null
        && HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart).Count()
> 0)
    {
        //session exists
        shoppingCartList = HttpContext.Session.Get<List<ShoppingCart>>(WC.SessionCart);
    }
    List<int> prodInCart = shoppingCartList.Select(i => i.ProductId).ToList();
    IEnumerable<Product> prodList = _prodRepo.GetAll(u => prodInCart.Contains(u.Id));

    ProductUserVM = new ProductUserVM()
    {
        ApplicationUser = applicationUser,
    };
    foreach(var cartObj in shoppingCartList)
    {
        Product prodTemp = _prodRepo.FirstOrDefault(u => u.Id == cartObj.ProductId);
        prodTemp.TempSqFt = cartObj.SqFt;
        ProductUserVM.ProductList.Add(prodTemp);
    }
    return View(ProductUserVM);
}

[HttpPost]
[ValidateAntiForgeryToken]
[ActionName("Summary")]
public async Task<IActionResult> SummaryPost(IFormCollection collection, ProductUserVM
ProductUserVM)
{
    var claimsIdentity = (ClaimsIdentity)User.Identity;
    var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);

    if (User.IsInRole(WC.AdminRole))
    {

```

```

//we need to create an order

//var orderTotal = 0.0;

//foreach(Product prod in ProductUserVM.ProductList)
//{
//  orderTotal += prod.Price * prod.TempSqFt;
//}

OrderHeader = new OrderHeader()
{
    CreatedByUserId = claim.Value,
    FinalOrderTotal = ProductUserVM.ProductList.Sum(x=>x.TempSqFt*x.Price),
    City = ProductUserVM.ApplicationUser.City,
    StreetAddress = ProductUserVM.ApplicationUser.StreetAddress,
    State = ProductUserVM.ApplicationUser.State,
    PostalCode = ProductUserVM.ApplicationUser.PostalCode,
    FullName = ProductUserVM.ApplicationUser.FullName,
    Email = ProductUserVM.ApplicationUser.Email,
    PhoneNumber = ProductUserVM.ApplicationUser.PhoneNumber,
    OrderDate = DateTime.Now,
    OrderStatus = WC.StatusPending
};

_orderHRepo.Add(orderHeader);
_orderHRepo.Save();

foreach (var prod in ProductUserVM.ProductList)
{
    OrderDetail = new OrderDetail()
    {
        OrderHeaderId = orderHeader.Id,
        PricePerSqFt = prod.Price,
        Sqft = prod.TempSqFt,
        ProductId = prod.Id
    };
    _orderDRepo.Add(orderDetail);
}

_orderDRepo.Save();

```

```

string nonceFromTheClient = collection["payment_method_nonce"];
var request = new TransactionRequest
{
    Amount = Convert.ToDecimal(orderHeader.FinalOrderTotal),
    PaymentMethodNonce = nonceFromTheClient,
    OrderId=orderHeader.Id.ToString(),
    Options = new TransactionOptionsRequest
    {
        SubmitForSettlement = true
    }
};
var gateway = _brain.GetGateway();
Result<Transaction> result = gateway.Transaction.Sale(request);

if (result.Target.ProcessorResponseText == "Approved")
{
    orderHeader.TransactionId = result.Target.Id;
    orderHeader.OrderStatus = WC.StatusApproved;
}
else
{
    orderHeader.OrderStatus = WC.StatusCancelled;
}
_orderHRepo.Save();
return RedirectToAction(nameof(InquiryConfirmation), new { id=orderHeader.Id });
}
else
{
    //we need to create an inquiry
    var PathToTemplate = _webHostEnvironment.WebRootPath +
Path.DirectorySeparatorChar.ToString()
+ "templates" + Path.DirectorySeparatorChar.ToString() +
"Inquiry.html";
    var subject = "New Inquiry";

```

```

string HtmlBody = "";
using (StreamReader sr = System.IO.File.OpenText(PathToTemplate))
{
    HtmlBody = sr.ReadToEnd();
}
//Name: { 0}
//Email: { 1}
//Phone: { 2}
//Products: {3}
StringBuilder productListSB = new StringBuilder();
foreach (var prod in ProductUserVM.ProductList)
{
    productListSB.Append($" - Name: { prod.Name } <span style='font-size:14px;'> (ID:
{prod.Id})</span><br />");
}
string messageBody = string.Format(HtmlBody,
    ProductUserVM.ApplicationUser.FullName,
    ProductUserVM.ApplicationUser.Email,
    ProductUserVM.ApplicationUser.PhoneNumber,
    productListSB.ToString());

await _emailSender.SendEmailAsync(WC.EmailAdmin, subject, messageBody);
InquiryHeader = new InquiryHeader()
{
    ApplicationUserId = claim.Value,
    FullName = ProductUserVM.ApplicationUser.FullName,
    Email = ProductUserVM.ApplicationUser.Email,
    PhoneNumber = ProductUserVM.ApplicationUser.PhoneNumber,
    InquiryDate = DateTime.Now
};
_inqHRepo.Add(inquiryHeader);
_inqHRepo.Save();
foreach (var prod in ProductUserVM.ProductList)
{

```

```

        InquiryDetail = new InquiryDetail()
        {
            InquiryHeaderId = inquiryHeader.Id,
            ProductId = prod.Id,
        };
        _inqDRepo.Add(inquiryDetail);
    }
    _inqDRepo.Save();
    TempData[WC.Success] = "Inquiry submitted successfully";
}
return RedirectToAction(nameof(InquiryConfirmation));
}
public IActionResult InquiryConfirmation(int id=0)
{
    OrderHeader = _orderHRepo.FirstOrDefault(u => u.Id == id);
    HttpContext.Session.Clear();
    return View(orderHeader);
}
public IActionResult Remove(int id)
{
    List<ShoppingCart> shoppingCartList = new List<ShoppingCart>();
    if (HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart) != null
        && HttpContext.Session.Get<IEnumerable<ShoppingCart>>(WC.SessionCart).Count()
> 0)
    {
        //session exists
        shoppingCartList = HttpContext.Session.Get<List<ShoppingCart>>(WC.SessionCart);
    }
    shoppingCartList.Remove(shoppingCartList.FirstOrDefault(u => u.ProductId == id));
    HttpContext.Session.Set(WC.SessionCart, shoppingCartList);
    return RedirectToAction(nameof(Index));
}
[HttpPost]
[ValidateAntiForgeryToken]

```

```
public IActionResult UpdateCart(IEnumerable<Product> ProdList)
{
    List<ShoppingCart> shoppingCartList = new List<ShoppingCart>();
    foreach(Product prod in ProdList)
    {
        shoppingCartList.Add(new ShoppingCart { ProductId = prod.Id, SqFt = prod.TempSqFt
});
    }
    HttpContext.Session.Set(WC.SessionCart, shoppingCartList);
    return RedirectToAction(nameof(Index));
}
public IActionResult Clear()
{
    HttpContext.Session.Clear();
    return RedirectToAction("Index","Home");
}
}
```

ДОДАТОК В

(презентаційні матеріали)

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення



Кваліфікаційна робота на тему: «Веб-додаток для продажу будівельних матеріалів»

Виконав:
студент IV курсу, група ІПЗ19-1
Пенцак В.О.

Керівник:
д-р фіз.-мат. наук, професор
Бедратюк Л.П.

МЕТА ТА ЗАВДАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ



МЕТОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ Є РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ЯКЕ ДОЗВОЛИТЬ АВТОМАТИЗУВАТИ РОБОТУ ПІДПРИЄМСТВА, А ТАКОЖ МАТИМЕ СУЧАСНИЙ ТА ІНТУЄТИВНО-ЗРОЗУМЛИЙ ІНТЕРФЕЙС, ЩО СИСТЕМАТИЗУЄ ВЗАЄМОДІЮ ПРОДАВЦЯ ТА КЛІЄНТА

ЗАВДАННЯ, ЯКІ ВИРІШУВАТИМУТЬСЯ ПІД ЧАС ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ:

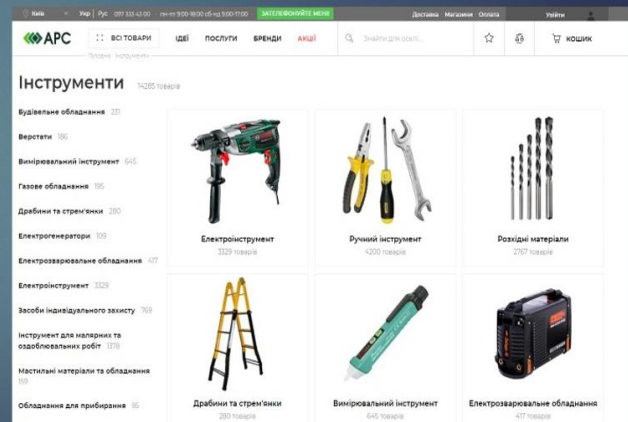
- ВИЗНАЧЕННЯ АКТУАЛЬНОСТІ ТЕМИ ТА ЇЇ ПРАКТИЧНОГО ЗАСТОСУВАННЯ;
- ДОСЛІДЖЕННЯ НАЯВНИХ МЕТОДІВ
- ВИРІШЕННЯ ТА ЇХ АНАЛІЗ;
- ДОСЛІДЖЕННЯ РІЗНИХ ТИПІВ БАЗ ДАНИХ;
- ВИБІР ТА ОБГРУНТУВАННЯ АРХІТЕКТУРИ ПЗ;
- ОПИС ОСОБЛИВОСТЕЙ ТА ФУНКЦІОНАЛУ СЕРВЕРНОЇ ТА КЛІЄНТСЬКОЇ ЧАСТИНИ ЗАСТОСУНКУ
- АНАЛІЗ РІЗНИХ МЕТОДІВ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ;
- РОЗРОБКА БАЗИ ДАНИХ;
- РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ТА КЛІЄНТСЬКОГО ІНТЕРФЕЙСУ;
- РОЗРОБКА КЕРІВНИЦТВА КОРИСТУВАЧА
- АНАЛІЗ МЕТОДІВ ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКІВ;
- ЗДІЙСНЕННЯ ТЕСТУВАННЯ ПЗ;

НАЯВНІ МЕТОДИ ВИРІШЕННЯ ТА ЇХ АНАЛІЗ



ТИПОВИЙ ПРЕДСТАВНИК ТЕМАТИЧНИХ ІНТЕРНЕТ-МАГАЗИНІВ, ЯКИЙ ВКЛЮЧАЄ В СОБІ ОСНОВНІ ВИМОГИ ДЛЯ ТОГО, ЩОБ ВВАЖАТИСЯ ВДАЛИМ РІШЕННЯМ

- ВЛУЧНО ПІДБРАНИЙ ДИЗАЙН САЙТУ;
- ЗРУЧНА ПАНЕЛЬ ВИБОРУ КАТЕГОРІЙ;
- ІНФОРМАТИВНИЙ ТА ЗРОЗУМЛИЙ ІНТЕРФЕЙС;
- ДОБРЕ РЕАЛІЗОВАНА НАВИГАЦІЯ;
- ШИРОКИЙ ПОШУК ТОВАРУ ЗА ФІЛТРАМИ



АКТУАЛЬНІСТЬ ТЕМИ



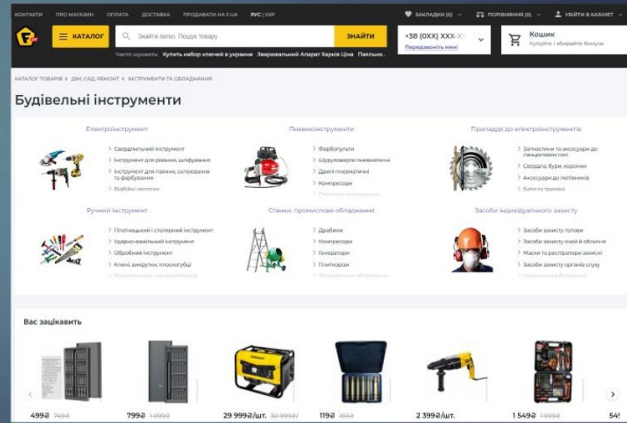
АКТУАЛЬНІСТЬ ТЕМИ полягає у тому, що на теперішній час, БІЛЬШІСТЬ ПІДПРИЄМЦІВ ВЗЯЛИ КУРС НА ЕЛЕКТРОННУ КОМЕРЦІЮ, ТОБТО ПЕРЕХІД ДО ВВЕДЕННЯ СВОГО БІЗНЕСУ САМЕ ЗА ДОПОМОГОЮ ІНТЕРНЕТ-МАГАЗИНІВ ПОПИТ НА СТВОРЕННЯ ТАКИХ ВЕБ-ЗАСТОСУНКІВ НЕ СПАДАЄ, А НАВІТЬ ПРОДОВЖУЄ РОСТИ. ОБ'ЄКТИВНИЙ ПОГЛЯД ДАЄ ЗРОЗУМІТИ, ЩО Є ПОТРЕБА СТВОРЕННЯ ТАКОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, АДЖЕ ЙОГО СТВОРЕННЯ ТАКОЖ Є КОМЕРЦІЙНО ПРИВАБЛИВИМ, БО ТАКИМ ЧИНОМ ДАЄ ЗМОГУ ОТРИМАТИ НОВИХ КЛІЄНТІВ, А ТАКОЖ ДАЄ ЗМОГУ ДЕЩО МОДЕРНІЗУВАТИ БІЗНЕС-ПРОЦЕСИ ПІДПРИЄМСТВА

НАЯВНІ МЕТОДИ ВИРІШЕННЯ ТА ЇХ АНАЛІЗ



ОКРЕМА КАТЕГОРІЯ, ЩО ВІДНОСИТЬСЯ ДО МАРКЕТПЛЕЙСІВ ТОБТО ВЕЛИКИЙ ІНТЕРНЕТ-МАЙДАНЧИК, В ЯКОМУ Є ОКРЕМИЙ «ВІДДІЛ» З ТЕМАТИЧНИМ ВМІСТОМ

- ПРОСТИЙ, АЛЕ ВІДМІННО ПІДБРАНИЙ ДИЗАЙН;
- РОЗГОРНУТА ПАНЕЛЬ КАТЕГОРІЙ;
- ІНТУЇТИВНО-ЗРОЗУМЛИЙ ІНТЕРФЕЙС;
- ПРОСТА ТА ЗРУЧНА НАВІГАЦІЯ;
- ЗРУЧНА ПАНЕЛЬ ПОШУКУ ТОВАРІВ ЗА НАЗВОЮ;

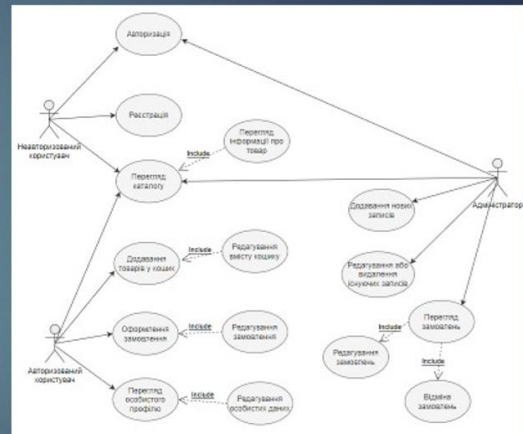


ОСНОВНІ АКТОРИ ПЗ



Визначення основних акторів та побудова діаграми варіантів використання - етапи, що значно спрощують визначення вимог до програмного забезпечення

Актор	Короткий опис
Неавторизований користувач	Має можливість переглядати асортимент
Авторизований користувач	Має ті ж можливості, що і незареєстрований користувач, окрім цього може додавати товари до кошику, редагувати вміст кошику, здійснювати оформлення доставки та сплату покупки
Адміністратор	Може додавати/змінювати усі категорії та товари, змінювати замовлення клієнтів та опрацьовувати запити на оформлення замовлення.



ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



ДЛЯ ЗВИЧАЙНОГО КОРИСТУВАЧА:

РЕЄСТРАЦІЯ/АВТОРИЗАЦІЯ;
ПЕРЕГЛЯД КАТАЛОГУ ТОВАРІВ;
ПЕРЕГЛЯД ІНФОРМАЦІЇ ПРО ТОВАР;
ДОДАВАННЯ ТОВАРУ У КОШИК;
РЕДАГУВАННЯ ВІСТУ КОШИКА;
ОФОРМЛЕННЯ ЗАМОВЛЕННЯ;
ПЕРЕГЛЯД ОСОБИТ СОГО КАБІНЕТУ;
НАЛАШТУВАННЯ ОСОБИТ СОГО КАБІНЕТУ;

ДЛЯ АДМІНІСТРАТОРА:

РЕЄСТРАЦІЯ/АВТОРИЗАЦІЯ;
ДОДАВАННЯ НОВИХ ЗАПИСІВ;
РЕДАГУВАННЯ/ВИДАЛЕННЯ ІСНУЮЧИХ ЗАПИСІВ;
ПЕРЕГЛЯД ЗАМОВЛЕНЬ;
СТВОРЕННЯ НОВИХ ЗАМОВЛЕНЬ;
РЕДАГУВАННЯ ТА ВІДСЛІДКОВУВАННЯ СТАНУ
ЗАМОВЛЕННЯ;
СТВОРЕННЯ ОБЛІКОВИХ ЗАПИСІВ З ПРАВАМИ
АДМІНІСТРАТОРА

ПОСТАНОВКА ЗАДАЧ



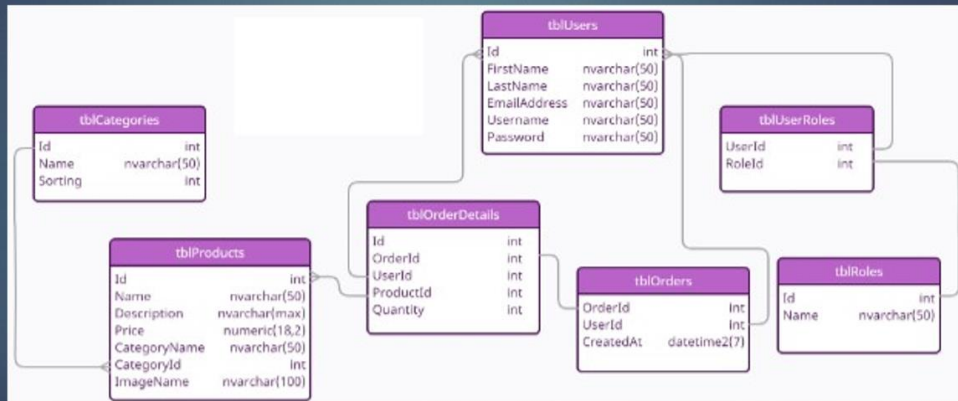
ОТЖЕ, ДОСЛІДИВШИ ПРЕДМЕТНУ ОБЛАСТЬ, ПРОАНАЛІЗУВАВШИ НАЯВНЕ ПРОГРАМНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ, ТА ВИЗНАЧИВШИ ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ БУЛО ПОСТАВЛЕНО ТАКІ ЗАДАЧІ:

- РЕАЛІЗАЦІЯ КАТАЛОГУ ТОВАРІВ;
- РЕАЛІЗАЦІЯ РЕЄСТРАЦІЇ ТА АВТОРИЗАЦІЇ КОРИСТУВАЧА;
- РЕАЛІЗАЦІЯ НАЛАШТУВАННЯ ОСОБИТ СОГО ПРОФІЛЮ КОРИСТУВАЧА;
- РЕАЛІЗАЦІЯ КОШИКУ ТОВАРІВ;
- РЕАЛІЗАЦІЯ ОПЛАТИ;
- РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ КЕРУВАННЯ САЙТОМ ДЛЯ АДМІНІСТРАТОРА (ДОДАВАННЯ НОВИХ ЗАПИСІВ, РЕДАГУВАННЯ ТА ВИДАЛЕННЯ ІСНУЮЧИХ);
- РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ ДЛЯ КЕРУВАННЯ ТА ВІДСЛІДКОВУВАННЯ ЗАМОВЛЕНЬ ДЛЯ АДМІНІСТРАТОРА



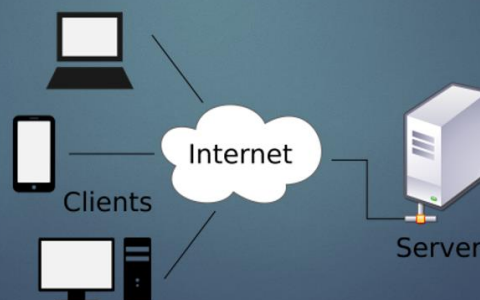
Логічна модель бази даних

НА ОСНОВІ УСІХ ЗІБРАНИХ ДАНИХ, ПІСЛЯ З'ЯСУВАННЯ УСІХ СУТНОСТЕЙ,ЩО БУДУТЬ У БАЗІ ДАНИХ, З'ЯСУВАННЯ АТРИБУТІВ, ЩО БУДУТЬ У КОЖНІЙ СУТНОСТІ,ВНАЧЕННЯ ОБМЕЖЕНЬ, ЯКІ БУДУТЬ ЗАСТОСОВУВАТИСЯ ДЛЯ СУТНОСТЕЙ І ПРОВЕДЕННЯ НОРМАЛІЗАЦІЇ БАЗИ ДАНИХ БУЛО ПОБУДОВАНО ЛОГІЧНУ МОДЕЛЬ БАЗИ ДАНИХ



Архітектура веб-застосунку

КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА Є ОДНІЄЮ З НАЙПОШИРЕНІШИХ АРХІТЕКТУР ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У ЦЬЙ АРХІТЕКТУРІ ФУНКЦІЇ ТА ОБОВ'ЯЗКИ РОЗПОДІЛЯЮТЬСЯ МІЖ ДВОМА ОСНОВНИМИ СКЛАДОВИМИ: КЛІЄНТАМИ І СЕРВЕРАМИ. КЛІЄНТИ — ЦЕ КОРИСТУВАЧІ АБО ПРОГРАМИ, ЯКІ ЗВЕРТАЮТЬСЯ ДО СЕРВЕРА ДЛЯ ОТРИМАННЯ ПЕВНИХ ПОСЛУГ АБО РЕСУРСІВ СЕРВЕРІ — ЦЕ КОМПОНЕНТИ, ЯКІ НАДАЮТЬ ЦІ ПОСЛУГИ АБО РЕСУРСИ КЛІЄНТАМ



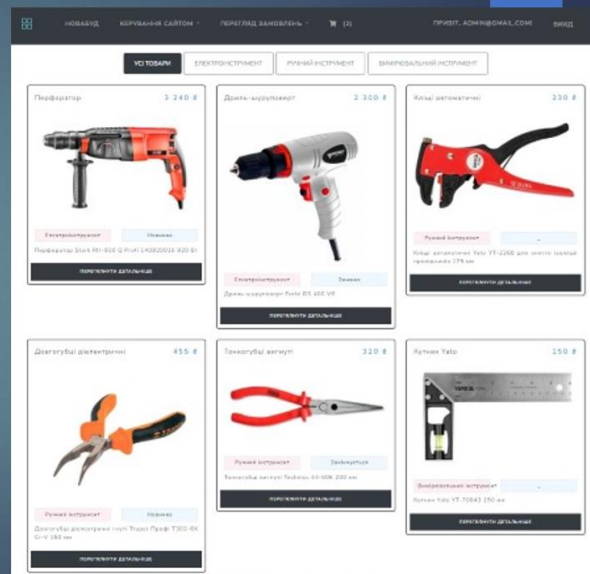
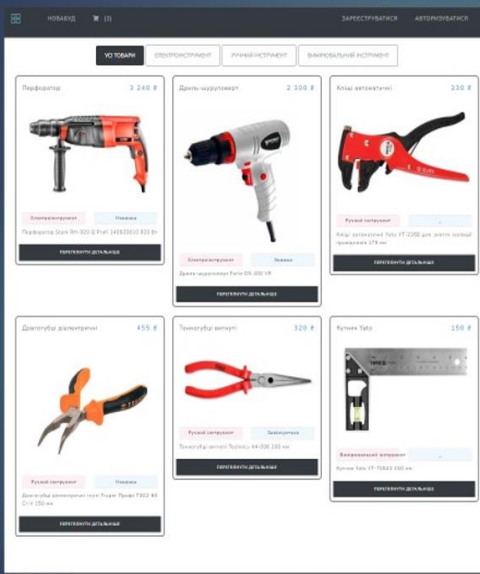
Засоби розробки ПЗ



ВЕБ-ДОДАТОК РОЗРОБЛЯВСЯ ЗАСОБАМИ ТА ТЕХНОЛОГІЯМИ ПЛАТФОРМИ ASP.NET. ASP.NET – ПЛАТФОРМА ДЛЯ РОЗРОБКИ ВЕБ-ЗАСТОСУНКІВ. КЛІЄНТСЬКА ЧАСТИНА ДОДАТКУ РОЗРОБЛЯЛАСЯ ЗА ДОПОМОГОЮ HTML, CSS ТА JAVASCRIPT. СЕРВЕРНА ЧАСТИНА ЗАСТОСУНКУ РОЗРОБЛЯЛАСЯ ЗА ДОПОМОГОЮ C#, ТАК ЯК ЦЯ МОВА ПРОГРАМУВАННЯ Є МОВОЮ, ЩО ДОЗВОЛЯЄ СТВОРЮВАТИ ЗАСТОСУНКИ, ЩО ВИКОНУЮТЬСЯ В .NET, ЧАСТИНОЮ ЯКОГО Є ASP.NET. ЗАСТОСУНОК РОЗРОБЛЯВСЯ НА ОСНОВІ ПЛАТФОРМИ ASP.NET, ПРИ ЦЬОМУ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ ASP.NET MVC. ВІДМІННІСТЮ ЦЬОГО ФРЕЙМВОРКУ Є ТЕ, ЩО УСІ ЗАСТОСУНКИ СТВОРЕНІ НА ЙОГО ОСНОВІ РЕАЛІЗУЮТЬСЯ ЗА ДОПОМОГОЮ ШАБЛОНУ MVC(МОДЕЛЬ-ПРЕДСТАВЛЕННЯ-КОНТРОЛЕР). У ЯКОСТІ СУБД ДЛЯ ЗАСТОСУНКУ ЗАСТОСОВУВАВСЯ MICROSOFT SQL SERVER.



РЕАЛІЗАЦІЯ ДОДАТКУ.ГОЛОВНА СТОРІНКА



РЕАЛІЗАЦІЯ ДОДАТКУ. РЕЄСТРАЦІЯ/АВТОРИЗАЦІЯ



NOBAYUD 🛒 (0) РЕЄСТРАЦІЯ АВТОРИЗАЦІЯ

РЕЄСТРАЦІЯ

Логін

Електронна пошта

+380 Номер телефону

Пароль Підтвердження паролю

РЕЄСТРАЦІЯ

2023 квіт

NOBAYUD 🛒 (0) РЕЄСТРАЦІЯ АВТОРИЗАЦІЯ

АВТОРИЗАЦІЯ

Електронна пошта

Пароль

ВХОДИ

Забудили свій пароль?

РЕЄСТРАЦІЯ

2023 квіт



РЕАЛІЗАЦІЯ ДОДАТКУ.КОШИК ТОВАРІВ



NOBAYUD 🛒 (2) ПРИВІТ, VADYM@GMAIL.COM! ВІДОД

Кошик

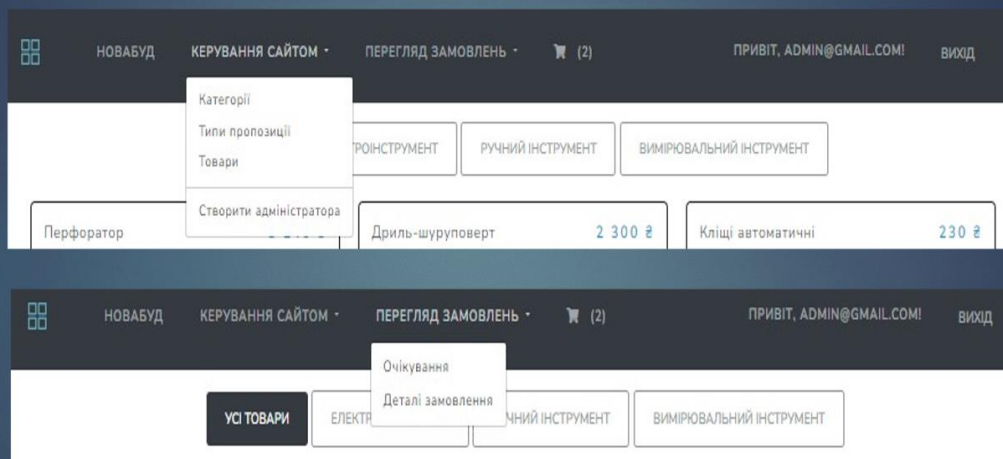
ОЧИСТИТИ КОШИК ПОВЕРНУТИСЯ ДО ТОВАРІВ

Ціна за одиницю	Ціна за одиницю
 ДРИЛЬ-ШУРУПОВЕРТ Дриль шуруповерт Forte DS 400 VLi	\$2300/одиниця <input type="text"/>
 ДОВГОГУБЦІ ДІЕЛЕКТРИЧНІ Довгогубці діелектричні леза Тисер Проріз Т302 6X Сг V 150 мм	\$455/одиниця <input type="text"/>

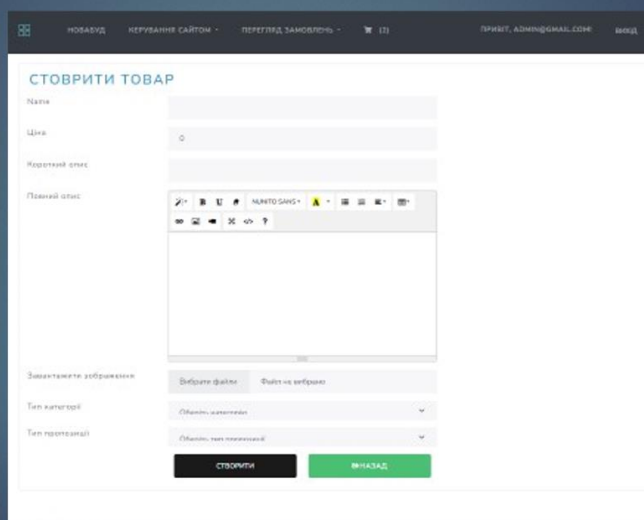
ПЕРЕЙТИ ДО ОФОРМЛЕННЯ

Пенцзяк Вадим 2023

РЕАЛІЗАЦІЯ ДОДАТКУ.КЕРУВАННЯ САЙТОМ















РЕАЛІЗАЦІЯ ДОДАТКУ.СТВОРЕННЯ ЗАПИСІВ



РЕАЛІЗАЦІЯ ДОДАТКУ.РЕДАГУВАННЯ ЗАПИСІВ



СПИСОК ТОВАРІВ

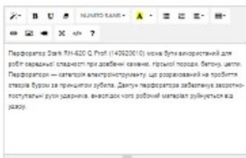
НАЗВА ТОВАРУ	ЦІНА	КАТЕГОРІЯ	ТИП ПРОПОЗИЦІЇ	
Перфоратор	3240	Електроінструмент	Новинка	 
Дельта-азуловарт	2200	Електроінструмент	Знижка	 
Кілер автоматичні	330	Ручний інструмент	—	 
Двогубці діелектричні	455	Ручний інструмент	Промітка	 
Танцювальні окуляри	320	Ручний інструмент	Занекуплено	 
Купчик Yabo	150	Діагностичний інструмент	—	 

РЕДАГУВАННЯ ТОВАРУ

Назва: Перфоратор

Річок: 3240

Короткий опис: Перфоратор Skilz RM-830 Q Plus 1.65x200x100 830 Вт

Повний опис: 
Перфоратор Skilz RM-830 Q Plus (1.65x200x100) нова лінійка інструментів для роботи в будівництві. Спеціально розроблений для роботи в будівництві. Перфоратор — це електричний інструмент, що розроблений на базі лінійки Skilz. Він має високу продуктивність. Діє як ударний інструмент. Поступово розробляється, вивчаючи роботу інструментів на ринку.

Завантажити зображення:

Тип категорії: Електроінструмент

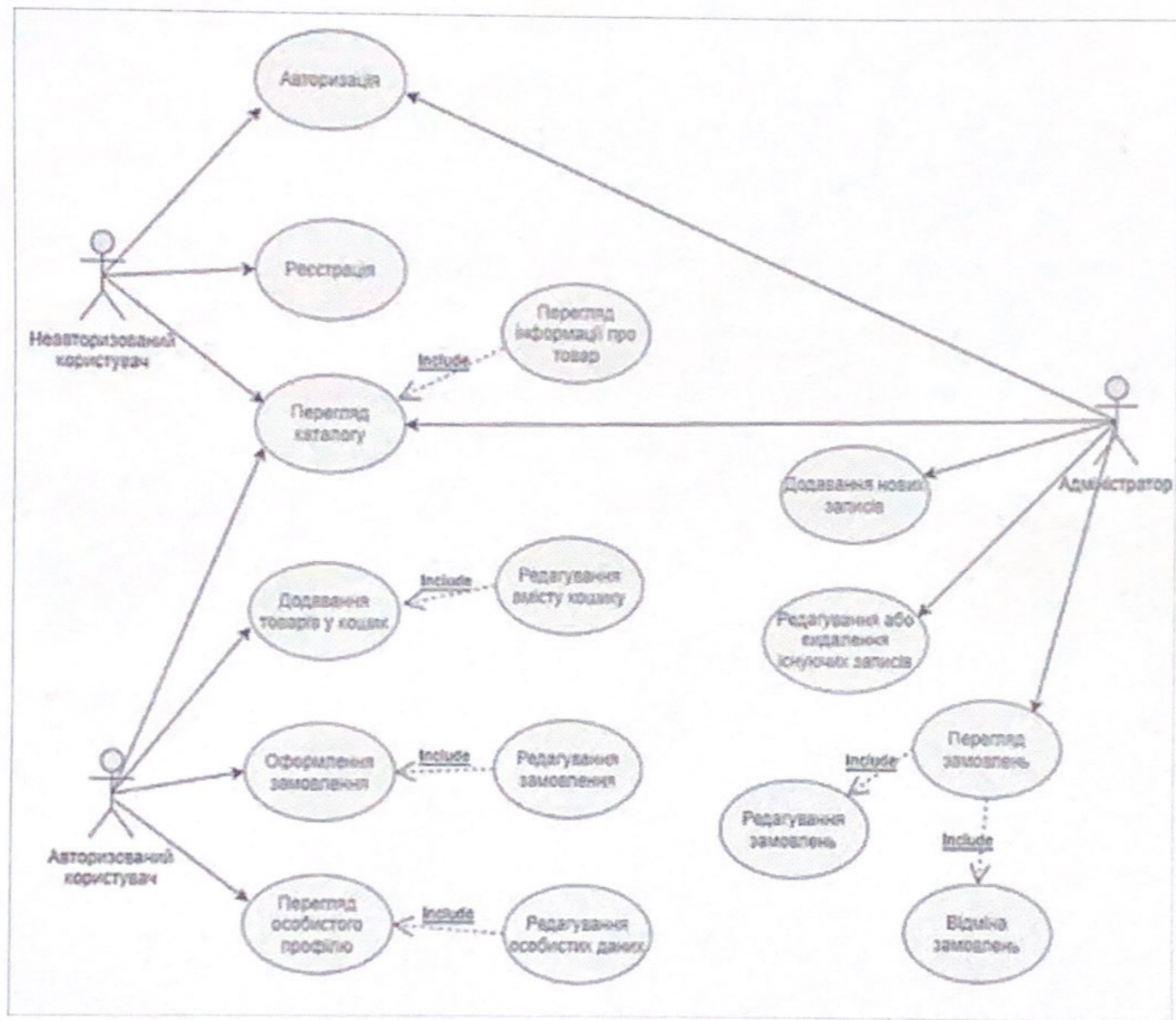
Тип пропозиції: Новинка

ВИСНОВКИ

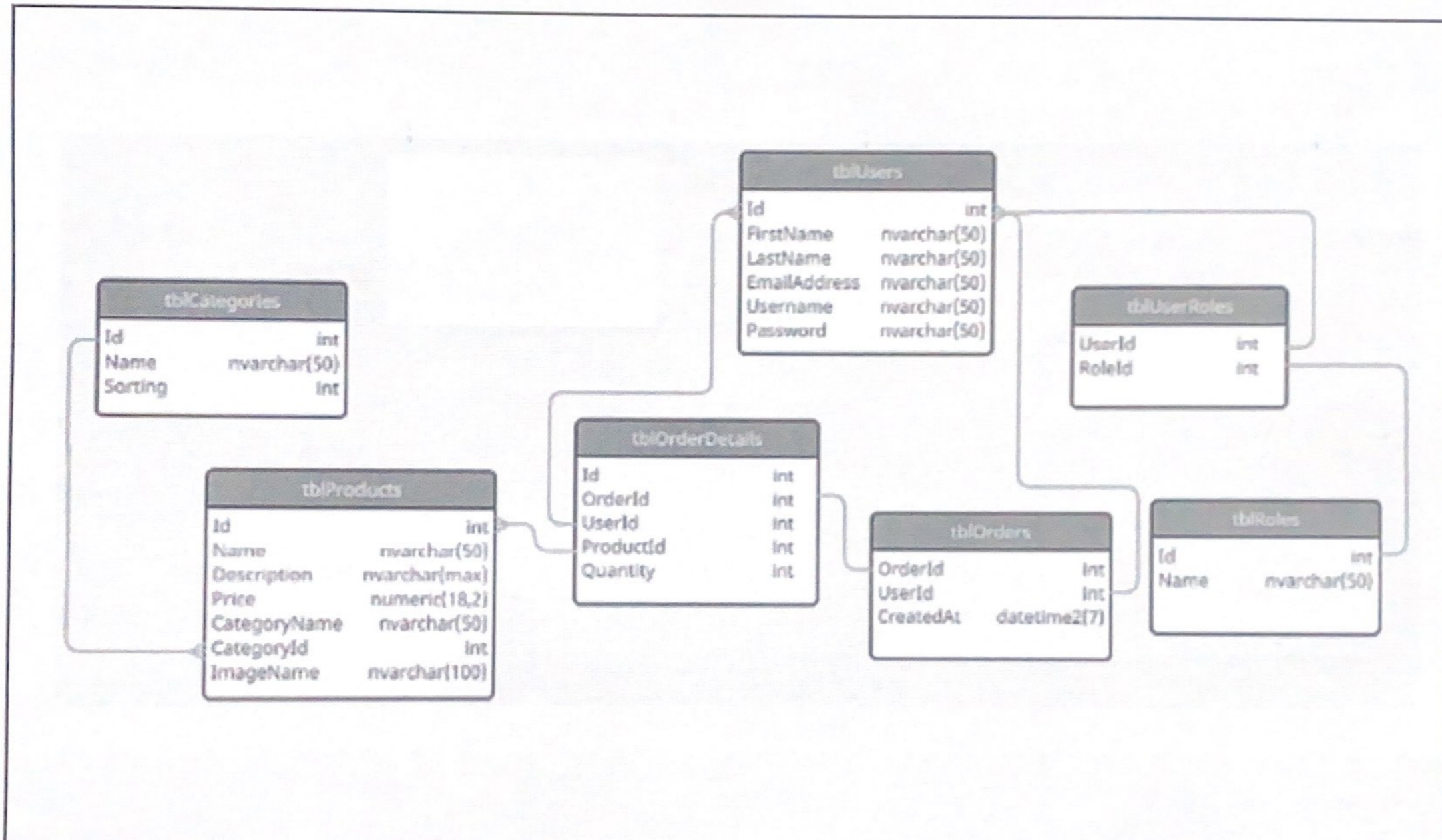


У процесі виконання кваліфікаційної роботи, було проведено аналіз у галузі інтернет-магазинів будівельних інструментів. Проведено аналіз існуючих рішень, визначено їх основні переваги та недоліки, для врахування під час реалізації додатку. Проведено аналіз засобів розробки та здійснено їх обґрунтований вибір з поясненням усіх переваг. Здійснено опис усіх наявних інструментів, які застосовувалися під час розробки. Проаналізовано наявні типи баз даних та здійснено вибір бази даних, яка підходить за усіма критеріями. Описані основні актори ПЗ та побудовано діаграми для опису роботи додатку. У результаті виконаної роботи було реалізовано ПЗ для автоматизації роботи підприємства, яке має простий та інтуїтивно-зрозумілий інтерфейс.

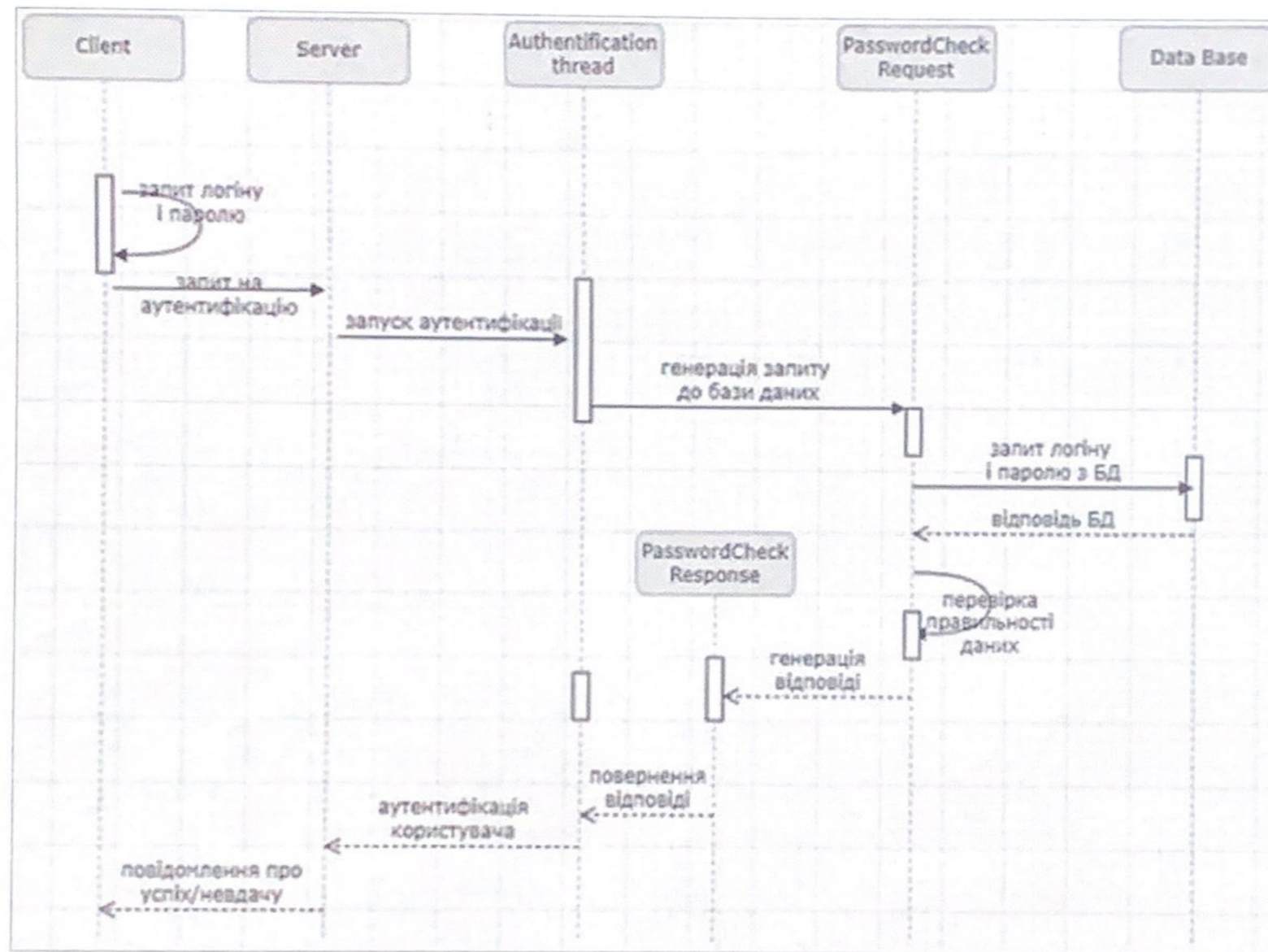
Графічна частина



				КвРІПЗ.190138.01.15.Е8		
Зм. Док.	№ док.	Підпис	Дата	Літера	Маса	Масштаб
Розробив	Тенчик В.О.	<i>[Signature]</i>	05.08			
Коривни	Бадратюк Л.П.	<i>[Signature]</i>	05.08	Аркуш 1	Аркушів 3	
Н. Контр.	Гурман І.В.	<i>[Signature]</i>	05.08	ХНУ, ІПЗ-19-1		
Зад. Кат.	Бадратюк Л.П.	<i>[Signature]</i>	05.08			



				КвРІПЗ.190138.01.15.Е8			
Зм. Док.	№ докум.	Підпис	Дата	ER-діаграма бази даних	Літос	Маса	Масштаб
Розробив	Танцюк В.О.	<i>[Signature]</i>	5.08		Аркуш 2	Аркуш 3	
Керує	Бедратюк Л.П.	<i>[Signature]</i>	5.08				
Н. Контр.	Гурман І.В.	<i>[Signature]</i>	3.06	ХНУ, ІПЗ-19-1			
Зад. Каф.	Бедратюк Л.П.	<i>[Signature]</i>	5.08				



				КвРІПЗ.190138.01.15.E8			
Зм.	Док.	№ докм.	Підпис	Дата	Листів	Маса	Масштаб
Розробив	Тенчак В.О.		<i>[Signature]</i>	6.08			
Корівник	Бодратюк Л.П.		<i>[Signature]</i>	8.08			
Н. Контр.	Гурман І.В.		<i>[Signature]</i>	8.08			
Зва. Каф.	Бодратюк Л.П.		<i>[Signature]</i>	8.08			
					Листів 3		Листів 3
					ХНУ, ІПЗ-19-1		

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Дипломник Пенцак Вадим Олександрович

Тема Веб-застосунок для продажу будівельних інструментів

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 ; кількість сторінок записки 72

1. Короткий зміст пояснювальної записки та прийнятих рішень у кваліфікаційній роботі було 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 архітектуру, яка буде використовуватися при розробці. У третьому розділі побудовано фізичну модель бази даних, підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт. Після цього було написано керівництво користувача і проведено тестування системи

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є актуальною, оскільки сьогодні попит на різні веб-застосунки для продажу товарів постійно зростає. Під час розробки застосунку було враховано недоліки існуючих рішень, а також застосовано нові технології

5. Негативні сторони роботи У роботі пошук товарів реалізовано лише по категоріям – було б доцільно додати пошук по назві або розширену сітку фільтрів

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 Говорущенко Тетяна Олександрівна, доктор технічних наук, професор, зав.кафедри комп'ютерної інженерії та інформаційних систем (КІПС) ХНУ

“ 5 ” сервісо 2023 р.


(підпис)

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

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

Назва: «Веб-застосунок для продажу будівельних інструментів»

Автор: Пенцак Вадим Олександрович

Спеціальність: 121 – Інженерія програмного забезпечення

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

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

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.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої 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 загальнозживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/ схожості, складає 4,73% і адресується до 174 джерел з Інтернету і 69 джерел з Бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата _____

Завідувач кафедри



Леонід БЕДРАТЮК

Гарант освітньої програми



Леонід БЕДРАТЮК

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

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

здобувача вищої освіти

Пенцака В.О.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-19-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/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 плагіату в текстах робіт.

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

1.06.2023
дата


Прізвище

Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1015380210

Дата перевірки:
01.06.2023 22:50:21 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
01.06.2023 22:52:25 EEST

ID користувача:
100005589

Назва документа: КвР Пенцак ІПЗ-19-1(для перевірки)

Кількість сторінок: 75 Кількість слів: 11251 Кількість символів: 86889 Розмір файлу: 2.70 MB ID файлу: 1015045648

4.73% Схожість

Найбільша схожість: 2.55% з джерелом з Бібліотеки (ID файлу: 1015045634)

2.39% Джерела з Інтернету

174

Сторінка 77

3.64% Джерела з Бібліотеки

69

Сторінка 78

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

1

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 8%

ID: 114530 Назва: БКР Веб-застосунок для продажу будівельних інструментів Додано в БД: 2023-06-01 Автора: Пенцак В.О. Керівники: Бедрацюк Л.П. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	65835	573	3427 (5%)	42 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми