

# ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

## «АРХІТЕКТУРА ПРОГРАМНОГО КОМПЛЕКСУ ІНТЕРНЕТ-ПЛАТФОРМИ ПІДПРИЄМСТВА ТОРГІВЛІ»

Студента 2м курсу, 2 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми «Інженерія  
програмного забезпечення»

Міщенко Володимира  
Юрійовича

\_\_\_\_\_

підпис студента

Науковий керівник  
PhD, доцент кафедри  
інженерії програмного  
забезпечення та кібербезпеки

Десятко Альона  
Миколаївна

\_\_\_\_\_

підпис керівника

Гарант освітньої програми  
кандидат педагогічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

Котенко Наталія  
Олексіївна

\_\_\_\_\_

підпис гаранта

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Освітня програма 121 «Інженерія програмного забезпечення»

**Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«13» грудня 2022 р.

**Завдання**

**на випускн кваліфікаційну роботу студентів**

Міщенко Володимир Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Архітектура програмного  
комплексу інтернет-платформи підприємства торгівлі»

Затверджена наказом ректора від «06» грудня 2022 р. № 3285

2. Строк здачі студентом закінченої роботи 1 грудня 2023

3. Цільова установка та вихідні дані до роботи

Мета роботи розробка архітектури програмного комплексу для інтернет-  
платформи підприємства торгівлі

Об'єкт дослідження інтернет-платформа, яка використовується на  
підприємстві торгівлі

Предмет дослідження є архітектура програмного продукту (інтернет-  
платформа підприємства торгівлі).



4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Актуальність предметної області

1.2. Моделювання предметної області

1.3. Огляд наявних аналогів

1.4. Постановка задачі, визначення вимог та цілей

1.5. Проектування логічної структури

1.4. Висновки до розділу 1

РОЗДІЛ 2 МОДЕЛЮВАННЯ АРХІТЕКТУРИ ТА БАЗИ ДАНИХ

2.1. Моделювання предметної області

2.2. Огляд різних архітектурних стилів

2.3. Вибір оптимальної архітектури для підприємства торгівлі

2.4. Опис структури бази даних

2.5. Висновки до розділу 2

РОЗДІЛ 3 ЗАСОБИ РЕАЛІЗАЦІЇ

3.1. Інструменти для розробки

3.2. Опис інтерфейсу програми

3.3. Основні функції інтерфейсу користувача

3.4. Умови виконання програми

3.5. Повідомлення користувачу

3.6. Керівництво адміністратора

3.7. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ТЕСТУВАННЯ ДОДАТКА

ДОДАТКИ





## 6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	07.11.2022	07.11.2022
2.	<i>Розробка та затвердження завдання на роботу магістра (стац/заоч)</i>	13.12.2022	13.12.2022
3.	<i>Вступ та перелік літературних джерел</i>	24.02.2023	24.02.2023
4.	<i>Розробка технічного завдання</i>	15.03.2023	15.03.2023
5.	<i>Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</i>	10.04.2023	10.04.2023
6.	<i>Розділ 2. РЕАЛІЗАЦІЯ</i>	24.05.2023	24.05.2023
7.	<i>Розділ 3. ТЕХНОЛОГІЧНИЙ РОЗДІЛ</i>	06.09.2023	06.09.2023
8.	<i>Розробка програми та методики тестування</i>	18.10.2023	18.10.2023
9.	<i>Написання наукової статті</i>	17.05.2023	17.05.2023
10.	<i>Керівництво користувача</i>	25.10.2023	25.10.2023
11.	<i>Висновки та пропозиції</i>	01.11.2023	01.11.2023
12.	<i>Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)</i>	06.11.2023	06.11.2023
13.	<i>Підготовка автореферату та презентації доповіді</i>	06.11.2023	06.11.2023
14.	<i>Попередній захист випускної кваліфікаційної роботи</i>	20.11.2023 – 24.11.2023	20.11.2023 – 24.11.2023
15.	<i>Здача зброшурованої випускної кваліфікаційної роботи</i>	01.12.2023	01.12.2023
16.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	02.12.2023	02.12.2023
17.	<i>Підготовка до публічного захисту випускної кваліфікаційної роботи</i>	05.12.2023- 06.12.2023	05.12.2023- 06.12.2023

7. Дата видачі завдання «13» грудня 2022 р.

8. Науковий керівник випускної кваліфікаційної роботи \_\_\_\_\_

Десятко А.М.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми \_\_\_\_\_

Котенко Н.О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент \_\_\_\_\_

Міщенко В.Ю.

(прізвище, ініціали, підпис)





## АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена моделюванню архітектури інтернет-платформи підприємства торгівлі та програмній реалізації. Випускна кваліфікаційна робота на тему «Архітектура програмного комплексу інтернет-платформи підприємства торгівлі» містить 55 сторінок, 29 рисунків, 1 таблицю. Перелік використаних джерел налічує 16 найменувань.

Було проаналізовано предметну область, визначено вимоги і цілі, створено логічну структуру, вибрано оптимальну архітектуру та базу даних для інтернет-платформи.

В результаті розроблена архітектура інтернет-платформи підприємства торгівлі та програмна реалізація та програмно реалізовано у вигляду інтернет-платформи.

**Ключові слова:** Інтернет-платформа, сервер, архітектура програмного комплексу, HTML, CSS, JAVASCRIPT, C#

## ABSTRACT

In accordance with the purpose of the study, the work is devoted to modeling the architecture of the Internet platform of a trade enterprise and its software implementation. The final qualification work on the topic "Architecture of the software complex of the Internet platform of a trade enterprise" contains 55 pages, 29 figures, 1 table. The list of references includes 16 titles.

The subject area was analyzed, requirements and goals were defined, a logical structure was created, and the optimal architecture and database for the Internet platform were selected.

As a result, the architecture of the Internet platform of the trade enterprise and the software implementation were developed and programmatically implemented in the form of a website.

Keywords: Internet platform, server, software architecture, HTML, CSS, JAVASCRIPT, C#



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних, використовуються для зберігання, організації та доступу до інформації.

MSSQL – Microsoft SQL Server, система управління базами даних (СУБД), розроблена компанією Microsoft.

HTML – Hypertext Markup Language, стандартна мова розмітки, яка використовується для створення та відображення веб-сторінок у веб-браузерах.

CSS – Cascading Style Sheets, мова, яка використовується для опису зовнішнього вигляду та стилю веб-сторінок, створених за допомогою HTML.

C# - об'єктно-орієнтована мова програмування розроблена компанією Microsoft.

JavaScript- високорівнева, інтерпретована мова програмування, яка використовується для створення динамічних та інтерактивних інтернет-платформів.

Чат-бот- програмне забезпечення, розроблене для автоматичного взаємодії з користувачами через чатові інтерфейси.

Сервер - комп'ютер або програма, яка надає ресурси, послуги або дані іншим комп'ютерам.

MVC - архітектурний шаблон програмування, який розділяє програму на три компоненти: Model (модель), View (вид), Controller (контролер).

<i>ДТЕУ 121 02-11.МР</i>							
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.		Криворучко О.В.		19.09.23			
Керівник		Десятко А.М.		19.09.23			
Гарант		Котенко Н.О.		19.09.23			
Розробив		Міщенко В.Ю.		19.09.23			
<i>Перелік умовних скорочень</i>							
<i>архітектура програмного комплексу інтернет- платформи підприємства торгівлі</i>					<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
					<i>ПС</i>	<i>2</i>	<i>55</i>
<i>Факультет інформаційних технологій</i>							
<i>2м курс, 2 група</i>							

<b>ЗМІСТ</b>	
<b>ВСТУП</b> .....	<b>3</b>
<b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	<b>7</b>
1.1. Актуальність предметної області.....	7
1.2. Огляд наявних аналогів.....	9
1.3. Постановка задачі, визначення вимог та цілей.....	13
1.4 Проектування логічної структури.....	18
1.5 Висновки до Розділу 1.....	19
<b>РОЗДІЛ 2 МОДЕЛЮВАННЯ АРХІТЕКТУРИ ТА БАЗИ ДАНИХ</b> .....	<b>21</b>
2.1. Моделювання предметної області.....	21
2.2. Огляд різних архітектурних стилів.....	28
2.3. Вибір оптимальної архітектури для підприємства торгівлі.....	31
2.4. Опис структури бази даних.....	32
2.5 Висновки до Розділу 2.....	34
<b>РОЗДІЛ 3 ЗАСОБИ РЕАЛІЗАЦІЇ</b> .....	<b>36</b>
3.1. Інструменти для розробки.....	36
3.2. Опис інтерфейсу програми.....	38
3.3. Основні функції інтерфейсу користувача.....	40
3.4. Умови виконання програми.....	46
3.5. Повідомлення користувачу.....	47
3.6. Керівництво адміністратора.....	48
3.7 Висновки до Розділу 3.....	51
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ</b> .....	<b>53</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	<b>54</b>
<b>ТЕХНІЧНЕ ЗАВДАННЯ</b> .....	<b>56</b>
<b>ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ</b> .....	<b>59</b>
<b>ДОДАТКИ</b> .....	<b>55</b>
<b>АКТ ВПРОВАДЖЕННЯ РОБІТ</b> .....	<b>60</b>

<i>ДТЕУ 121 02-11.МР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		01.11.23
Керівник		Десятко А.М.		01.11.23
Гарант		Котенко Н.О.		01.11.23
Розробив		Міщенко В.Ю		01.11.23
<i>Архітектура програмного комплексу інтернет-платформи підприємства торгівлі</i>				
<i>Зміст</i>				
<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>		
<i>Зміст</i>	3	55		
<i>Факультет інформаційних технологій 2м курс, 2 група</i>				



## ВСТУП

У сучасних умовах розвитку електронної комерції та зростання конкуренції в сфері торгівлі, інтернет-платформи стають незамінним інструментом для підприємств, що бажають успішно працювати на ринку. Інтернет-платформи підприємств торгівлі стають основою для ефективної взаємодії з клієнтами, просування продуктів та послуг, а також підвищення конкурентоспроможності.

Розробка ефективної архітектури програмного комплексу для інтернет-платформи підприємства торгівлі є актуальним завданням, яке дозволяє забезпечити ефективне управління бізнесом, покращити якість обслуговування клієнтів та забезпечити конкурентні переваги.

В рамках розробки інтернет-платформи підприємства торгівлі, ключовим етапом є проектування та розробка архітектури програмного комплексу, яка визначає основні засади, структуру та компоненти платформи.

Мета дипломного проекту полягає у розробці архітектури програмного продукту для інтернет-платформи підприємства торгівлі, яка відповідає сучасним вимогам електронної комерції та забезпечує ефективну роботу платформи для підвищення ефективності бізнес-процесів, автоматизації операцій та забезпечення зручного та безпечного середовища для клієнтів.

Важливо розробити архітектуру, яка забезпечить зручну навігацію, швидкий доступ до інформації, безпеку та надійність операцій.

Об'єктом дослідження є сама інтернет-платформа, яка

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 02-11.МР</i>			
Зав. каф.		Криворучко О.В.		24.02.23	<i>Архітектура програмного комплексу інтернет-платформи підприємства торгівлі</i>	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		24.02.23		В	4	55
Гарант		Котенко Н.О.		24.02.23		<i>Факультет інформаційних технологій 2м курс, 2 група</i>		
Розробив		Міщенко В.Ю.		24.02.23				
					<i>Вступ</i>			

використовується на підприємстві торгівлі. Це інтернет-платформа, що надає можливість підприємству вести торговельну діяльність через інтернет.

Предметом дослідження є архітектура програмного продукту (інтернет-платформа підприємства торгівлі).

Архітектура програмного комплексу відіграє ключову роль у створенні та оптимізації функціональності, продуктивності, масштабованості та безпеки інтернет-платформи.

Результатом дослідження є розроблена архітектура інтернет-платформи підприємства торгівлі, що задовольнятиме потреби підприємства та сприятиме його успішній роботі в інтернет-середовищі, за допомогою мов програмування *JavaScript*, *C#*, системою управління базами даних *MSSQL* та мовою розмітки гіпертексту *HTML*, *CSS* та фреймворком *ASP.NET Core MVC*.

Для успішного виконання завдання по розробленню архітектури інтернет-платформи підприємства торгівлі необхідно виконати наступні кроки:

1. Детально вивчити вимоги: проаналізувати вимоги підприємства торгівлі до інтернет-платформи з урахуванням потреб бізнесу.
2. Дослідити технології: ознайомитися з різними технологічними рішеннями, які можуть бути використані при розробці програмного комплексу, проаналізувати аналогічні інтернет платформи підприємств торгівлі.
3. Вибрати стратегію розробки.
4. Спроекувати архітектуру: розробити модель інтернет-платформи, визначити структуру, модулі та їх взаємозв'язки.
5. Розробити інтернет-платформу: розробити інтернет-платформу підприємства торгівлі з використанням *JavaScript*, *C#*, *ASP.NET Core MVC*, *HTML*, *CSS*. Зробити інтернет-платформу адаптивною під мобільні пристрої.

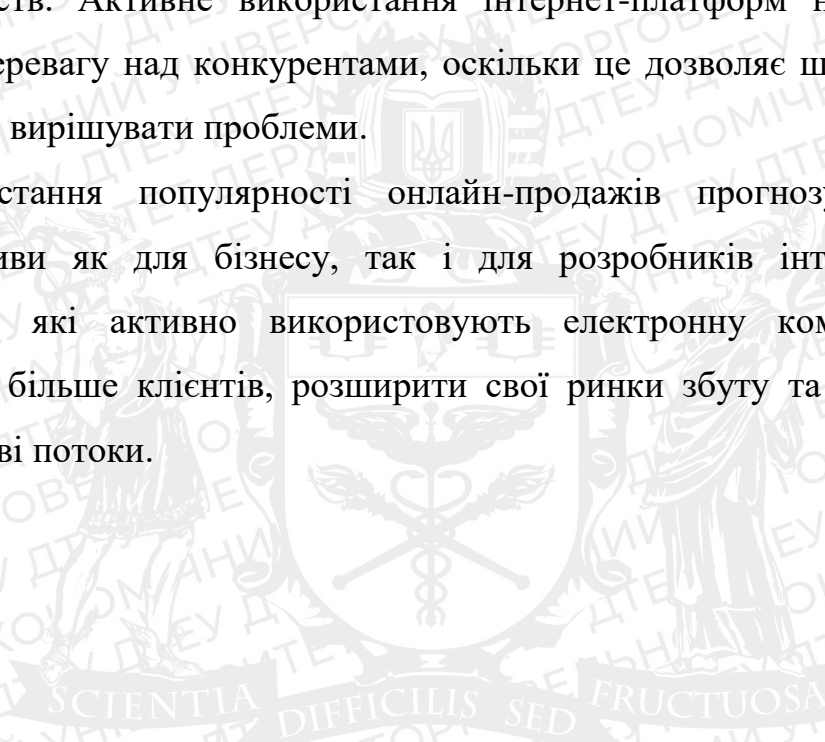
						Аркуш
					ДТЕУ 121 02-11.МР	6
Зм.	Аркуш	№ докум	Підпис	Дата		



6. Протестувати та вдосконалити: провести тестування інтернет-платформи, виявити та виправити помилки та недоліки. Вдосконалити архітектуру на основі отриманих результатів тестування та зворотного зв'язку.

Розробка архітектури програмного комплексу інтернет-платформи підприємства торгівлі є важливим кроком у розвитку сучасних торговельних підприємств. Активне використання інтернет-платформ надає компаніям значну перевагу над конкурентами, оскільки це дозволяє швидко реагувати на зміни і вирішувати проблеми.

Зростання популярності онлайн-продажів прогнозується і надає перспективи як для бізнесу, так і для розробників інтернет-платформ. Компанії, які активно використовують електронну комерцію, можуть залучити більше клієнтів, розширити свої ринки збуту та підвищити свої прибуткові потоки.



						Аркуш
					ДТЕУ 121 02-11.МР	6
Зм.	Аркуш	№ докум	Підпис	Дата		

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1. Актуальність предметної області

Актуальність предметної області, пов'язаної з розробкою архітектури програмного комплексу для інтернет-платформи підприємства торгівлі, є надзвичайно високою в сучасному світі. Розвиток електронної комерції та онлайн-торгівлі перетворюється на ключовий елемент успішного функціонування багатьох підприємств.

Онлайн-торгівля надає підприємствам унікальну можливість просувати свої товари та послуги через інтернет, залучати нових клієнтів та збільшувати обсяги продажів. За останні роки споживачі все більше переходять до онлайн-покупок, шукаючи зручності, доступності та широкого вибору. Це створює потужний попит на розробку та вдосконалення інтернет-платформ для торговельних підприємств.

Архітектура програмного комплексу є ключовим елементом успішної реалізації інтернет-платформи торговельного підприємства. Вона визначає структуру, взаємозв'язки та функціональні можливості системи.

Правильно спроектована архітектура дозволяє забезпечити швидку та надійну роботу платформи, масштабованість, безпеку даних та зручний користувацький досвід.

У зв'язку зі зростанням конкуренції в онлайн-торгівлі, необхідність постійного оновлення та адаптації до змін ринку, розробка архітектури програмного комплексу для інтернет-платформи стає важливим завданням

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 02-11.МР			
Зав. каф.		Криворучко О.В.		10.04.23	Архітектура програмного комплексу інтернет-платформи підприємства торгівлі	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		10.04.23		РІ	7	55
Гарант		Котенко Н.О.		10.04.23		Факультет інформаційних технологій 2м курс, 2 група		
Розробив		Мищенко в.Ю.		10.04.23				
					Аналіз предметної області			



для підприємств торгівлі. Вона дозволяє підприємствам торгівлі забезпечувати конкурентоспроможність, ефективність та інноваційність своєї діяльності, залучати нових клієнтів та забезпечувати задоволення потреб сучасного споживача.

Крім того, розробка архітектури програмного комплексу для інтернет-платформи підприємства торгівлі спрямована на вирішення таких актуальних завдань:

- Оптимізація торговельних процесів: впровадження ефективної архітектури дозволяє автоматизувати ключові бізнес-процеси, такі як управління запасами, обробка замовлень, доставка товарів, облік фінансів та інші. Це допомагає підприємствам економити час, зусилля та ресурси, забезпечуючи ефективну роботу всієї системи.

- Покращення користувацького досвіду: розробка зручного інтерфейсу та функціональних можливостей інтернет-платформи дозволяє забезпечити зручність та задоволення користувачів. Висока швидкість завантаження сторінок, зручний пошук товарів, персоналізований підхід та інші фактори створюють позитивний враження у клієнтів та збільшують їх лояльність до бренду.

- Забезпечення безпеки: захист персональних даних клієнтів та фінансових транзакцій є критичним аспектом в електронній комерції. Розробка архітектури, що включає надійні механізми шифрування, захист від шахрайства та зловживань, дозволяє забезпечити конфіденційність та безпеку інформації.

Інтеграція з іншими системами: сучасні підприємства торгівлі в практичності потребують інтеграції з іншими системами, такими як системи управління складом, обліку, логістики, електронних платежів та іншими.

Розробка архітектури програмного комплексу дозволяє створити зручний механізм обміну даними та взаємодії з іншими системами, що

						Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

сприяє автоматизації бізнес-процесів та підвищує ефективність діяльності підприємства.

- Аналітика та звітність: розроблена архітектура повинна передбачати збір, обробку та аналіз даних про продажі, поведінку клієнтів, популярність товарів та інші фактори. Це дозволяє здійснювати ефективний маркетинговий аналіз, прогнозування попиту, управління запасами та приймати обґрунтовані рішення з покращення діяльності підприємства.

Усі ці аспекти демонструють актуальність розробки архітектури програмного комплексу для інтернет-платформи підприємства торгівлі. Вона допомагає підприємствам ефективно використовувати потенціал електронної комерції, забезпечує конкурентну перевагу, покращення користувацького досвіду та забезпечує взаємодію з іншими системами. Розробка архітектури є необхідним етапом у розвитку сучасних торговельних підприємств і сприяє їх успішності на ринку.

## 1.2. Огляд наявних аналогів

При дослідженні аналогів інтернет-платформ, було відібрано чотири інтернет-платформи з продажу автотоварів :

- 1) exist.ua
- 2) E-shop автозапчастин
- 3) E-detail
- 4) Autopartner.in.ua

						Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



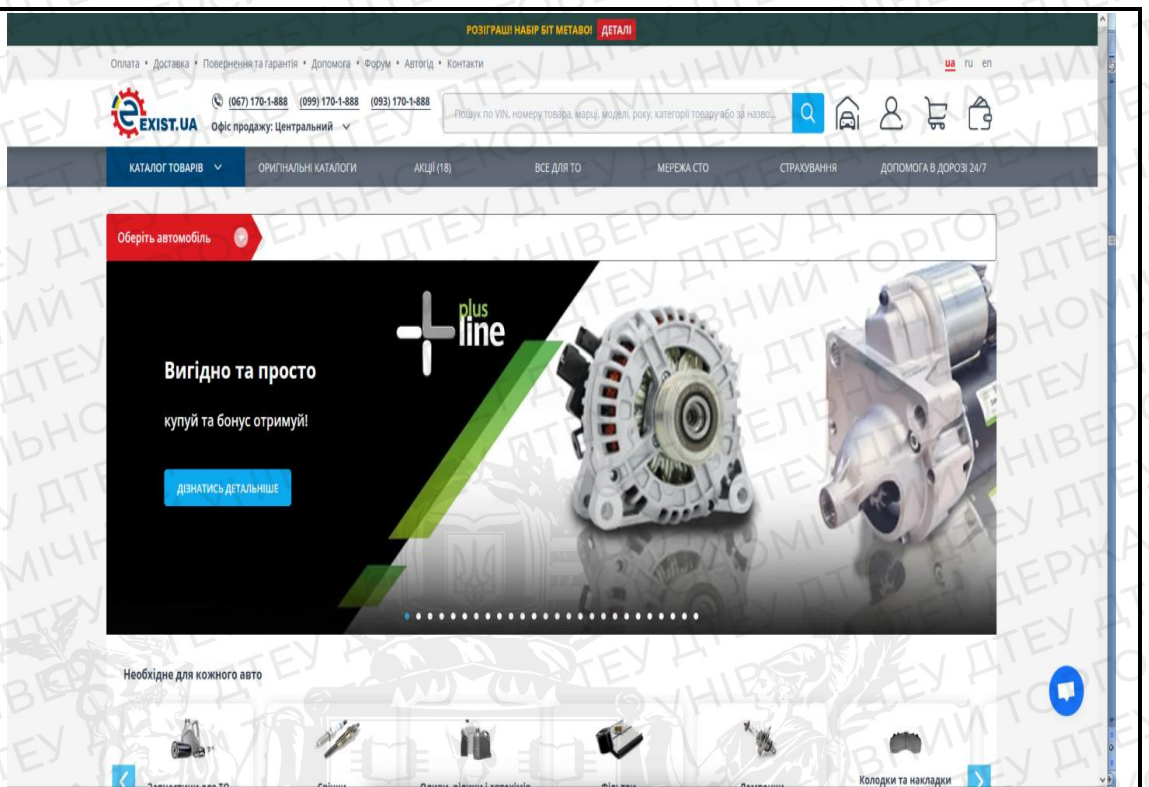


Рисунок. 1.1 Інтернет-платформа з продажу автотоварів «exist.ua»

Джерело: [1]



Рис. 1.2. Інтернет-платформа з продажу автотоварів «E-shop автозапчастини»

Джерело: [2]

					Аркуш
					10
Зм.	Аркуш	№ докум	Підпис	Дата	

ДТЕУ 121 02-11.МР

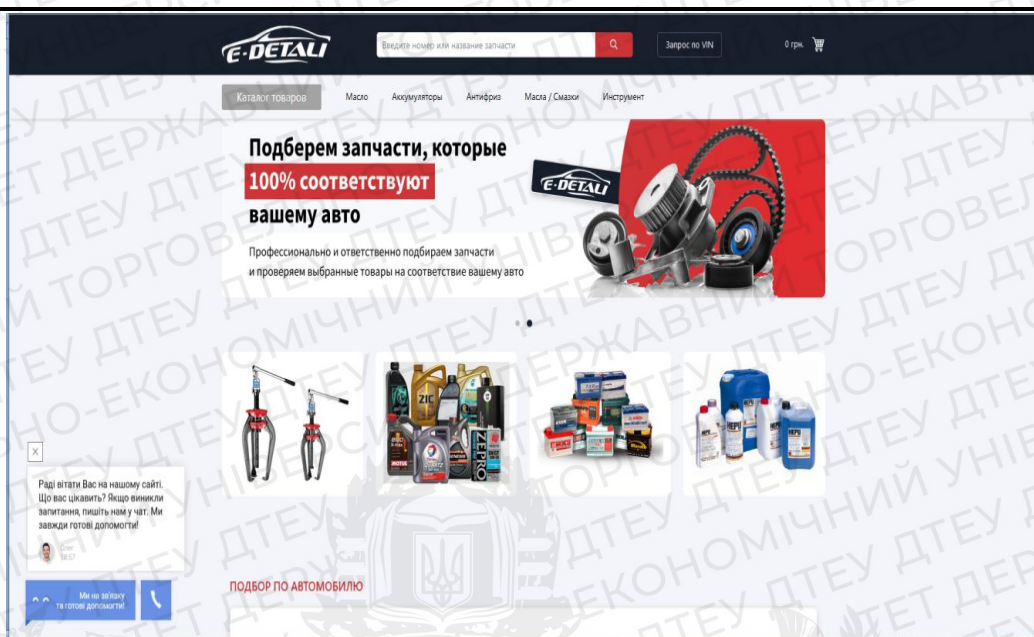


Рис. 1.3. Интернет-платформа з продажу автотоварів «E-detail»

Джерело: [3]

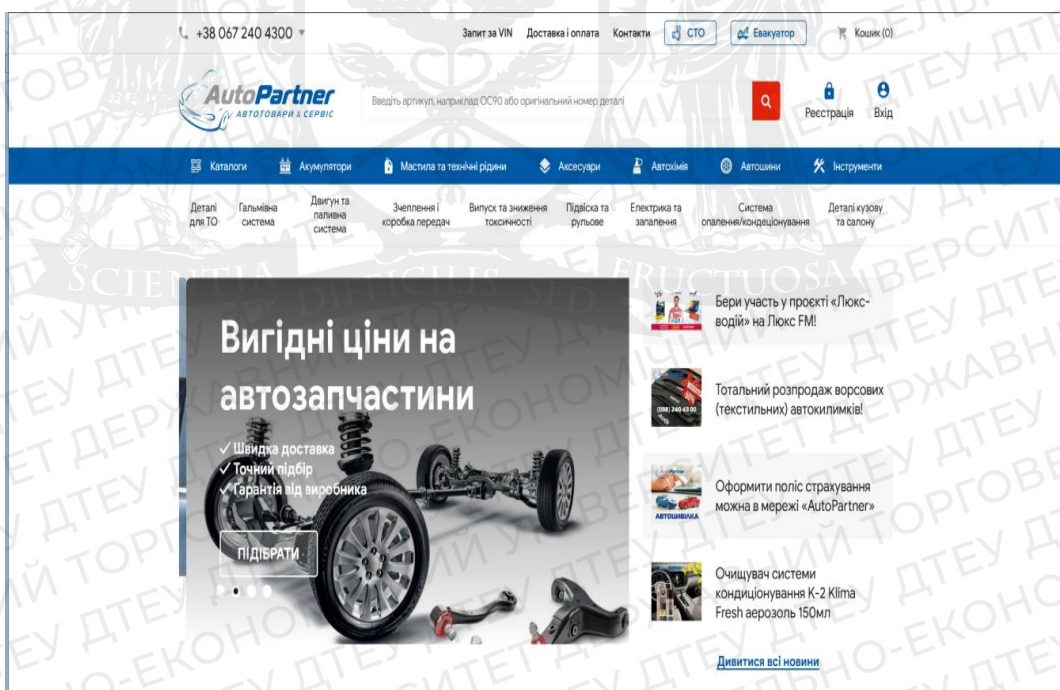


Рис. 1.4. Интернет-платформа з продажу автотоварів

«Autopartner.in.ua»

Джерело: [4]

					Аркуш
					ДТЕУ 121 02-11.МР
Зм.	Аркуш	№ докум	Підпис	Дата	11



Таблиця 1.1

## ПОРІВНЯННЯ ІНТЕРНЕТ-ПЛАТФОРМІВ\*

Критерій порівняння	Exist.ua	E-shop автозапчастини	E-detail	Autopartner.i n.ua
Кольорова гама	Сірий, синій та білі кольори	Синій, білий	Темно-синій, блакитний та білий кольори	Білий та синій кольори
Варіативність сортування товарів	За ціною, доставкою, наявністю, найменуванням та популярністю	За ціною, новизною, наявністю	-	За ціною, новизною, топом продаж, наявністю, популярністю, назвою
Наявність пошуку по товарах	+	+	+	(тільки за він кодом)
Наявність особового кабінету	+	+	-	+
Зворотній зв'язок	+	+	+	-
Наявність відгуку про товар	+	-	+	-
Наявність рейтингу	+	-	-	-

\*Джерело: згенеровано автором при аналізі [1-4]

						Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

Порівнявши конкурентні інтернет-платформи, було встановлено, що більшість інтернет-платформ надає можливість пошуку товарів, має варіативність сортування та функцію зворотнього зв'язку. З недоліків виявлено, що більшість конкурентів нехтує наявністю відгуків про товар та систему рейтингу.

Отже, розробка та впровадження всіх наведених функціональних можливостей на інтернет-платформі має ряд значущих переваг:

1. Конкурентна перевага: Інтернет-платформа буде мати перевагу завдяки повному спектру функціоналу, який задовольняє різноманітні потреби користувачів.

2. Кращий користувацький досвід: Забезпечення розширеного функціоналу, такого як різні способи сортування, пошук, особистий кабінет і відгуки, допомагає користувачам знаходити потрібні товари швидше і зручніше.

3. Збільшення лояльності клієнтів: Особистий кабінет, можливість залишати відгуки та отримувати зворотній зв'язок підвищують лояльність клієнтів і сприяють повторним покупкам.

Розробка всіх цих пунктів має значущий потенціал покращити позицію на ринку, залучити більше клієнтів і забезпечити їм кращий користувацький досвід, що відобразиться у зростанні обсягів продажів і задоволенні клієнтів.

### 1.3. Постановка задачі, визначення вимог та цілей

Розробка оптимальної архітектури програмного комплексу дозволить підприємству оптимізувати та автоматизувати багато процесів, що пов'язані з продажем товарів, управлінням запасами, обробкою замовлень. Інтернет-платформа дозволить підприємству створити зручне та легкодоступне середовище для клієнтів, де вони зможуть здійснювати покупки, переглядати товари, робити замовлення та спілкуватися з працівниками підприємства. Це

						Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



покращить взаємодію з клієнтами, забезпечить їхнє задоволення та збільшити лояльність.

Створення архітектури програмного забезпечення включає кілька етапів, які допомагають визначити структуру та функціональність системи. Основні етапи створення архітектури програмного комплексу для інтернет-платформи підприємства торгівлі включають наступні:

**Визначення вимог:** цей етап включає аналіз бізнес-вимог і потреб клієнтів, а також визначення функціональних та нефункціональних вимог до системи. Для успішного розроблення архітектури необхідно чітко зрозуміти, які задачі має виконувати програмний комплекс та які вимоги він повинен задовольняти.

**Проектування загальної архітектури:** на цьому етапі визначається загальна структура програмного комплексу. Розглядаються різні компоненти системи, їх взаємодія та організація. Зазвичай використовуються архітектурні шаблони та підходи, такі як клієнт-серверна архітектура, розподілена архітектура чи мікросервісна архітектура, для досягнення бажаної функціональності та шкальованості системи.

**Проектування компонентів:** на цьому етапі визначаються окремі компоненти системи та їх взаємодія. Розглядається функціональність кожного компонента, його інтерфейси та взаємодія з іншими компонентами. Компоненти можуть включати бази даних, сервери, модулі обробки даних, інтерфейси користувача та інші.

**Вибір технологій:** на цьому етапі визначаються технології та інструменти, які будуть використовуватися для реалізації архітектури. Вибір технологій залежить від потреб проекту, особливостей платформи та доступних ресурсів. Це можуть бути мови програмування, фреймворки, бази даних, протоколи взаємодії та інші.

**Проектування безпеки:** забезпечення безпеки є важливою частиною

						Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

архітектури програмного комплексу. На цьому етапі визначаються механізми захисту даних, автентифікації користувачів, контролю доступу та інші аспекти безпеки. Потрібно враховувати потенційні загрози та розробляти стратегії захисту.

Реалізація та тестування: після проектування архітектури програмного комплексу настає етап його реалізації. Розробники виконують програмування компонентів системи, інтегрують їх та проводять тестування для перевірки правильності реалізації функціональності та відповідності вимогам. На цьому етапі також проводяться різноманітні тести, включаючи модульні, інтеграційні та системні тести, для забезпечення якості розробленої системи.

Впровадження та супровід: після успішної реалізації та тестування архітектури програмного комплексу, він готовий до впровадження. На цьому етапі система встановлюється на платформі підприємства торгівлі, і відбувається її запуск та налагодження. Після впровадження системи проводиться супровід, що включає підтримку, моніторинг та вирішення можливих проблем, а також апгрейди та покращення архітектури в майбутньому.

Всі ці етапи створення архітектури програмного комплексу важливі для забезпечення ефективності, масштабованості, безпеки та надійності інтернет-платформи підприємства торгівлі. Професійний підхід до проектування та реалізації архітектури є вирішальним для успіху проекту та забезпечення конкурентоспроможності підприємства.

Цілі розробки архітектури програмного комплексу для інтернет-платформи підприємства торгівлі включають наступне:

- Розробка ефективної та масштабованої архітектури: одна з головних цілей полягає у створенні архітектурного рішення, яке буде ефективним, гнучким та легким у розширенні. Це дозволить підприємству торгівлі збільшувати свою присутність в Інтернеті та обслуговувати більшу

						Аркуш
					ДТЕУ 121 02-11.МР	15
Зм.	Аркуш	№ докум	Підпис	Дата		



кількість користувачів.

- **Забезпечення безпеки та захисту даних:** розробка архітектури повинна передбачати механізми безпеки, які забезпечують захист конфіденційності, цілісності та доступності даних. Це включає захист від несанкціонованого доступу, атак хакерів, витоку даних та інших загроз.

- **Оптимізація продуктивності та швидкості роботи:** важливим аспектом розробки архітектури є забезпечення високої продуктивності та швидкості роботи інтернет-платформи. Це включає оптимізацію запитів до бази даних, кешування даних, використання швидких алгоритмів та технологій, які дозволяють

- **Розширення функціональності:** однією з метою розробки архітектури програмного комплексу є створення платформи, яка має широкі можливості та функціональність. Це може включати можливість виконання онлайн-замовлень, оплати, керування інвентарем, аналітики продажів, зв'язок з клієнтами та багато іншого.

- **Забезпечення зручного інтерфейсу користувача:** важливим аспектом розробки архітектури є створення зручного та інтуїтивно зрозумілого інтерфейсу користувача. Це означає, що інтернет-платформа повинна мати привабливий дизайн, легку навігацію, зрозумілі форми та елементи управління, що спрощують взаємодію користувача з платформою.

- **Масштабованість та гнучкість:** архітектура програмного комплексу повинна бути масштабованою та гнучкою, щоб забезпечити можливість розширення функціональності, додавання нових модулів та пристосування до змінних потреб підприємства торгівлі. Це важливо для забезпечення довгострокового розвитку та успіху платформи.

- **Технічна підтримка та обслуговування:** розробка архітектури повинна передбачати механізми для технічної підтримки та обслуговування платформи. Це включає виявлення та виправлення помилок, моніторинг

						Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

роботи платформи, забезпечення безперебійної роботи та забезпечення оновлень та покращень.

Ці задачі спрямовані на створення ефективної, безпечної, функціональної та зручної в інтерфейсі інтернет-платформи.

Вимоги до інтернет-платформи підприємства торгівлі можуть бути різноманітними, залежно від конкретних потреб та характеристик підприємства. Однак, основними вимогами можуть бути:

**Функціональність:**

- Можливість створення та відображення каталогу товарів з детальними описами, зображеннями та характеристиками.
- Функціонал онлайн-замовлень та оплати товарів.
- Управління корзиною покупок та оформлення замовлення.
- Можливість реалізації різних способів доставки товарів та обробки замовлень.
- Функціонал аналітики продажів, звітності та статистики.

**Користувацький інтерфейс:**

- Зручний та інтуїтивно зрозумілий інтерфейс для користувачів, який дозволяє швидко здійснювати пошук, перегляд товарів та оформлення замовлень.
- Мобільна сумісність, щоб платформа була доступна на різних пристроях, включаючи смартфони та планшети.
- Легкість навігації та використання, зрозумілість структури та розташування елементів інтерфейсу.

**Безпека:**

- Захист персональних даних користувачів, включаючи особисту інформацію, платіжні дані та іншу конфіденційну інформацію.
- Застосування шифрування даних та захисту від несанкціонованого доступу до системи.

						Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



Масштабованість:

– Можливість розширення та адаптації платформи для впровадження нових функцій, модулів та інтеграції з іншими системами.

Підтримка клієнтів:

– Надання можливості зворотного зв'язку та підтримки користувачів через чати, електронну пошту або телефон.

– Наявність системи керування замовленнями та можливість стежити за статусом замовлень.

Адміністративні функції:

– Можливість керування товарами, категоріями, акціями та знижками через адміністративний інтерфейс.

– Забезпечення контролю доступу та прав адміністраторів до функціональних можливостей системи.

Доступність та надійність:

– Гарантований доступ до платформи з урахуванням вимог до часу роботи та мінімізація перебоїв у роботі.

#### 1.4 Проектування логічної структури

Проектування логічної структури - це важливий процес у розробці програмної системи, який визначає, як будуть організовані та взаємодіяти різні елементи та сторінки інтернет-платформи для досягнення конкретних цілей.

Логічна структура включає: сторінку «*Auto elit*» з товарами, яка має заголовок, який розташований в верхній частині інтернет-платформи та містить розіли «Головна», «Контакти», «Доставка», «Пошук», «Корзина», «Особистий кабінет»

«Категорія товарів» включає в себе «Диски», «Моторні оливи», «Сидіння», які в свою чергу містять власні під категорії товарів.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	18

Для взаємодії з товарами присутні такі засоби як «Фільтри», «Сортування», «Пошук за вибором».

Нижня частина інтернет-платформи містить основну інформацію яка присутня на інтернет-платформі в таких пунктах: «Контакти», «Навігація», «Додатково», «Слідкуйте за нами»

На рисунку 1.5 представлено логічну структуру інтернет-платформи з продажу автомобільних товарів.

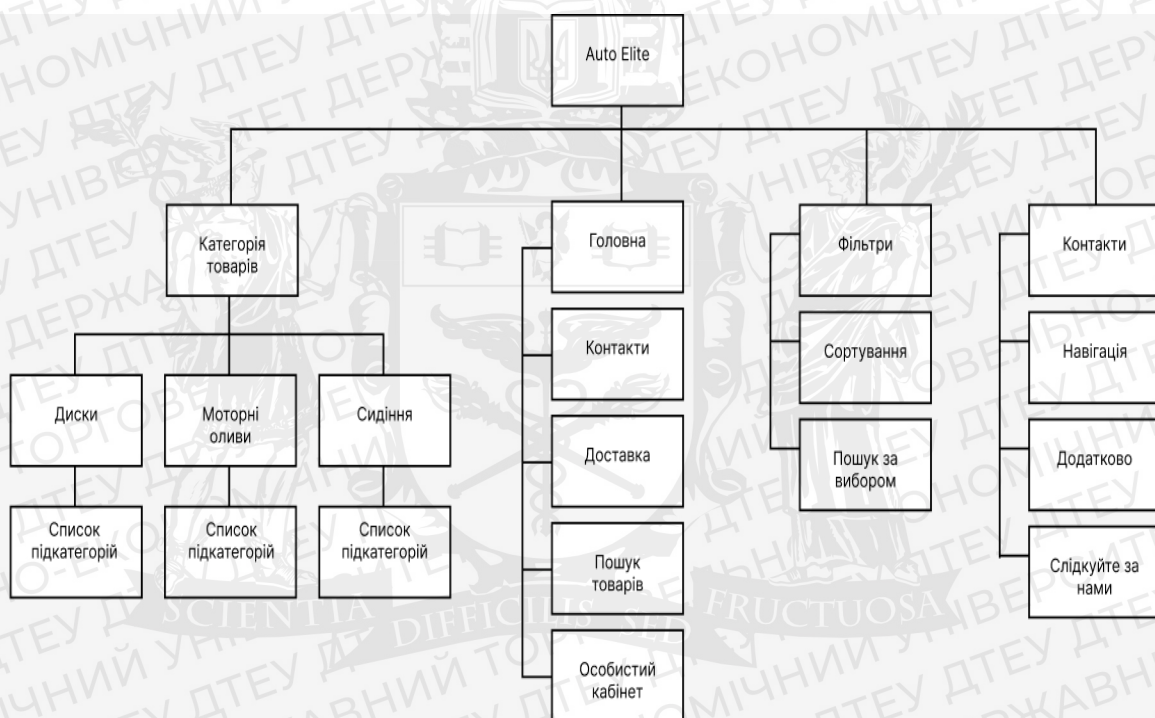


Рис. 1.5. Логічна структура інтернет-платформи з продажу автомобільних товарів.

Джерело: розроблено автором в середовищі Figma (скріншот з екрану)

### 1.5 Висновки до Розділу 1

В першому розділі було проведено аналіз актуальності теми, пов'язаної з розробкою архітектури програмного комплексу для інтернет-платформи підприємства торгівлі. Були розглянуті ключові аспекти, що обґрунтовують необхідність такої розробки, включаючи зростання електронної комерції,

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	
					19	



конкурентоспроможність підприємств у цьому сегменті, необхідність покращення користувацького досвіду, забезпечення безпеки даних, аналітику та звітність.

Проведено дослідження та огляд існуючих інтернет-платформи, які спеціалізуються на продажу авто товарів, які допомогли зрозуміти ринкову ситуацію та вимоги користувачів. Проектування логічної структури визначило, як різні елементи та сторінки інтернет-платформи будуть організовані та взаємодіяти для досягнення конкретних цілей.



						Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

## РОЗДІЛ 2

### МОДЕЛЮВАННЯ АРХІТЕКТУРИ ТА БАЗИ ДАНИХ

#### 2.1. Моделювання предметної області

Для побудови архітектури інтернет-платформи можна використовувати різні підходи.

UML (Unified Modeling Language) - це стандартизована мова моделювання, яка використовується для побудови програмного забезпечення. UML надає засоби для опису різних аспектів програмного забезпечення, включаючи структуру, поведінку та взаємодію між складовими. UML має багато різних діаграм, кожна з яких описує різний аспект програмного забезпечення [5].

UML дозволяє розробникам побудувати детальну модель системи, яка може включати в себе діаграми класів, діаграми послідовностей та діаграми діяльності, діаграма прецедентів, діаграма станів, діаграма компонентів, діаграма розгортання.

Узагальнюючи, побудова архітектури інтернет-платформи з використанням UML є важливим етапом розробки будь-якої системи. Вона дозволяє розробникам краще зрозуміти вимоги до системи, визначити її основні компоненти та забезпечити її ефективну та безпечну роботу.

Використання UML дозволяє зменшити ризики помилок та підвищити якість розробки, що є важливим для будь-якої компанії, що займається розробкою інтернет-платформ.

Для візуального представлення структури, поведінки та взаємодії системи або програмного забезпечення розглянемо діаграми UML на

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 02-11.МР</i>			
Зав. каф.		Криворучко О.В.		24.05.23	<i>Архітектура програмного комплексу інтернет- платформи підприємства торгівлі</i>	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		24.05.23		P2	10	55
Гарант		Котенко Н.О.		24.05.23		<i>Факультет інформаційних технологій 2м курс, 2 група</i>		
Розробив		Мищенко В.Ю.		24.05.23				
					<i>Моделювання архітектури та бази даних</i>			



прикладях.

Інформація про агентів, та опис випадків використання:

Агенти-користувачі: Клієнт – клієнт інтернет-платформи. Менеджер з продажу: користувач, що використовує функції адміністрування інтернет-платформи.

Агенти-зовнішні системи: База даних – база даних, яка зберігає інформацію.

В даній діаграмі наведені відношення між агентами та використаннями, що демонструє зв'язки між агентами та їх поведінкою в системі (подано на Рис.2.1).

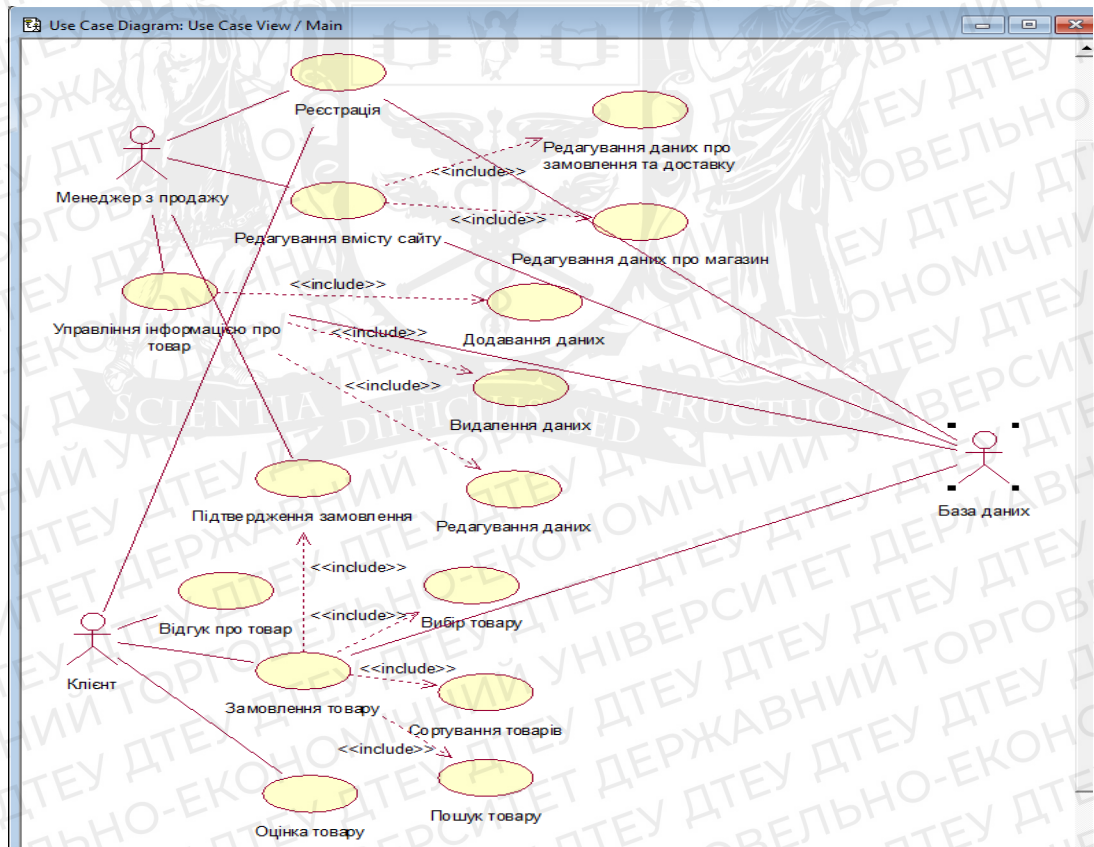


Рис. 2.1. UML-діаграма використання програмного продукту інтернет-платформи.

Джерело: розроблено автором в середовищі IBM Rational Rose Enterprise Edition (скріншот з екрану)

					Аркуш
					22
Зм.	Аркуш	№ докум	Підпис	Дата	

Для здійснення замовлення гостю не потрібно реєструватись, він може вказати особисті дані на етапі замовлення, якщо клієнт авторизований це спрощує процес замовлення, так як необхідно лише підтвердити адресу.

На рисунку 2.2 представлена діаграма декомпозиції варіанта використання "Оформлення замовлення". Спочатку авторизований клієнт додає або видаляє товари, за необхідності встановлює потрібну кількість одиниць кожного товару. Після цього користувач вказує свою адресу обирає тип доставки та спосіб оплати. Якщо тип оплати обрано через карту, то в клієнта є можливість здійснити оплату на інтернет-платформі. З свого боку менеджер перевіряє отримане замовлення та обробляє його.

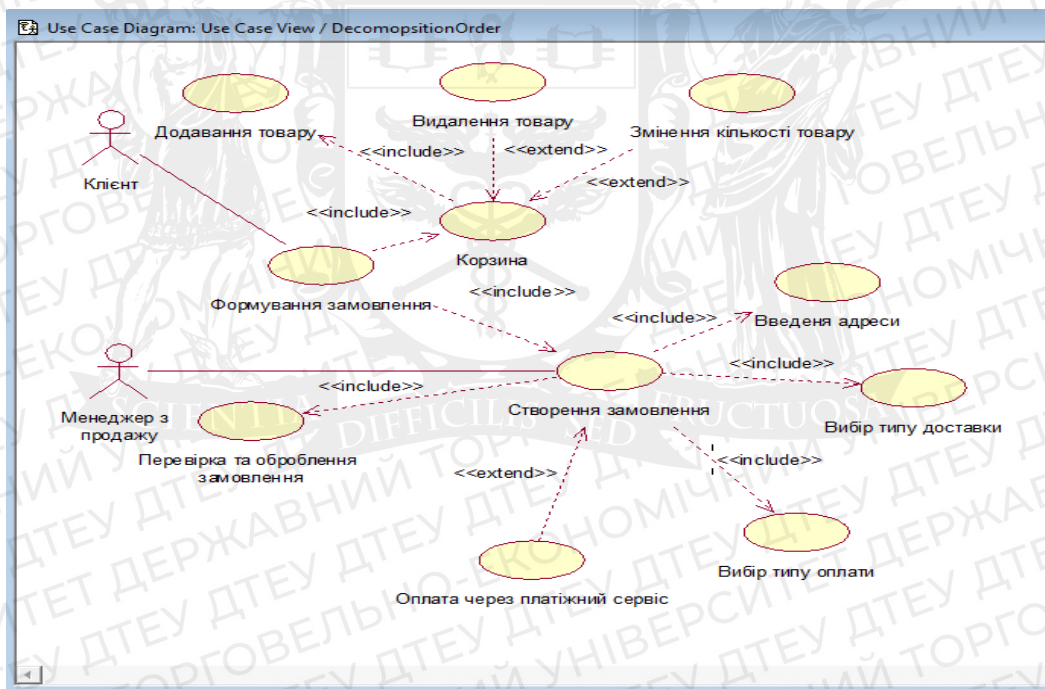


Рис. 2.2. UML-діаграма декомпозиції варіанта використання «Оформлення замовлення».

Джерело: розроблено автором в середовищі IBM Rational Rose Enterprise Edition (скріншот з екрану)

Діаграма класів є потужним інструментом для моделювання структури та взаємодії компонентів системи. На рисунку 2.3 представлені основні

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	
					23	



класи, такі як Product, Comment, ProductImage, ShopCartItem, OrderDetails, Order, Category, User, Advertisement і ContentSettings, а також визначені їхні взаємозв'язки.

Клас Product є центральним елементом системи, представляючи товари, які доступні для покупки. Класи Comment та ProductImage доповнюють інформацію про товари, дозволяючи користувачам залишати коментарі. ShopCartItem відображає процес додавання товарів до кошика, тоді як OrderDetails визначає деталі замовлення.

User відображає користувачів системи, і зв'язаний з класом Order. Category розподіляє товари, а Advertisement слідує за рекламними акціями. Налаштування вмісту (ContentSettings) дозволяють контролювати та налаштовувати різні аспекти системи.

Взаємозв'язки між цими класами визначають, як дані переміщуються та взаємодіють у межах системи. Завдяки діаграмі класів можна краще розробляти, розширювати та підтримувати інтернет-платформу, забезпечуючи її оптимальну функціональність і зручність для користувачів.

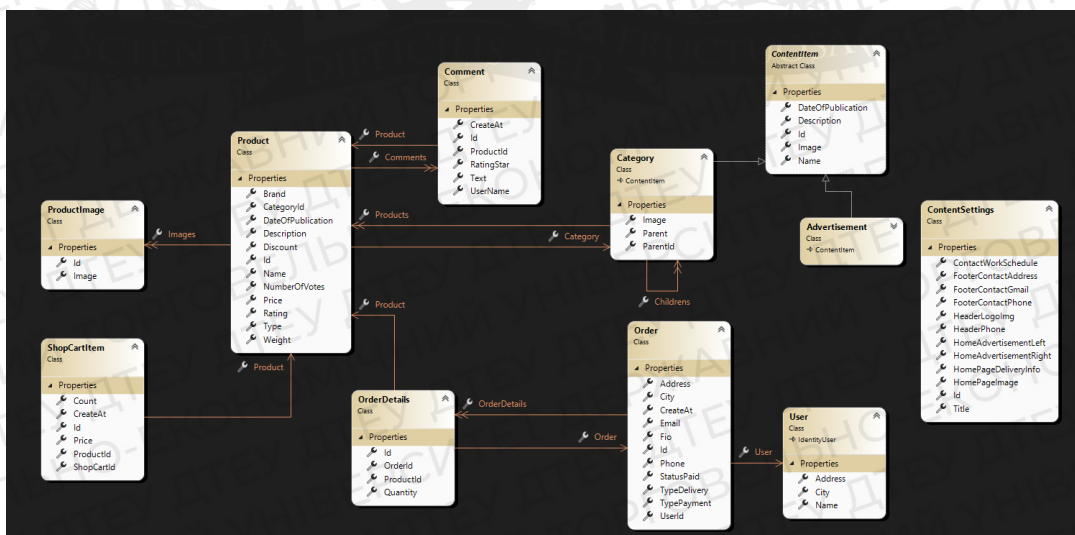


Рис. 2.3. Діаграма класів

Джерело: розроблено автором в середовищі Visual Studio 2022 (скріншот з екрану)

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	24

На діаграмі послідовності "Оформлення замовлення", яка зображена на рисунку 2.4, відображено процес оформлення замовлення, в якому беруть участь два агенти. Спочатку користувач переглядає вміст свого кошика та вносить необхідні зміни до товарів, які він хоче придбати, а потім переходить на сторінку оформлення замовлення. На цій сторінці він заповнює необхідну інформацію та підтверджує своє замовлення.

У той же час, менеджер переглядає замовлення, та за необхідності може його скасувати.

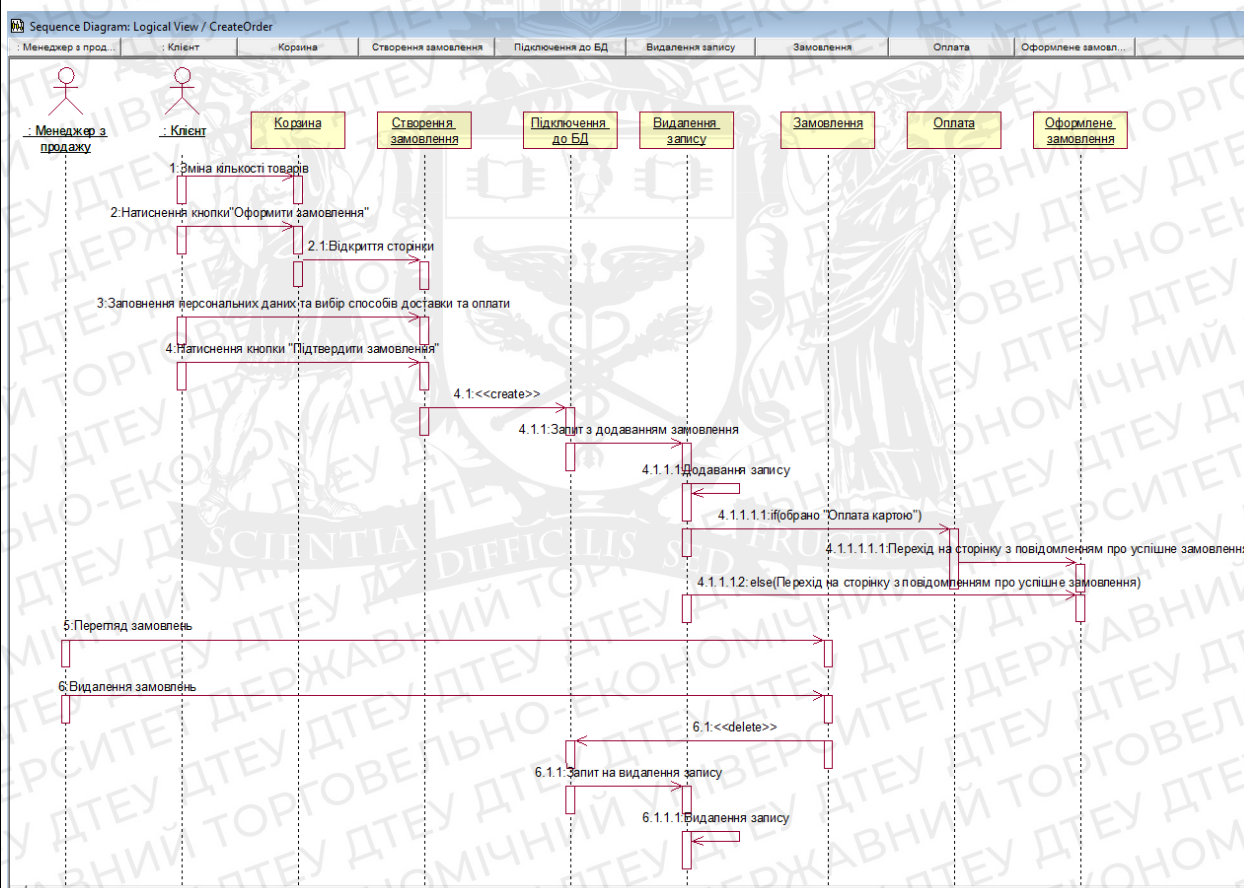


Рис. 2.4. UML-діаграма послідовності «Оформлення замовлення».

Джерело: розроблено автором в середовищі IBM Rational Rose Enterprise Edition (скріншот з екрану)

На рисунку 2.5 представлена діаграма стану для підсистеми "Користувач". Після того, як відвідувач інтернет-платформи ініціалізує свій

					Аркуш
					25
Зм.	Аркуш	№ докум	Підпис	Дата	





оформлення замовлення. Діаграма демонструє, що після того, як користувач відвідує головну сторінку інтернет-платформи, він має можливість перейти до перегляду товарів, використовуючи пошуковий рядок або перехід до каталогу. В будь-якому випадку система генерує запит до бази даних і надає користувачу список доступних товарів.

Зі сторінки перегляду товарів користувач може перейти до більш докладної інформації про товар. З цієї сторінки він може додати товар в кошик або повернутися до вибору інших товарів через каталог або пошук. Після того, як користувач заповнює всі необхідні дані для оформлення замовлення, відбувається перехід на сторінку, на якій відображається повідомлення про успішне замовлення з можливістю переходу на головну, або до замовлених товарів.

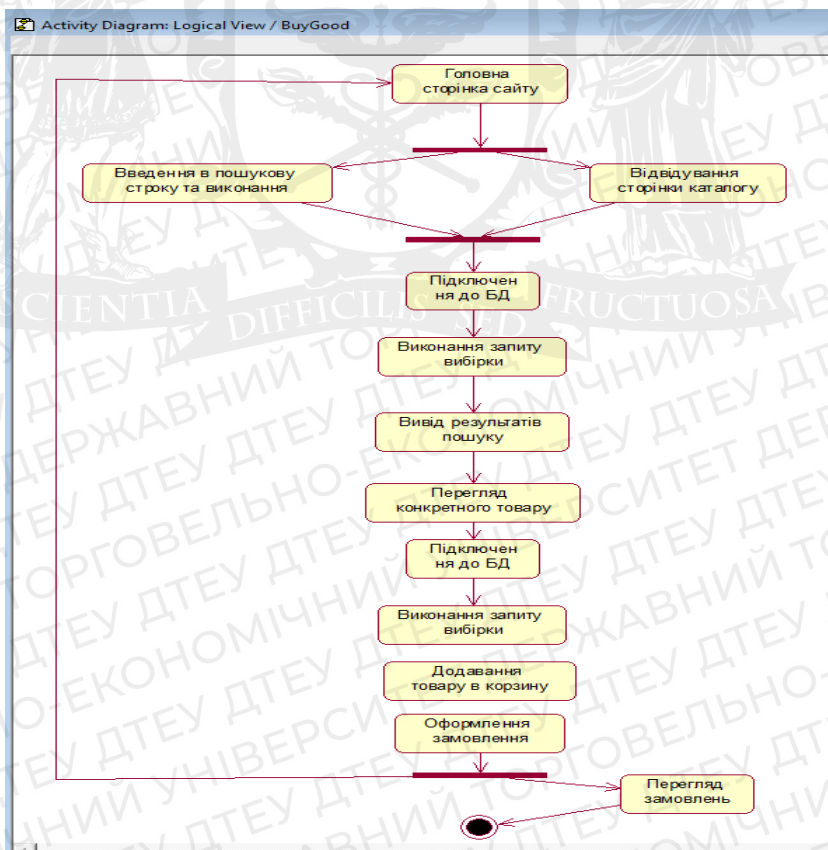


Рис. 2.6. UML-діаграма діяльності «Замовлення товару».

Джерело: розроблено автором в середовищі IBM Rational Rose Enterprise Edition (скріншот з екрану)

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	
						27



## 2.2. Огляд різних архітектурних стилів

На основі цих факторів можна розглянути такі архітектурні підходи:

**Мікросервісна архітектура:** Застосування мікросервісної архітектури передбачає розбиття системи на невеликі, незалежні компоненти (мікросервіси), які можуть бути розгорнуті та масштабовані окремо. Кожен мікросервіс відповідає за конкретну функціональність або модуль, і взаємодіє з іншими мікросервісами через API. Цей підхід сприяє гнучкості, швидкодії та легкості масштабування системи.

**Подійно-орієнтована (Event-driven) архітектура:** В цьому підході система реагує на події та виконує відповідні дії. Кожен компонент може публікувати та підписуватись на події, що забезпечує розподілену обробку даних та функціональності. Цей підхід підходить для систем з великою кількістю асинхронних операцій та підвищує масштабованість та реактивність системи.

**Схема відділення клієнта та сервера:** Цей підхід передбачає розділення логіки клієнтської та серверної частин системи. Клієнтська частина (фронтенд) відповідає за інтерфейс користувача та взаємодію з користувачем, а серверна частина (бекенд) обробляє запити, виконує бізнес-логіку та доступ до бази даних. Цей підхід забезпечує розділення обов'язків, спрощує розробку та підтримку системи.

**Схема "Тонкий клієнт - Товстий сервер":** В цій архітектурі більша частина логіки та обробки даних розташована на сервері, а клієнт (веб-браузер) виконує основну функцію відображення інтерфейсу та взаємодії з користувачем. Сервер забезпечує обробку запитів, виконання бізнес-логіки та доступ до бази даних. Цей підхід дозволяє зменшити навантаження на клієнтську сторону та спрощує розробку, але може вимагати більшої потужності та масштабування серверної інфраструктури.

						Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

Схема SOA (Service-Oriented Architecture): SOA передбачає побудову системи з набору незалежних сервісів, які виконують певні функції та можуть бути використані іншими компонентами системи. Кожен сервіс надає своє API для взаємодії з іншими сервісами. Цей підхід забезпечує модульність, гнучкість та перевикористання компонентів системи.

Схема SPA (Single-Page Application): SPA - це підхід, при якому весь інтерфейс та логіка клієнтської частини завантажуються один раз при початковому завантаженні сторінки, і подальша взаємодія з системою відбувається без перезавантаження сторінки. Цей підхід забезпечує більш плавну та швидку взаємодію з системою, але може вимагати більшої ресурсоемності клієнтського пристрою та складнішої розробки.

Схема MVC (Model-View-Controller): MVC - це популярний архітектурний підхід, який розділяє систему на три основні компоненти: модель (Model), вид (View), контролер (Controller).

Цей підхід сприяє розділенню відповідальностей між компонентами системи, полегшує розробку та підтримку коду, а також сприяє повторному використанню компонентів.

Схема MVP (Model-View-Presenter): MVP - це варіація архітектурного підходу MVC, яка акцентує увагу на презентаційній логіці.

Модель (Model), вид (View), презентер (Presenter).

MVP розділяє презентаційну логіку від бізнес-логіки, що полегшує тестування.

Схема MVVM (Model-View-ViewModel): MVVM - це архітектурний підхід, який спрямований на розділення логіки представлення даних від їх моделі та відображенням в інтерфейсі користувача.

Модель (Model), вигляд (View), ViewModel.

MVVM сприяє розділенню логіки представлення даних та забезпечує більшу гнучкість у розробці інтерфейсу користувача. Вона також підтримує

						Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



зв'язування даних між виглядом та ViewModel, що дозволяє автоматично оновлювати інтерфейс при зміні даних.

PWA (Progressive Web App): PWA - це підхід, який поєднує переваги інтернет-платформи та можливості мобільних додатків. PWA забезпечує можливість встановлення інтернет-платформи на пристрої користувача та доступ до нього навіть без підключення до Інтернету. Використовуючи сучасні веб-технології, PWA може забезпечити швидкий завантаження сторінок, взаємодію з пристроями та використання різноманітних функцій, таких як повідомлення, офлайн-режим та інші.

Додаткові архітектурні підходи до розробки інтернет-платформи включають:

Service-Oriented Architecture (SOA): SOA - це підхід, який розглядає систему як набір сервісів, які надають функціональні можливості та взаємодіють між собою за допомогою стандартизованих протоколів. Кожен сервіс може бути розгорнутий та масштабований окремо, що дозволяє системі бути гнучкою та легко розширюватись.

Layered Architecture: Цей підхід передбачає розділення системи на шари (логічні рівні), де кожен шар виконує певні функції та має свої обмеження. Наприклад, можуть бути визначені шари для представлення даних, бізнес-логіки та доступу до даних. Цей підхід спрощує розробку, тестування та підтримку системи, оскільки кожен шар може бути розглянутий окремо [6].

При виборі архітектурного підходу до розробки інтернет-платформи важливо враховувати вимоги до системи, масштаб проекту, доступні ресурси, команду розробників та інші фактори, щоб забезпечити оптимальну архітектуру.

						Аркуш
						30
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

### 2.3. Вибір оптимальної архітектури для підприємства торгівлі

Архітектурний підхід MVC (Model-View-Controller) є популярним та ефективним в багатьох веб-розробках. Ось деякі переваги використання архітектури MVC:

**Розділення відповідальностей:** MVC дозволяє чітко розділити логіку додатку на три компоненти - модель, представлення та контролер. Це полегшує управління кодом, розуміння логіки програми та підтримку кодової бази.

**Модульність:** Компоненти MVC (модель, представлення та контролер) можуть бути розроблені та підтримувані окремо. Це сприяє модульності, що полегшує розширення, тестування та зміни в системі без впливу на інші компоненти.

**Гнучкість у відображенні даних:** Модель та контролер не залежать від конкретного представлення даних, що дозволяє легко змінювати спосіб відображення, наприклад, змінювати веб-інтерфейс, додавати підтримку мобільних пристроїв тощо.

**Підтримка багаторазового використання коду:** Завдяки розділенню на модель та контролер, можна багаторазово використовувати код для обробки та маніпулювання даними, незалежно від конкретного представлення.

**Легше тестування:** Кожен компонент (модель, представлення та контролер) може бути тестований окремо, що полегшує виявлення та виправлення помилок.

Схему архітектурного підходу MVC представлено на рисунку 2.7.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	31



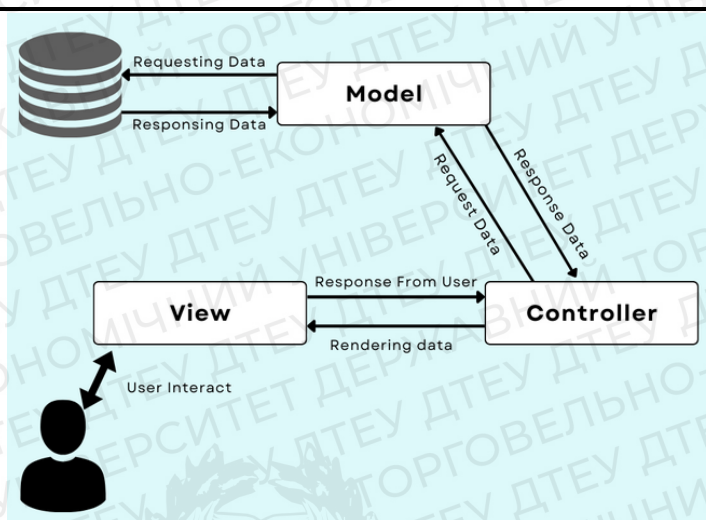


Рис. 2.7. Схема архітектурного підходу MVC

Джерело: [7]

Враховуючи ці переваги, архітектурний підхід MVC є гнучким, модульним та добре підходить для розробки інтернет-платформи, де чітко розділена логіка, керування та представлення даних є важливими аспектами.

## 2.4. Опис структури бази даних

Головне завдання бази даних полягає у забезпеченні надійного та безпечного зберігання даних, а також у наданні зручного доступу до цих даних для їх обробки та використання. Інформація в базі даних організована у вигляді структурованих таблиць, які мають рядки та стовпці. Це дозволяє ефективно управляти та обробляти великі обсяги даних.

Основна мета та переваги використання баз даних включають:

- Збереження даних: Бази даних дозволяють тривалий зберігати різноманітні види інформації, від текстових документів до мультимедійних файлів.
- Організація та структурування даних: Дані організовані у логічні таблиці та пов'язані між собою, що полегшує їхню структурування та

						Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

організацію.

– Ефективне керування даними: Бази даних дозволяють виконувати різні операції з даними, такі як додавання, видалення, зміна та пошук, забезпечуючи ефективний доступ до інформації.

– Безпека даних: Вони надають можливість обмежувати доступ до даних і встановлювати правила безпеки, щоб захистити конфіденційну інформацію.

– Забезпечення цілісності даних: Бази даних гарантують, що дані залишаються цілими і консистентними під час їхньої обробки та зміни.

Рисунок 2.8 демонструє таблиці які були утворені в результаті проєктування та розробки БД.

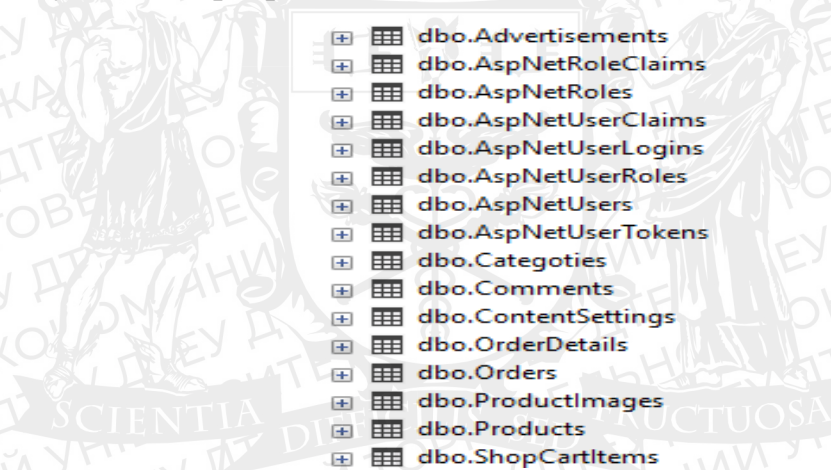


Рис. 2.8. Створені таблиці

*Джерело: розроблено автором в середовищі Microsoft SQL Server Management Studio(скріншот з екрану)*

При створенні бази даних використовувався підхід Code First Approach.

Підхід Code First в EF Core дозволяє визначати моделі класів, які представляють таблиці в БД, а також відношення між цими таблицями, використовуючи атрибути або методи Fluent API.

EF Core автоматично створює БД на основі визначених моделей, включаючи створення таблиць, відношень між ними та відповідних стовпців

						Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



згідно з визначеними правилами. Крім того, EF Core автоматично генерує SQL-скрипти для створення, зміни або видалення структури БД згідно зі змінами в моделях.

На рисунку 2.9 продемонстровано підхід Code First Approach.

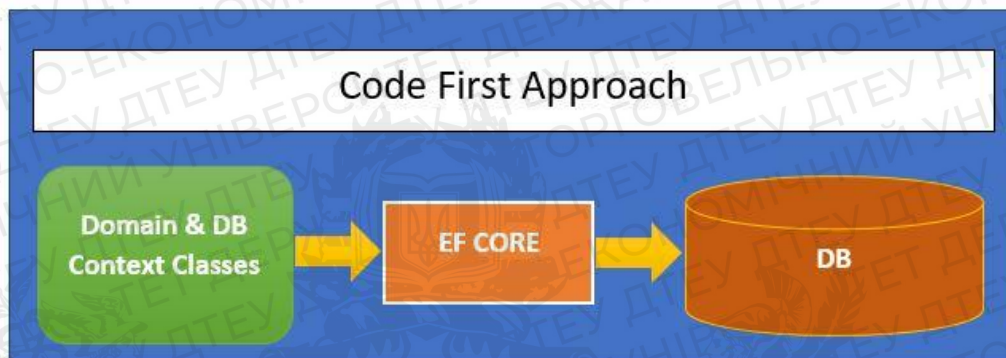


Рис..2.9. Схема Code First Approach

Джерело: [8]

Основні переваги підходу Code First в EF Core включають:

- Швидкий розвиток додатків: розробники можуть фокусуватися на визначенні моделей та відношень між ними, використовуючи рівень абстракції класів, замість витрати часу на написання SQL-скриптів.
- Легка міграція: EF Core надає механізми міграції, що дозволяють змінювати схему

## 2.5 Висновки до Розділу 2

В розділі було проведено моделювання предметної області інтернет-платформи для підприємства торгівлі з використанням UML. Було створено діаграми UML, які дозволили візуалізувати структуру, поведінку та взаємодію різних компонентів системи. Створено діаграму використання, де ідентифікувалися різні категорії агентів, такі як клієнти та менеджери з продажу, і їх взаємодія з системою. Ця діаграма дозволила чітко визначити

						Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

ролі та відповідальності кожного агента в системі. Було відображено взаємодію зовнішньої системи (бази даних) з системою інтернет-платформи. Це допомогло зрозуміти, як дані обмінюються між системою та базою даних. Діаграма декомпозиції варіанта використання продемонструвала процес оформлення замовлення, а діаграма класів продемонструвала структуру системи та стосунки між класами, що використовуються для моделювання та управління інформацією. Діаграма послідовності надала візуальне подання процесу оформлення замовлення та показала послідовність дій і взаємодію різних складових системи під час цього процесу. Діаграма стану підсистеми користувача відобразила різні стани та можливі переходи підсистеми під час взаємодії з системою, що допомогло зрозуміти і змодельовати її поведінку. Діаграма діяльності продемонструвала послідовність дій та процеси, пов'язані з процесом оформлення та замовлення товарів.

У результаті моделювання предметної області з використанням UML було створено базовий каркас для подальшої розробки архітектури програмного комплексу для інтернет-платформи підприємства торгівлі. Цей каркас визначив структуру системи та зв'язки між її складовими, що є важливим етапом у процесі розробки.

Також були розглянуті різні архітектурні стилі, такі як подійно-орієнтована архітектура, мікросервісна архітектура, сервер-клієнт та інші. Вибір оптимальної архітектури враховує специфіку підприємства торгівлі, його потреби в масштабованості, підтримці, безпеці та інших аспектах тож в даному варіанті була обрана MVC.

В цьому розділі надано опис структури бази даних та підходу Code First Approach у контексті розробки. Підход Code First є потужним інструментом для створення та керування базами даних, дозволяючи розробникам зосередитися на логіці додатку та спрощуючи роботу з базами даних.

						Аркуш
						35
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



## РОЗДІЛ 3

### ЗАСОБИ РЕАЛІЗАЦІЇ

#### 3.1. Інструменти для розробки

Середовище розробки **Visual Studio 2022**

Visual Studio 2022 є однією з найпопулярніших та потужних інтегрованих середовищ розробки (IDE), яка використовується для створення різноманітних програмних рішень. Вона надає розробникам широкий набір інструментів та функцій, які сприяють продуктивності, якості та ефективності розробки програмного забезпечення.

Visual Studio 2022 підтримує широкий спектр мов програмування, таких як C#, C++, Visual Basic, F#, JavaScript, Python та багато інших.

Мови програмування та платформи:

**Мова програмування C# (C Sharp)** є однією з найпопулярніших мов програмування в сфері розробки програмного забезпечення. Розроблена компанією Microsoft, C# поєднує в собі елементи об'єктно-орієнтованого програмування, що робить її потужним інструментом для розробки різноманітних додатків.

Основною особливістю мови C# є об'єктно-орієнтоване програмування: C# підтримує повну парадигму об'єктно-орієнтованого програмування (ООП), що дозволяє розробникам створювати класи, об'єкти, успадкування, поліморфізм та інші концепції ООП. Це сприяє створенню модульного, легко зрозумілого і підтримуваного коду [9].

**Платформа .NET:** C# є основною мовою програмування для

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 02-11.МР			
Зав. каф.		Криворучко О.В.		06.09.23	Архітектура програмного комплексу інтернет-платформи підприємства торгівлі	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		06.09.23		РЗ	36	55
Гарант		Котенко Н.О.		06.09.23		Факультет інформаційних технологій 2м курс, 2 група		
Розробив		Мищенко В.Ю.		06.09.23				
					Засоби реалізації			

платформи .NET, розробленої компанією Microsoft. Платформа .NET надає розширені можливості для розробки програмного забезпечення, включаючи роботу зі збіркою класів .NET Framework, побудову інтернет-платформ, роботу з базами даних, розробку мобільних додатків та інше [10].

### **Фреймворк ASP.NET Core MVC**

ASP.NET Core MVC (Model-View-Controller) є фреймворком для розробки інтернет-платформ на платформі ASP.NET Core. Він пропонує розділення додатку на логічні компоненти: модель (Model), представлення (View) та контролер (Controller), що сприяє покращенню модульності, перевикористання коду та підтримує принцип розділення обов'язків [11].

### **Технологія доступу до баз даних Entity Framework Core**

Entity Framework Core є розширенням Entity Framework, яке було розроблене для платформи .NET Core. Він забезпечує швидку та ефективну роботу з реляційними базами даних, дозволяючи розробникам працювати з даними в термінах об'єктів. З EF Core можна взаємодіяти з базою даних без прямого використання SQL-запитів, а замість цього використовувати LINQ (Language Integrated Query) - мовний запитовий механізм, що дозволяє писати запити до даних у структурованому та типобезпечному способі.

Переваги використання EF Core включають зручність взаємодії з базою даних за допомогою LINQ, автоматичний мапінг об'єктів на таблиці бази даних, підтримку міграцій для змін структури бази даних та підтримку різних провайдерів баз даних. Ці переваги роблять EF Core вигідним вибором для розробки програмного комплексу для підприємства торгівлі, оскільки дозволяють зосередитися на розробці функціональності, забезпечуючи зручний доступ до даних та ефективне управління базою даних [12].

### **Мова гіпертекстової розмітки HTML**

HTML (HyperText Markup Language) є стандартною мовою розмітки для створення веб-сторінок. Вона використовується для опису структури та

						Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



семантики вмісту веб-сторінки, такого як заголовки, абзаци, списки, таблиці та багато іншого. HTML використовує теги для визначення різних елементів сторінки та їх атрибутів для надання додаткових властивостей [13].

### **CSS (Cascading Style Sheets)**

У світі веб-розробки CSS (Cascading Style Sheets) є необхідним інструментом для стилізації та оформлення веб-сторінок. Він дозволяє розробникам контролювати вигляд і відображення елементів HTML, забезпечуючи красивий та привабливий зовнішній вигляд для інтернет-платформи [14].

### **MSSQL Server**

(Microsoft SQL Server) - це реляційна система управління базами даних (СУБД), розроблена компанією Microsoft. Вона надає потужні можливості для зберігання, управління та маніпулювання даними відповідно до принципів реляційної моделі [15].

### **JavaScript**

JavaScript - це високорівнева, об'єктно-орієнтована мова програмування, яка використовується для розробки інтернет-платформ. Вона була створена для надання можливостей взаємодії з користувачем на вебсторінках [16].

## **3.2. Опис інтерфейсу програми**

Після відкриття інтернет-платформи користувачу відображається сторінка з інформацією про компанію та можливістю відразу визначитись з товаром (рисунок 3.1).

						Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

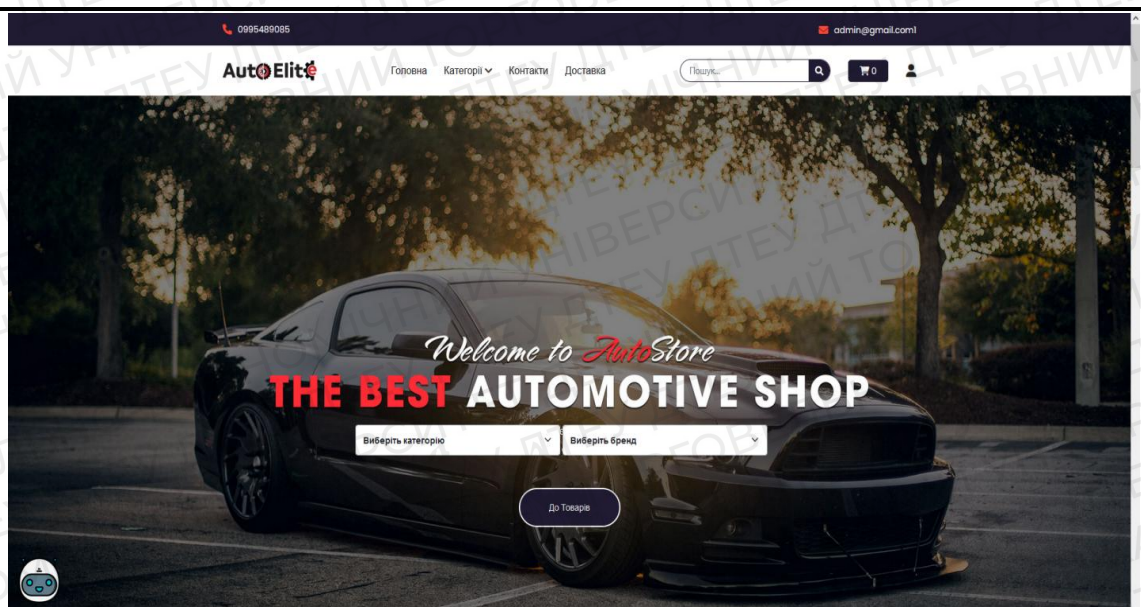


Рис. 3.1. Сторінка входу до інтернет-платформи

Джерело: розроблено автором, скріншот з екрану

Нижче знаходиться основана інформація про товари, їх доставка та акції. Також відображаються блоки товарів з цікавими пропозиціями, та блок з усіма категоріями які представлені на інтернет-платформі.

Щоб перейти на головну сторінку, користувач може вибрати пункт «Головна», або виконати вибір за категорією та брендом і натиснути «До товарів», після чого відобразиться сторінка з товарами (рисунок 3.2)

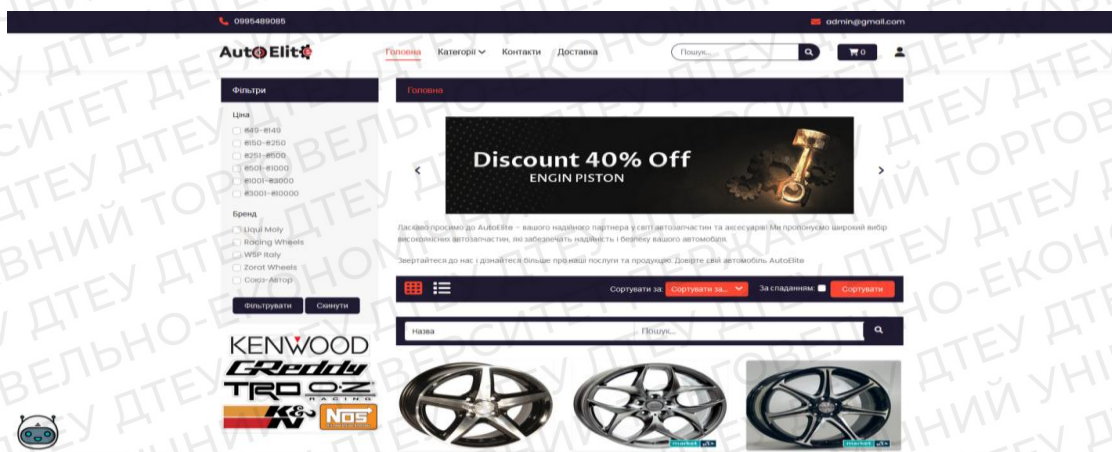


Рис. 3.2. Головна сторінка інтернет-платформи

Джерело: розроблено автором, скріншот з екрану

					Аркуш
					39
Зм.	Аркуш	№ докум	Підпис	Дата	

ДТЕУ 121 02-11.МР



На сторінці входу, або на головній сторінці, користувач може зареєструватись, або авторизуватись, відвідати інші сторінки, використати пошук за потрібним товаром, обрати товар додавши його в корзину та скористуватись чат-ботом для шручнійшої навігації по інтернет-платформи.

Щоб використати чат-бота, необхідно натиснути на робота в нижньому лівому куті екрану, після чого відобразиться меню, де можна обрати пункти, які можуть зацікавити користувача (рисунк 3.3).

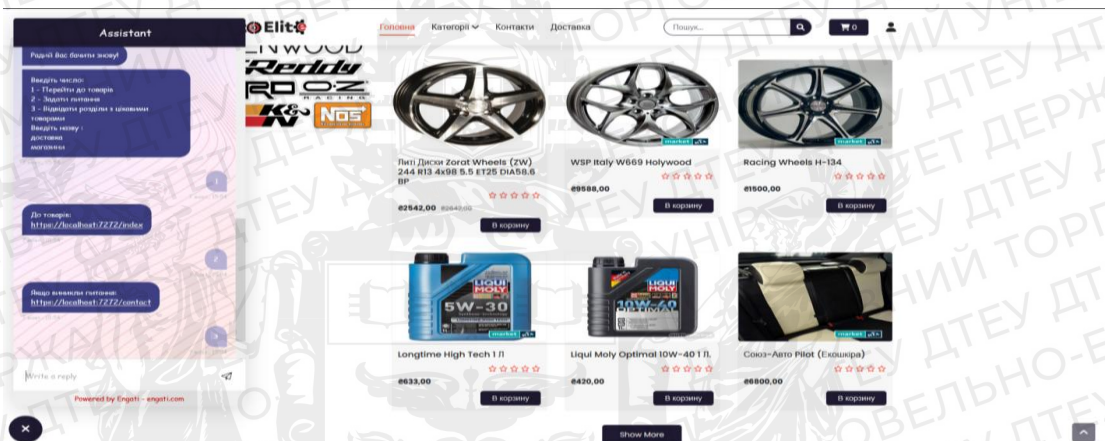


Рис. 3.3. Чат-бот інтернет-платформи.

*Джерело: розроблено автором, скріншот з екрану*

### 3.3. Основні функції інтерфейсу користувача

У цьому розділі розглянуто основні функції інтерфейсу користувача інтернет-платформи інтернет-платформи, таких як зареєстровані та незареєстровані. Незареєстрові користувачі мають обмежений функціонал, особливо в контексті швидкості замовлення товарів. Це обумовлено тим, що перед кожним оформленням замовлення незареєстровим користувачам необхідно вводити особисті відомості. У зареєстрованих користувачів відомості про них зберігаються в особистому кабінеті, що дозволяє автоматично використовувати їх під час замовлення товару. Також зареєстрований користувач має можливість відслідковувати історію

						Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

замовлень, та переглядати їх статус, що може допомогти в майбутніх замовленнях.

Користувач має можливість зареєструватися на інтернет-платформі, або увійти вже в створений кабінет, пройшовши авторизацію. Для цього потрібно натиснути на іконку в правому верхньому кутку де відкриється меню з можливістю реєстрації, або авторизації. Для реєстрації необхідно заповнити особисті відомості, такі як ім'я, електронна пошта, мобільний телефон, пароль, підтвердження паролю та *Google* капчу. Та натиснути «Зареєструватися» (рисунок 3.4).

Рис. 3.4. Сторінка реєстрації

*Джерело: розроблено автором, скріншот з екрану*

Наступним етапом, клієнту прийде на вказану електронну пошту повідомлення від магазину, де буде комбінація коду для підтвердження реєстрації (рисунок 3.5).

						Аркуш
					ДТЕУ 121 02-11.МР	41
Зм.	Аркуш	№ докум	Підпис	Дата		



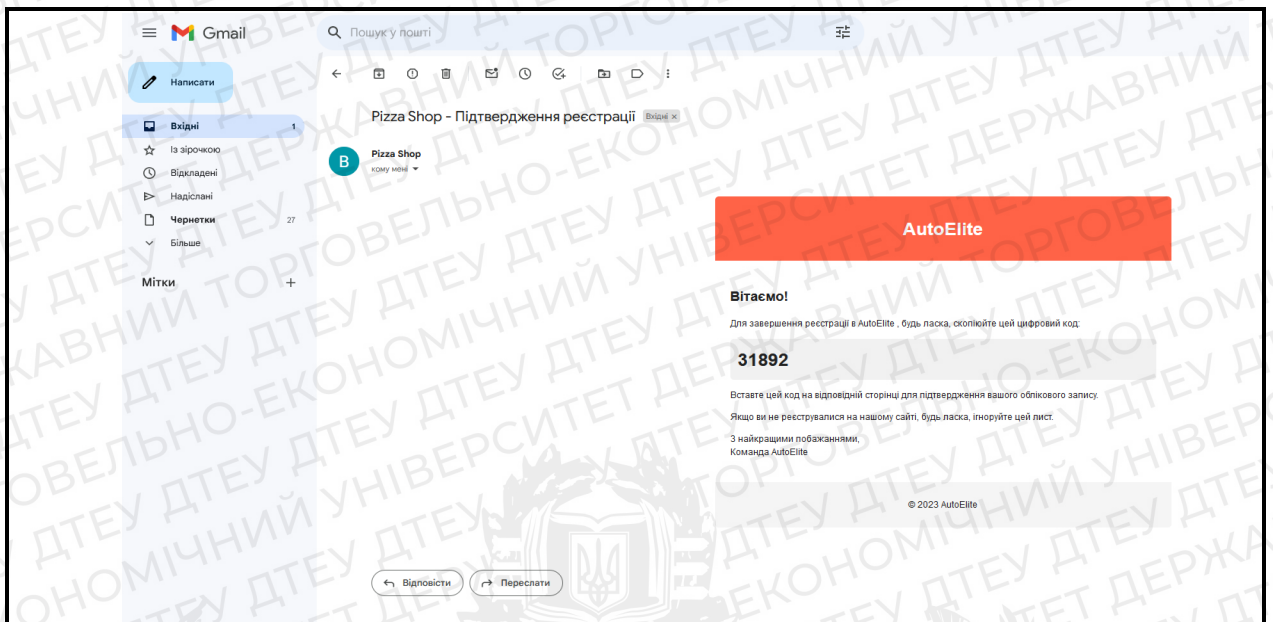


Рис. 3.5. Лист з кодом підтвердження реєстрації  
*Джерело: розроблено автором, скріншот з екрану*

Після вводу коретного коду користувач опиниться на головній сторінці з вже аунтефікованим акаунтом. Якщо всі необхідні поля на формі заповнені коректно, користувач буде зареєстрований, а потім відомості користувача будуть додані до бази даних. Якщо користувач вже має власний акаунт, йому достатньо пройти авторизацію. Для цього необхідно заповнити поля електронної пошти та пароллю коректними даними (рисунок 3.6).

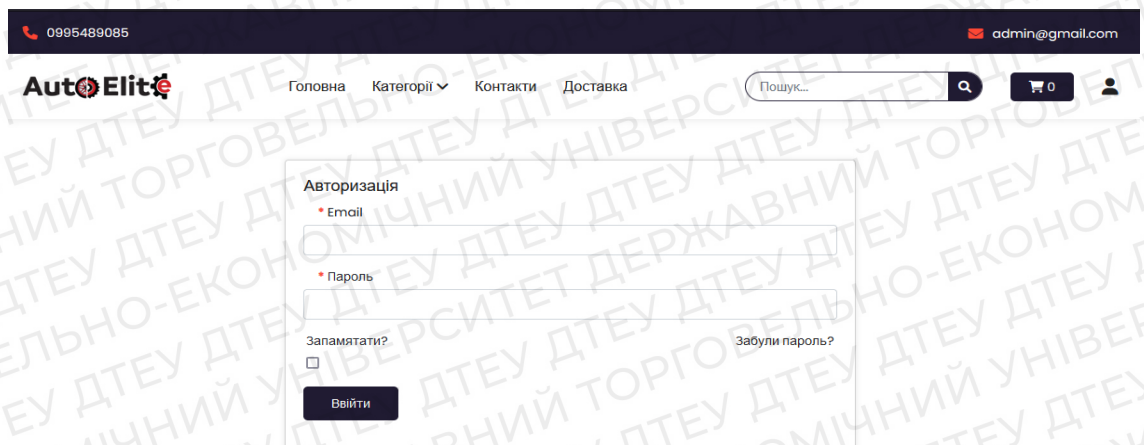


Рис. 3.6. Сторінка авторизації  
*Джерело: розроблено автором, скріншот з екрану*

						Аркуш
					ДТЕУ 121 02-11.МР	42
Зм.	Аркуш	№ докум	Підпис	Дата		

Після входу в особистий кабінет користувач має можливість редагувати, додавати та вилучати свої особисті дані. Для здійснення цих змін необхідно внести необхідні відомості у відповідні поля і натиснути кнопку "Змінити". Щоб полегшити реєстрацію поля з містом і адресою доставки було розміщено в особистому кабінеті для заповнення.

Також користувач може переглядати історію замовлень, їх статус, а за необхідності змінити пароль (рисунок 3.7).

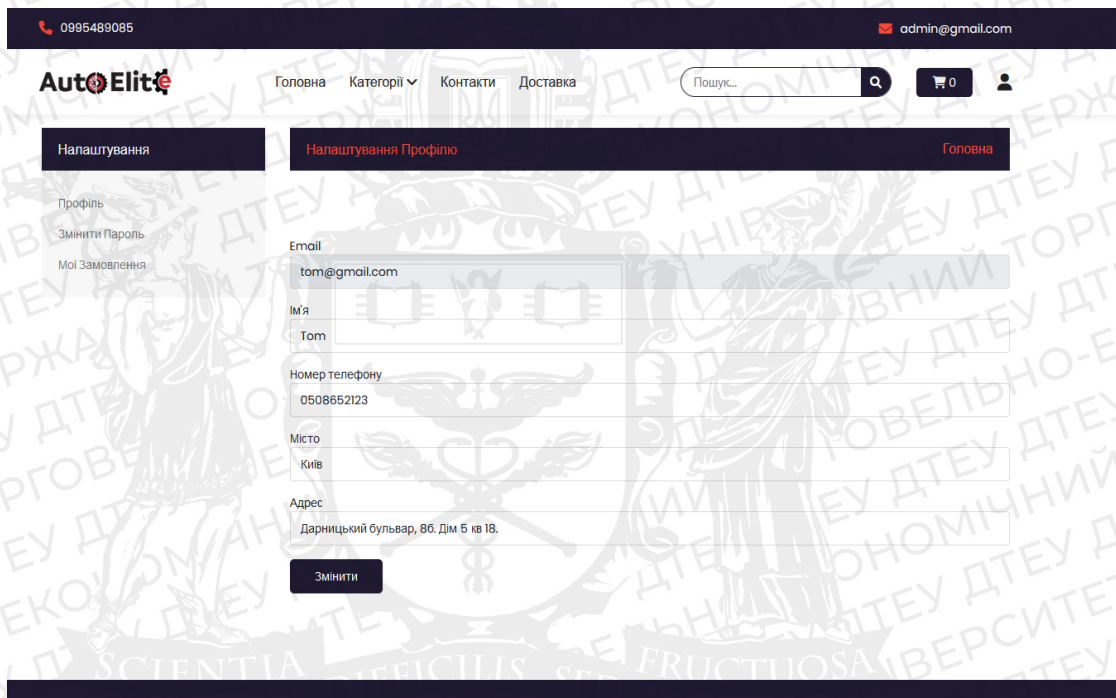


Рис. 3.7. Сторінка особистого кабінету

*Джерело: розроблено автором, скріншот з екрану*

Процедура придбання товарів розпочинається з переходу до каталогу товарів. Для цього потрібно вибрати опцію "Головна" з меню, яке розташоване у верхній частині інтернет-платформи і доступне на кожній сторінці. Після вибору "Головної" відбудеться переадресація на сторінку з асортиментом товарів.

Після додавання товару в корзину, відображається повідомлення про кількість товарів в корзині, який товар було додано та наступні дії:

						Аркуш
					ДТЕУ 121 02-11.МР	43
Зм.	Аркуш	№ докум	Підпис	Дата		



«В корзину», що переадресує користувача до сторінки з корзиною.

«Закрити», що закріє дане повідомлення і користувач зможе далі продовжити вибір товарів (рисунок 3.8).

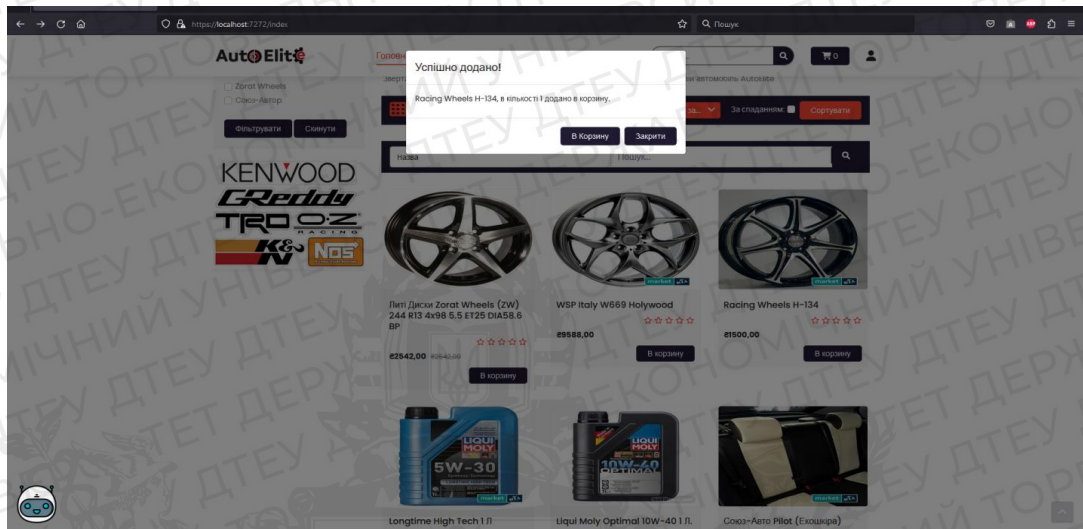


Рис. 3.8. Головна сторінка після додавання товару в корзину

Джерело: розроблено автором, скріншот з екрану

Натиснувши на піктограму корзини, користувачу буде відображено список обраних товарів, їх сумарна знижка та можливість змінити їх кількість (рисунок 3.9).

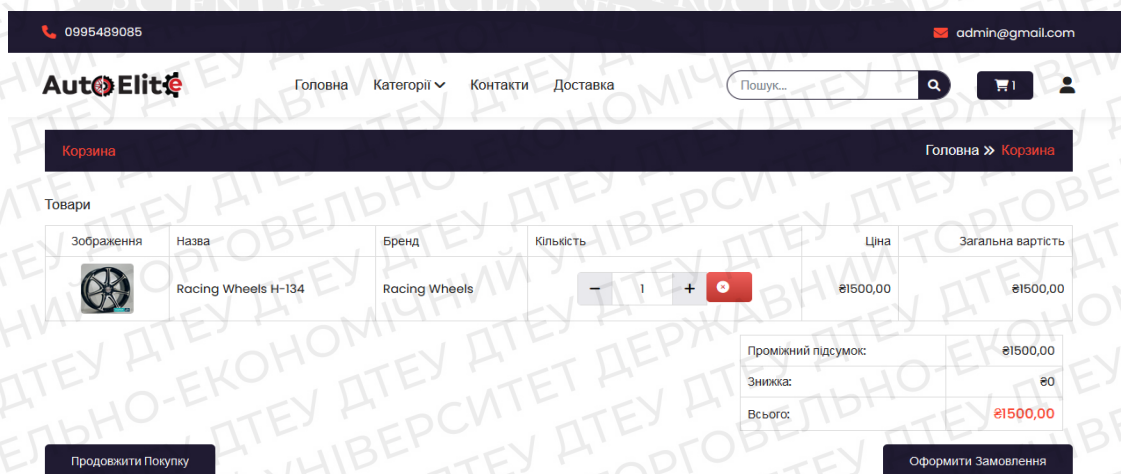


Рис. 3.9. Сторінка корзини

Джерело: розроблено автором, скріншот з екрану

					Аркуш
					44
Зм.	Аркуш	№ докум	Підпис	Дата	

ДТЕУ 121 02-11.МР

Після натиснення кнопки «Оформити замовлення», відбудеться переадресація на сторінку з вибором доставки замовлення, де користувач може змінити адресу доставки та обрати тип доставки, та спосіб оплати (рисунки 3.10).

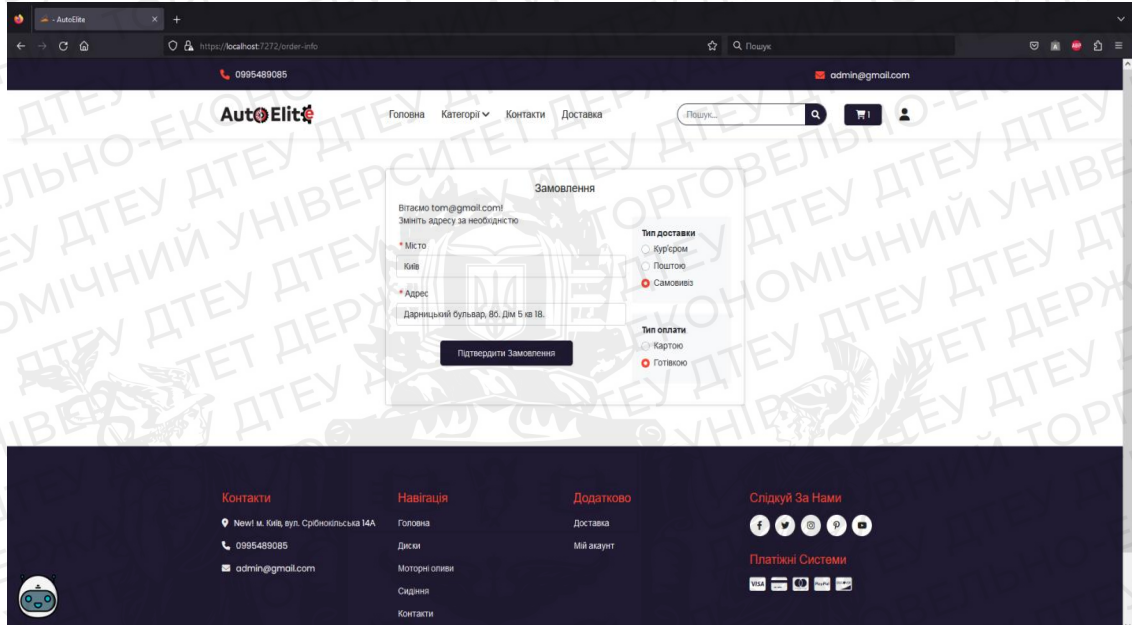


Рис. 3.10. Сторінка вибору доставки  
Джерело: розроблено автором, скріншот з екрану

Після вибору типу оплати «Готівкою», та підтвердження замовлення, користувачу відобразиться повідомлення про успішне замовлення (рисунки 3.11).

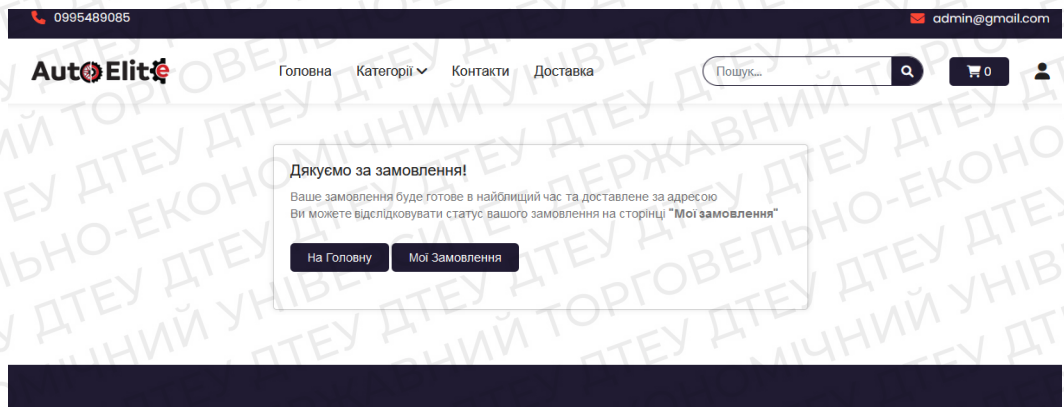


Рис. 3.11. Повідомлення про успішне замовлення  
Джерело: розроблено автором, скріншот з екрану

						Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



Якщо користувач обере тип оплати «Карткою», його переадресує на платіжну систему де він зможе оплатити замовлення. Після успішної оплати користувач побачить повідомлення про успішне замовлення (рисунок 3.12).

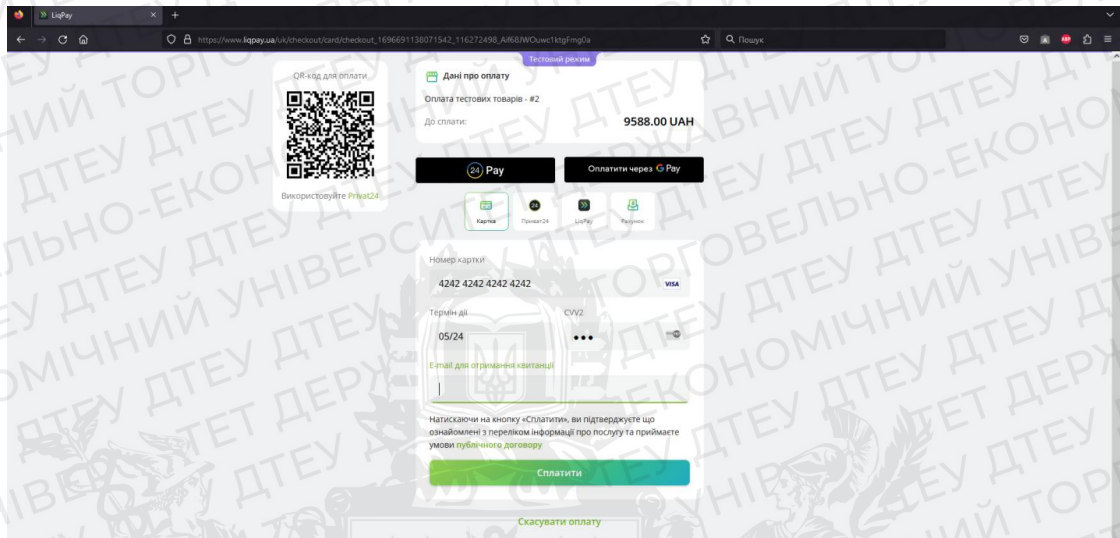


Рис. 3.12. Форма платіжного сервісу LiqPay

*Джерело: розроблено автором, скріншот з екрану*

### 3.4. Умови виконання програми

Для функціонування інтернет-платформи необхідно:

- Операційна система: Windows Server (наприклад, Windows Server 2016 або більше нова версія).
  - Процесор: Мінімум 2-ядерний, рекомендується більше для підвищеної продуктивності.
  - Оперативна пам'ять: Мінімум 2ГБ, рекомендується більше для оптимальної швидкодії.
  - Дисковий простір: Залежить від розміру магазину та кількості товарів, але рекомендується мінімум 20 ГБ.
- База даних:
- Система управління базами даних (СУБД): Microsoft SQL Server.

						Аркуш
						46
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

– Дисконий простір для бази даних: Від 10 ГБ і вище в залежності від обсягу даних.

Серверне програмне забезпечення:

- ASP.NET MVC: Остання версія фреймворка ASP.NET MVC.
- IIS (Internet Information Services): Веб-сервер IIS для хостингу ASP.NET додатків.

Мережевий доступ:

- Статична IP-адреса: Для надійного доступу до магазину.
- SSL-сертифікат: Для забезпечення безпеки передачі даних клієнтами.

Апаратурний склад який має бути: монітор, миша, клавіатура.

### 3.5. Повідомлення користувачу

Якщо користувач перейде на неіснуючу сторінку, або захоче переглянути товар якого не існує, він потрїпить на сторінку яка його повідомить про це (рисунок 3.13).

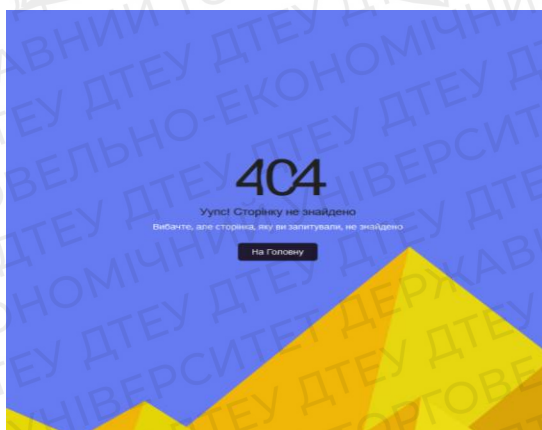


Рис. 3.13. Сторінка з помилкою статусу коду 404  
Джерело: розроблено автором, скріншот з екрану

						Аркуш
					ДТЕУ 121 02-11.МР	47
Зм.	Аркуш	№ докум	Підпис	Дата		



При заповненні форм, якщо користувач не вкаже дані обов'язкових полів, або не введе допустиму кількість символів, він отримає попередження (рисунок 3.14).

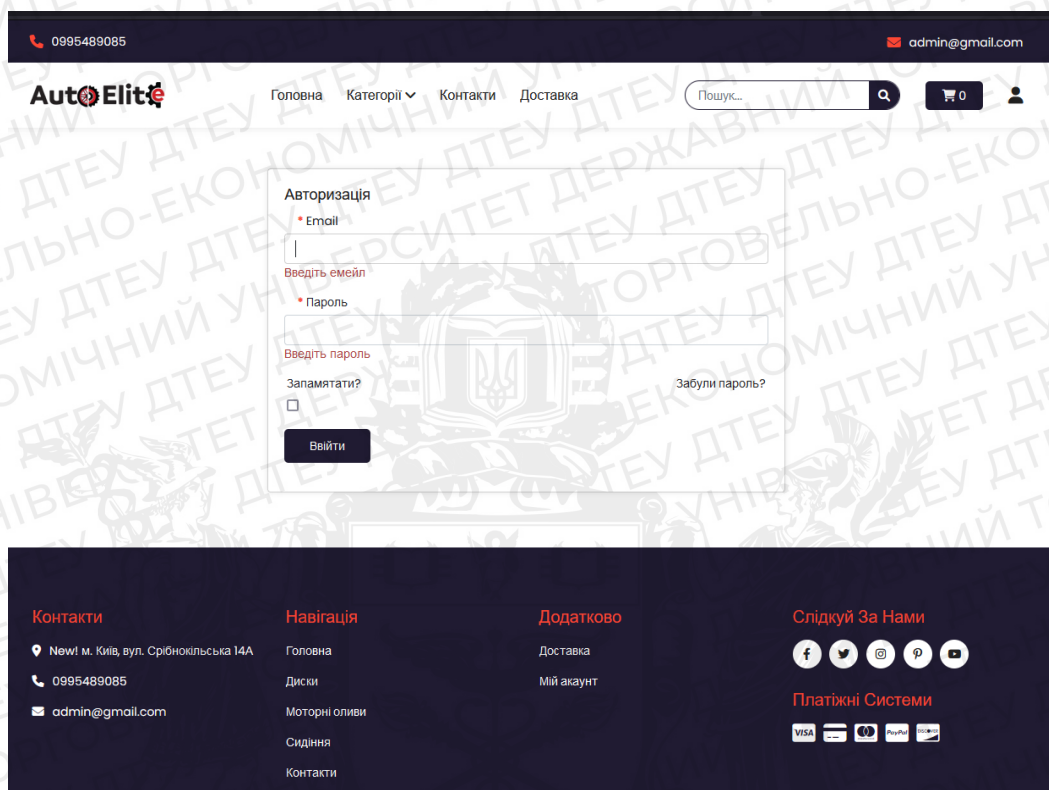


Рис. 3.14. Сторінка авторизації при відправці незаповнених полів  
 Джерело: розроблено автором, скріншот з екрану

### 3.6. Керівництво адміністратора

Файли інтернет-платформи зберігаються з такою структурою:

*ECommerce/Controllers* – контролери які відповідають за логіку обробки сторінок.

*ECommerce/Data* – тут зберігаються файли розширення, ініціалізації та конфігурації бази даних .

*ECommerce/Interfaces* – інтерфейси які відповідають за абстракцію для функціоналу.

						Аркуш
						48
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

*ECommerce/Models* – моделі які відповідають за представлення даних та бізнес-логіку.

*ECommerce/Repository* – репозиторії, які відповідають за доступ до даних і виконують роль посередника між моделями та джерелами даних, такими як бази даних.

*ECommerce/ViewComponents* – файли , які відповідають за побудову і відображення частин сторінок (компонентів) зі спеціалізованим відображенням та логікою вбудованого вигляду.

*ECommerce/ViewModels* – відповідають за представлення даних, які повинні бути відображені на сторінках. Вони представляють собою об'єкти даних, які підготовлені спеціально для конкретних сторінок чи компонентів інтерфейсу користувача, і надають можливість передавати ці дані з контролера в представлення для відображення.

*ECommerce/Views* – відповідає за відображення (presentation layer) інтернет-платформи. Містить HTML-файли та інші ресурси, які визначають, як додаток буде виглядати для користувача на сторінках інтернет-платформи.

*ECommerce/wwwroot* – відповідає за зберігання статичних ресурсів, таких як CSS файли, JavaScript файли, зображення, шрифти і інші файлові ресурси, які використовуються на сторінках інтернет-платформи.

*ECommerce/appsettings.json* - використовується для зберігання конфігураційних даних.

*ECommerce/Program.cs* - відповідає за налаштування та запуск інтернет-платформи. Він є важливим вхідним файлом, який дозволяє налаштувати середовище виконання програми.

Після того, як адміністратор успішно авторизується і перейде до панелі управління, він отримує доступ до головної сторінки. На цій сторінці він може переглядати загальну інформацію про активність клієнтів та загальну кількість товарів, замовлень, категорій, ролей (рисунок 4.15).

						Аркуш
						49
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



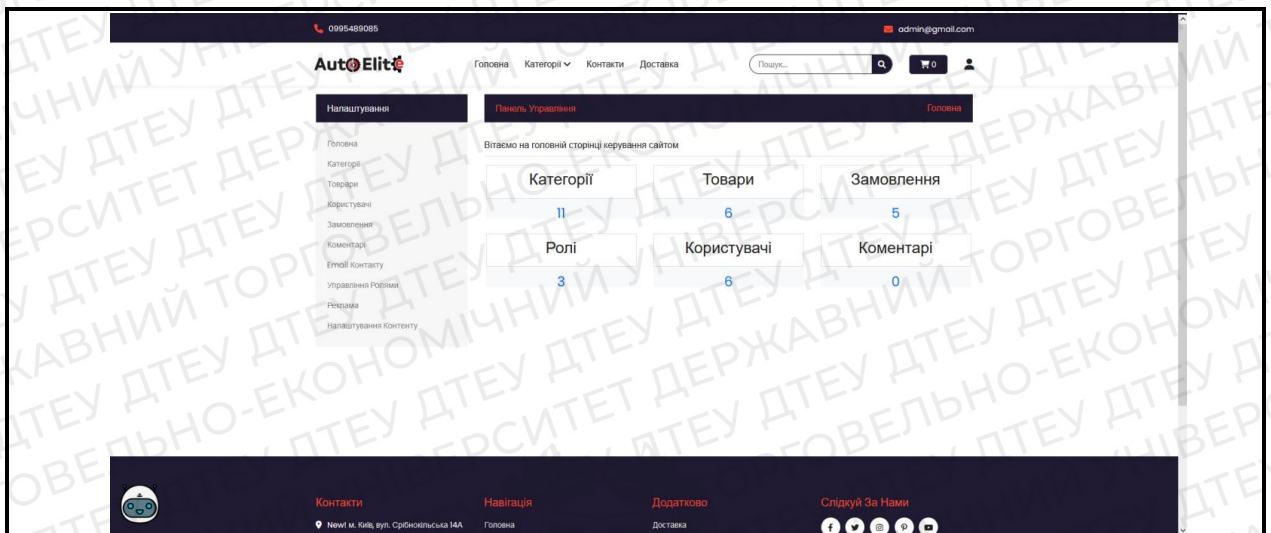


Рис. 4.15. Головна сторінка адміністратора  
*Джерело: розроблено автором, скріншот з екрану*

Подальший функціонал для адміністратора включає в себе розділ "Замовлення", де адміністратор може перевірити та видалити замовлення користувачів.

Важливим аспектом є розділ "Товари", який дозволяє адміністратору переглядати всі наявні товари у магазині. Тут є можливість пошук товарів за категоріями, назвою, описом, брендом для полегшення пошуку. Адміністратор може внести зміни до інформації про товари, видалити їх або додати нові.

Розділ "Коментарі" містить відгуки, які були залишені користувачами під товарами. Адміністратор може розглядати та видаляти ці відгуки.

"Email контакту" змінює електронну пошту на яку буде надходити зв'язок з адміністрацією інтернет-платформи через контактну форму.

"Категорії" дозволяють додавати нові категорії та підкатегорії товарів або видаляти застарілі.

Розділ "Користувачі" містить важливі відомості про клієнтів магазину.

В розділі "Реклама" дозволяє редагувати, додавати чи видаляти новини.

"Налаштування контенту" цей розділ створений для управління

						Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

контентом інтернет-платформи, таким як: логотип, заголовок, основна картинка

інтернет-платформи, контактна інформація та реклама на сторінці входу.

Головний адміністратор також має можливість надавати або забирати права іншим адміністраторам для взаємодії з контентом інтернет-платформи, розширюючи або обмежуючи їх можливості. За потреби може бути доданий новий адміністратор.

### 3.7 Висновки до Розділу 3

В рамках розділу були розглянуті різні інструменти та засоби розробки, включаючи популярні фреймворки, мови програмування та інші технології. Обраний набір інструментів та технологій відповідає потребам підприємства торгівлі та забезпечує реалізацію функціональних вимог.

У розділі був наданий докладний опис інтерфейсу інтернет-платформи, який включає: дизайн, структуру сторінок та елементи керування, такі як меню, кнопки, форми та інші елементи, які взаємодіють з користувачем.

У розділі були наведені основні функції, доступні користувачам на інтернет-платформі. Це включало в себе перегляд товарів, додавання їх до кошика, оформлення замовлення, перегляд інформації про акції та новини, а також можливість залишати відгуки.

Були описані умови виконання програми, наведені вимоги до введення даних користувачами та умови використання інтернет-платформи.

Один з важливих аспектів інтерфейсу - це сповіщення та повідомлення для користувачів. У розділі було описано механізми сповіщення про успішні дії, помилки або іншу інформацію, яка може бути корисною користувачам.

Розділ містив інформацію про роль та функції адміністратора інтернет-платформи. Це включало в себе можливості управління товарами,

						Аркуш
						51
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



замовленнями, відгуками користувачів, налаштування акцій та інших параметрів інтернет-платформи.

Детальний опис інтерфейсу користувача, його функцій та можливостей, умов виконання програми та керівництва адміністратора створило чітку картину та розуміння роботи інтернет-платформи для підприємства торгівлі. Це важливий етап в розробці проекту, що сприяє його успішному впровадженню та використанню.



						Аркуш
						52
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Тема виконаної роботи є актуальною у сучасному бізнес-середовищі, оскільки інтернет-платформи для підприємств торгівлі стали ключовим інструментом для залучення клієнтів та покращення обслуговування.

Розроблена архітектура програмного комплексу дозволила створити інтернет-платформу, яка забезпечує високу ефективність та зручність для користувачів. Ця платформа надає широкий спектр сервісів та функцій, що дозволяють користувачам швидко та зручно переглядати товари, робити замовлення, виконувати оплату та відстежувати стан замовлень. Завдяки розробленій архітектурі, платформа є надійною та масштабованою, забезпечуючи високий рівень безпеки та цілісності даних.

Використання сучасних мов програмування та технологій, таких як ASP.NET Core, C#, Entity Framework, дозволило створити ефективний та функціональний програмний комплекс.

Безпека є пріоритетом, і проект включає в себе заходи для захисту конфіденційної інформації, аутентифікації та авторизації користувачів.

Розроблена структура бази даних відповідає потребам та забезпечує ефективне зберігання даних.

Інтерфейс користувача є інтуїтивно зрозумілим та зручним у використанні. Важливим етапом є ефективне управління проектом, враховуючи терміни виконання та ресурси.

Загалом, програмне забезпечення є добре підготовленим та обґрунтованим, і сприяє покращенню діяльності підприємства торгівлі в онлайн-середовищі.

					<i>ДТЕУ 121 02-11.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			01.11.23	<i>Архітектура програмного комплексу інтернет-платформи підприємств торгівлі</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Десятко А.М.			01.11.23		<i>ВП</i>	<i>53</i>	<i>55</i>
Гарант	Котенко Н.О.			01.11.23		<i>Факультет інформаційних технологій</i> <i>2м курс, 2 група</i>		
Розробив	Міщенко В.Ю.			01.11.23				
					<i>Висновки та пропозиції</i>			

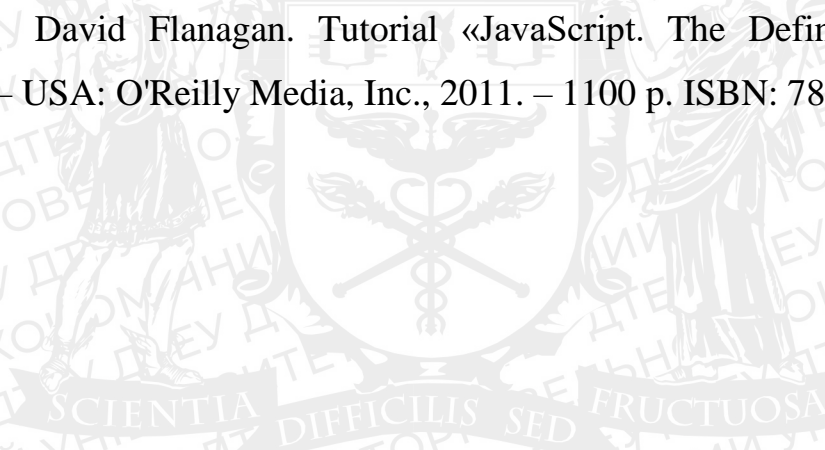


## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Exist.ua. Електронний ресурс. URL: <https://exist.ua/uk/> (10.04.2023)
2. E-shop автозапчастин. Електронний ресурс. URL: <https://eshop.elit.ua/> (10.04.2023)
3. E-detail. Електронний ресурс. URL: <https://e-detail.in.ua/> (10.04.2023)
4. Autopartner.in.ua. Електронний ресурс. URL: <https://autopartner.in.ua/> (10.04.2023)
5. Booch G, Rumbaugh J., Jacobson I. Tutorial «The Unified Modeling Language User Guide (2nd Edition)» – USA: Wesley Professional, 1998. – 512 p. ISBN: 0-201-57168-4
6. «Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition» – USA: Addison-Wesley, 1995. – 396 p. ISBN: 978-0201633610
7. Geeksforgeeks. Електронний ресурс. URL: <https://www.geeksforgeeks.org/introduction-to-laravel-and-mvc-framework/> (24.05.2023)
8. C-sharpcorner. Електронний ресурс. URL: <https://www.c-sharpcorner.com/article/introduction-of-the-entity-framework-core-6-0/> (24.05.2023)
9. Jon Skeet. Tutorial «C# in Depth: Fourth Edition 4th Edition» – USA: Manning Publications Co, 2019. – 528 p. ISBN: 9781617294532
10. Mark J. Price. Tutorial «C# 9 and .NET 5 – Modern Cross-Platform Development: Build intelligent apps, websites, and services with Blazor ASP.NET Core, and Entity Framework Core using Visual Studio Code, 5th Edition» – England: Packt Publishing, 2020. – 822 p. ISBN: 978-1-80056-810-5

<i>ДТЕУ 121 02-11.МР</i>				
Зм.	Аркуш	№ докум.	Підпис	Дата
Зав. каф.		Криворучко О.В.		01.11.23
Керівник		Десятко А.М.		01.11.23
Гарант		Котенко Н.О.		01.11.23
Розробив		Мищенко В.Ю.		01.11.23
<i>Архітектура програмного комплексу інтернет-платформи підприємства торгівлі</i>				
<i>Список використаних джерел</i>				
		<i>Факультет інформаційних технологій</i>		
		<i>2м курс, 2 група</i>		
		<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
		<i>СВД</i>	<i>54</i>	<i>55</i>

11. Adam Freeman. Tutorial «Pro ASP.NET MVC 5 (Expert's Voice in ASP.Net) 5th ed. Edition» – USA: Apress, 2013. – 832 p. ISBN: 978-1430265290
12. Jon P Smith. Tutorial « Entity Framework Core in Action, 2nd Edition» – USA: Manning Publications, 2021. – 624 p. ISBN: 9781617298363
13. Eric Freeman, Elisabeth Robson « Head First HTML5 Programming» – USA: O'Reilly Media, Inc., 2011. – 593 p. ISBN: 9781449390549
14. Ерік А. Мейер, Естелл Вейл. CSS повний довідник.-К.: Діалектика, 2019. – 1088с. ISBN: 978-5-907114-56-2
15. Waymire. Tutorial «Teach Yourself MS SQL Server 2000 in 21 Days» – USA: Sams Publishing, 2000. – 854 p. ISBN: 78-0-67231-969-3
16. David Flanagan. Tutorial «JavaScript. The Definitive Guide. 6th Edition» – USA: O'Reilly Media, Inc., 2011. – 1100 p. ISBN: 78-0-67231-969-3



					<i>ДТЕУ 121 02-11.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		55



## ТЕХНІЧНЕ ЗАВДАННЯ

### 1. Завдання для виконання

#### 1.1. Аналіз предметної області:

- Провести аналіз потреб підприємства торгівлі та конкурентного середовища.

#### 1.2. Проектування архітектури:

- Розробити детальний план архітектури програмного комплексу.
- Визначити структуру бази даних та забезпечити її інтеграцію.

#### 1.3. Розробка та тестування:

- Реалізувати серверну та клієнтську частини програмного комплексу.
- Провести тестування для виявлення та виправлення помилок.

#### 1.4. Впровадження та підтримка

### 2. Вимоги до програмного комплексу

#### 2.1. Функціональні вимоги:

- Реєстрація та авторизація користувачів з різними ролями (адміністратор, клієнт).
- Додавання, редагування та видалення товарів та послуг в базі даних.
- Можливість оформлення та оплати замовлень користувачами.
- Зв'язок з адміністрацією.
- Інформація про доставку.

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 02-11.МР			
Зав. каф.		Криворучко О.В.		06.09.23	Архітектура програмного комплексу інтернет-платформи підприємства торгівлі	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		06.09.23		ТЗ	56	55
Гарант		Котенко Н.О.		06.09.23		Факультет інформаційних технологій		
Розробив		Мищенко В.Ю.		06.09.23		2м курс, 2 група		
						Технічне завдання		

- Відгуки та рейтинг товарів.
- Адміністративний доступ до управління товарами, категоріями, відгуками, замовленнями та користувачами.
- Управління контентом.

## 2.2. Нефункціональні вимоги:

- Швидка та ефективна робота програмного комплексу.
- Інтуїтивний та зручний інтерфейс користувача.
- Забезпечення сумісності з різними веб-браузерами та пристроями.

## 3. Архітектурна концепція

### 3.1. Архітектура програмного комплексу:

- Застосування архітектурного підходу Model-View-Controller (MVC) для розділення логіки додатку на компоненти.
- Використання серверного та клієнтського програмного забезпечення для забезпечення багатокористувацького доступу та зручності користувачів.

### 3.2. Технології та мови програмування:

- ASP.NET Core та мова програмування C# для розробки серверної частини.
- HTML, CSS, JavaScript, jQuery для розробки клієнтської частини.

## 4. Архітектурні вимоги

### 4.1. Безпека:

- Захист від SQL-ін'єкцій та інших атак на базу даних.
- Використання механізмів аутентифікації та авторизації для контролю доступу до функціональності.

### 4.2. Масштабованість:

						Аркуш
						57
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	



- Забезпечення можливості масштабування системи з ростом навантаження.

## 5. База даних

### 5.1. Структура бази даних:

- Таблиця "Користувачі" для зберігання інформації про користувачів.
- Таблиця "Товари" для зберігання інформації про товари та послуги.
- Таблиця "Картинка товару" для зберігання картинок про товари.
- Таблиця "Замовлення" для зберігання інформації про замовлення користувачів.
- Таблиця "Деталі замовлення" для зберігання інформації про деталі замовлення користувачів.
- Таблиця "Корзина" для зберігання інформації про обрані користувачів.
- Таблиця "Коментар" для зберігання коментарів користувачів, та їх оцінку товару.
- Таблиця "Категорії" для зберігання категорій та під категорій товарів.
- Таблиця "Реклама" для зберігання рекламних оголошень.
- Таблиця "Налаштування контенту" для зберігання основних даних налаштування інтернет-платформи, таких як контактна інформація і т.д.

						Аркуш
						58
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

## ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

Щоб перевірити коректність роботи було створено модульні тести які передбачають виконання тестів окремими модулями (класи або методи) програмного комплексу для перевірки їхньої коректності та відповідності вимогам.

У процесі тестування даного програмного комплексу було використано бібліотеки Xunit та Moq (бібліотека для створення моків - фіктивних об'єктів) для створення і виконання модульних тестів.

**Xunit** - це фреймворк для модульного тестування в мові програмування C#. Він надає зручний інтерфейс для написання та виконання тестових сценаріїв, а також забезпечує зручну інтеграцію з іншими інструментами розробки.

**Moq** - це бібліотека для створення мок-об'єктів в мові програмування C#. Моки використовуються для імітації реальних об'єктів під час тестування, щоб ізолювати тестовий код від залежностей та забезпечити більший контроль над тестами.

Загалом було створено 113 тестів для контролерів, що покрито близько 80% коду.

Ось один з зразків тесту для тестування методу Categories контролера Category

```
[Fact]
public void CategoriesReturnsAViewResultWithAListOfCategories()
{
    // Arrange
```

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 02 - 9.МР			
Зав. каф.		Криворучко О.В.		06.09.23	Архітектура програмного комплексу інтернет-платформи підприємства торгівлі	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		06.09.23		ПМТ	59	57
Гарант		Котенко Н.О.		06.09.23		Факультет інформаційних технологій		
Розробив		Мищенко В.Ю.		06.09.23		2м курс, 6 група		
					Програма та методика тестування			



```

var mockCategory = new Mock<ICategory>();
var mockWebHostEnvironment = new Mock<IWebHostEnvironment>();

mockCategory.Setup(repo =>
repo.GetAllCategoriesWithSub(It.IsAny<QueryOptions>()))
.Returns(new PagedList<Category>(GetTestCategories(), new QueryOptions()));
var controller = new CategoryController(mockCategory.Object,
mockWebHostEnvironment.Object);

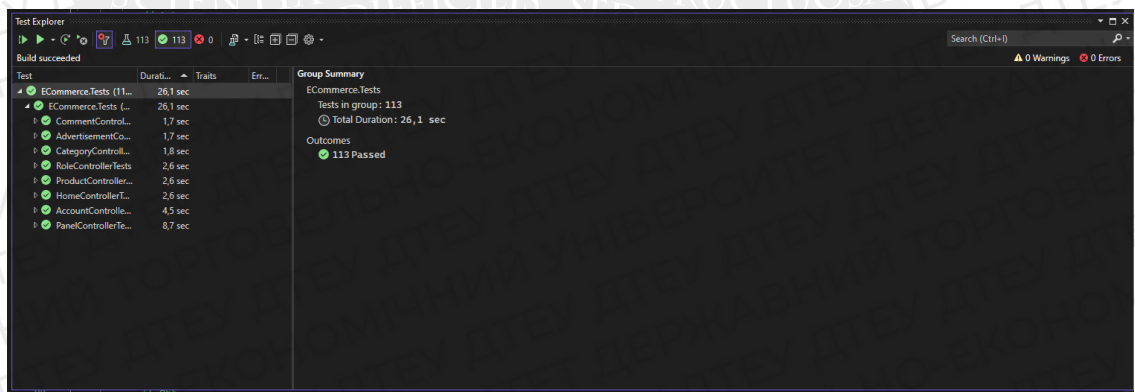
// Act
var result = controller.Categories(new QueryOptions());

// Assert
var viewResult = Assert.IsType<ViewResult>(result);
var model = Assert.IsAssignableFrom<PagedList<Category>>(viewResult.Model);
Assert.Equal(GetTestCategories().Count(), model.Count());
}
private IQueryable<Category> GetTestCategories()
{
var categories = new List<Category>
{
new Category{ Id = 1, Name = "First"},
new Category{ Id = 2, Name = "Second"},
new Category{ Id = 3, Name = "Third"},
new Category{ Id = 4, Name = "Forth"}
};

return categories.AsQueryable();
}

```

На рисунку представлено успішні результати проходження 113 тестів для різних контролерів.



Також проводилось тестування продуктивності під навантаженням.

Для цього було розроблено збережену процедуру що дозволяє генерувати тестові дані в БД, які будуть навантажувати систему.

```

public static async Task CreateSeedDataAsync(ApplicationContext context, int[] categories, int
rowCount = 25, bool clearData = true)
{

```

						Аркуш
						60
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

```

if (clearData)
{
    ClearData(context);
}
context.Database.SetCommandTimeout(TimeSpan.FromMinutes(10));
context.Database.ExecuteSqlRaw("drop procedure if exists CreateSeedData");
context.Database.ExecuteSqlRaw(@"
CREATE PROCEDURE CreateSeedData
@RowCount INT,
@firstCategoryId INT,
@secondCategoryId INT,
@thirdCategoryId INT
AS
BEGIN
SET NOCOUNT ON
DECLARE @i INT = 0;
DECLARE @currentId INT = 7;

WHILE @i < @RowCount
BEGIN
INSERT INTO Products (Name,Description,CategoryId,Brand,Weight,Price,Type)
VALUES (CONCAT('First - ',@i), CONCAT('Grate First - ', @i), @firstCategoryId,
CONCAT('Brand - ', @i), RAND() * (500-50+1), RAND() * (300 - 50 + 1), 0);
SET @i = @i + 1
END

SET @i = 0;

WHILE @i < @RowCount
BEGIN
INSERT INTO ProductImages (Image,ProductId)
VALUES (CONCAT('/productImages/', 'first.png'), @currentId);
SET @currentId = @currentId + 1;
SET @i = @i + 1
END

SET @i = 0;

WHILE @i < @RowCount
BEGIN
INSERT INTO Products (Name,Description,CategoryId,Brand,Weight,Price,Type)
VALUES (CONCAT('Second - ',@i), CONCAT('Ice second - ', @i),
@secondCategoryId,
CONCAT('Brand - ', @i), RAND() * (500-50+1), RAND() * (300 - 50 + 1), 1);
SET @i = @i + 1
END

SET @i = 0;

WHILE @i < @RowCount
BEGIN
INSERT INTO ProductImages (Image,ProductId)
VALUES (CONCAT('/productImages/', 'second.png'), @currentId);

```

						Аркуш
						61
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 02-11.МР</i>	



```

SET @currentId = @currentId + 1;
SET @i = @i + 1
END

SET @i = 0;

WHILE @i < @RowCount
BEGIN
INSERT INTO Products (Name,Description,CategoryId,Brand,Weight,Price,Type)
VALUES (CONCAT('Third - ',@i), CONCAT('Wow third- ', @i),
@thirdCategoryId,
CONCAT('Brand - ', @i), RAND() * (500-50+1), RAND() * (300 - 50 + 1), 0);
SET @i = @i + 1
END

SET @i = 0;

WHILE @i < @RowCount
BEGIN
INSERT INTO ProductImages (Image,ProductId)
VALUES (CONCAT('/productImages/', 'third.png'), @currentId);
SET @currentId = @currentId + 1;
SET @i = @i + 1
END

COMMIT
END");
context.Database.BeginTransaction();
context.Database.ExecuteSqlRaw($"EXEC CreateSeedData @RowCount
= {rowCount}, @firstCategoryId = {categories[0]}," +
"@secondCategoryId = {categories[1]}, @thirdCategoryId = {categories[2]}");
context.Database.CommitTransaction();
}

```

						Аркуш
						62
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-11.МР	

## ДОДАТКИ

Додаток А

### Лістинг програмного коду

Контекст бази даних для веб-додатку, все це дозволяє визначити структуру бази даних і налаштувати правила для моделі даних, що використовується в програмі, зокрема, використовується Entity Framework Core для роботи з базою даних і сутностями.

```
using ECommerce.Models;
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;
using PizzaShop.Models;
using PizzaShop.Models.Cart;
using PizzaShop.Models.Checkout;
using System.Reflection.Emit;

namespace WebApplication1.Data
{
    public class ApplicationDbContext : IdentityDbContext<User>
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
            //Database.EnsureDeleted();
            // Database.EnsureCreated();
        }

        public DbSet<Category> Categories { get; set; } = null!;
        public DbSet<Product> Products { get; set; } = null!;
        public DbSet<ShopCartItem> ShopCartItems { get; set; } = null!;
        public DbSet<Order> Orders { get; set; } = null!;
        public DbSet<OrderDetails> OrderDetails { get; set; } = null!;
        public DbSet<Comment> Comments { get; set; } = null!;
        public DbSet<Advertisement> Advertisements { get; set; } = null!;
        public DbSet<ProductImage> ProductImages { get; set; } = null!;
        public DbSet<ContentSettings> ContentSettings { get; set; } = null!;
        public DbSet<ModelForPanelIndex> ModelForPanelIndex { get; set; } = null!;
        protected override void OnModelCreating(ModelBuilder builder)
        {
            builder.Entity<Category>().Property(e =>
            e.DateOfPublication).HasDefaultValueSql("GETDATE()");
            builder.Entity<Product>().Property(e =>
            e.DateOfPublication).HasDefaultValueSql("GETDATE()");
            builder.Entity<Product>().Property(e => e.Rating).HasDefaultValue(0);
            builder.Entity<Product>().Property(e => e.NumberOfVotes).HasDefaultValue(0);
            builder.Entity<ShopCartItem>().Property(e=>
            e.CreateAt).HasDefaultValueSql("GETDATE()");
            builder.Entity<Order>().Property(e => e.CreateAt).HasDefaultValueSql("GETDATE()");
            builder.Entity<Order>().Property(e => e.StatusPaid).HasDefaultValue(false);
            builder.Entity<Comment>().Property(e => e.CreateAt).HasDefaultValueSql("GETDATE()");
```



```

        builder.Entity<Advertisement>().Property(e =>
            e.DateOfPublication).HasDefaultValueSql("GETDATE()");
        builder.Entity<ModelForPanelIndex>().HasNoKey();
        base.OnModelCreating(builder);
    }
}
}

```

Контроллер корзины ShopCartController для управления корзиною.

```

public class ShopCartController : Controller
{
    private readonly CartRepository _cart;
    private readonly IProduct _products;
    public ShopCartController(CartRepository cart, IProduct products)
    {
        _cart = cart;
        _products = products;
    }

    [Route("/cart")]
    [HttpGet]
    public async Task<IActionResult> Index()
    {
        ViewBag.CategoryId = double.NaN;
        return View(await _cart.GetShopCartItemAsync());
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> AddToCart(int productId, string returnUrl, int
quantity = 1, bool isModal = true)
    {
        var currentProduct = await _products.GetProductByIdAsync(productId);
        if (currentProduct != null)
        {
            await _cart.AddToCartAsync(currentProduct, quantity);
            if (isModal)
            {
                return PartialView("_ConfirmModal", (currentProduct.Name,
quantity));
            }
        }
        if (returnUrl != null)
        {
            return Redirect(returnUrl);
        }
        return RedirectToAction(nameof(Index));
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> IncrementQuantity(int shopCartItemId)
    {

```

```

        var currentShopCartItem = await
        _cart.GetShopCartItemByIdAsync(shopCartItemId);
        if (currentShopCartItem != null)
        {
            currentShopCartItem.Count += 1;
            await _cart.UpdateFromCartAsync(currentShopCartItem);
        }
        return RedirectToAction(nameof(Index));
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> DecrementQuantity(int shopCartItemId)
    {
        var currentShopCartItem = await
        _cart.GetShopCartItemByIdAsync(shopCartItemId);
        if (currentShopCartItem != null && currentShopCartItem.Count > 1)
        {
            currentShopCartItem.Count -= 1;
            await _cart.UpdateFromCartAsync(currentShopCartItem);
        }
        return RedirectToAction(nameof(Index));
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> RemoveFromCart(int shopCartItemId)
    {
        var currentShopCartItem = await
        _cart.GetShopCartItemByIdAsync(shopCartItemId);
        if (currentShopCartItem != null)
        {
            await _cart.RemoveFromCartAsync(currentShopCartItem);
        }
        return RedirectToAction(nameof(Index));
    }
}

```