

# ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

## «Веб-орієнтований додаток «Anime Gift-merch» з просування тематичних аніме товарів»

Студента 4 курсу, 7 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми  
«Інженерія  
програмного забезпечення»

\_\_\_\_\_

підпис студента

Дьяконова  
Віталія  
Олексійовича

Науковий керівник, PhD, доцент  
заступник завідувача кафедри  
програмного забезпечення та  
кібербезпеки

\_\_\_\_\_

підпис керівника

Десятко  
Альона  
Миколаївна

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

\_\_\_\_\_

підпис керівника

Рзасва Світлана  
Леонідівна

# Державний торговельно-економічний університет

Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення та кібербезпеки  
Освітній ступінь бакалавр  
Спеціальність 121 «Інженерія програмного забезпечення»

**Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О.В.

«14» листопада 2022 р.

## **Завдання на випускний кваліфікаційний проєкт студентів**

Дьяконову Віталію Олексійовичу  
(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «веб-орієнтований додаток з  
просування тематичних аніме товарів «Anime Gift-merch»»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту: розробка веб-додатку для реалізації аніме товарів, що зможе  
забезпечити зрозумілий і зручний інтерфейс для користувачів для покупки  
ними тих товарів які вони потребують.

Об'єкт дослідження: процес просування та реалізації тематичних аніме  
товарів в інтернет просторі.

Предмет дослідження: розробка та проектування архітектури веб-додатку  
з реалізації аніме товарів, який дозволить зручно купляти ті товари яких  
потребують користувачі.

4.Консультанти проекту із зазначенням розділів, які консультують:

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

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ТЕОРЕТИЧНОЇ ЧАСТИНИ ВЕБ ДОДАТКІВ

1.1. Актуальність вибраної теми

1.2. Технічне завдання проекту

1.3. Стан дослідження проблеми

1.4. Висновок по першому розділу

РОЗДІЛ 2. АНАЛІЗ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКА ДЛЯ ПРОСУВАННЯ АНІМЕ ТОВАРІВ

2.1. Огляд Сучасних технологій розробки веб-додатків

2.2. Архітектура веб додатків

2.3. Бази даних для веб-додатків

2.4. Вибір технологій для розробки веб-додатка

2.4. Забезпечення безпеки веб-додатків

2.6. Висновок по другому розділу

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКА З ПРОСУВАННЯ ТЕМАТИЧНИХ АНІМЕ ТОВАРІВ

3.1. Вибір архітектури веб-додатка з просування аніме товарів

3.2. Функціональні вимоги до веб-додатка з просування аніме товарів

3.3. Архітектура бази даних веб-додатка з просування аніме товарів

3.4. Розробка користувацького інтерфейсу веб-додатку

3.5. Реалізація функціональних модулів веб-додатку

3.6. Висновки по третьому розділу

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>розділ 1. Аналіз теоретичної частини розробки веб додатків</i>	27.01.2023	27.01.2023
5.	<i>розділ 2. Аналіз та проектування веб-додатка для просування тематичних аніме товарів</i>	03.03.2023	03.03.2023
6.	<i>розділ 3. Реалізація веб-додатка для просування тематичних аніме товарів</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	29.05.2023 - 02.06.2023	
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	05.06.2023	05.06.2023
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту  
Десятко А.М.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Дьяконов В.О.

(прізвище, ініціали, підпис)

## 11. Відгук керівника випускного кваліфікаційного проєкту

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Науковий керівник випускного кваліфікаційного проєкту

(підпис, дата)

Відмітка про попередній захист

(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента Дьяконов В.О.

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми Рзаєва С.Л.

(прізвище, ініціали, підпис)

Завідувач кафедри

Криворучко О.В.

(прізвище, ініціали, підпис)

«                    »

20                    р.

## АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці веб-додатку для просування тематичних аніме товарів

В результаті порівняльного аналізу аналогічних рішень визначено ....

Розробка серверної і клієнтської частини програмного продукту виконано в середовищі Visual Studio Code. Готовий програмний комплекс було успішно протестовано

## ABSTRACT

According to the purpose of the research, the work is devoted to the development of a web application for the promotion of thematic anime products As a result of the comparative analysis of similar decisions, development of the server and client part of the software product was carried out in the Visual Studio Code environment. The finished software package was successfully tested

## ЗМІСТ

<b>ВСТУП.....</b>	<b>8</b>
<b>РОЗДІЛ 1 АНАЛІЗ ТЕОРЕТИЧНОЇ ЧАСТИНИ РОЗРОБКИ ВЕБ ДОДАТКІВ.....</b>	<b>10</b>
1.1. Актуальність вибраної теми.....	10
1.2. Технічне завдання.....	12
1.3. Стан дослідження проблеми.....	16
1.4. Висновки по першому розділу.....	24
<b>РОЗДІЛ 2 АНАЛІЗ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКА ДЛЯ ПРОСУВАННЯ ТЕМАТИЧНИХ АНІМЕ ТОВАРІВ.....</b>	<b>26</b>
2.1. Огляд сучасних технологій розробки веб-додатків.....	26
2.2. Архітектура веб-додатків.....	28
2.3. Бази даних для веб-додатків.....	32
2.4. Вибір технологій для розробки веб-додатка.....	34
2.5. Забезпечення безпеки веб-додатків.....	35
2.6. Висновки по другому розділу.....	37
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКА ДЛЯ ПРОСУВАННЯ ТЕМАТИЧНИХ АНІМЕ ТОВАРІВ.....</b>	<b>39</b>
3.1. Вибір архітектури веб-додатка з просування аніме товарів.....	39
3.2. Функціональні вимоги до веб-додатка з просування аніме товарів.....	40
3.3. Архітектура бази даних веб-додатка з просування аніме товарів.....	42
3.4. Розробка користувацького інтерфейсу веб-додатку.....	45
3.5. Реалізація функціональних модулів веб-додатку.....	48
3.6. Висновки по третьому розділу.....	50
<b>ВИСНОВКИ І ПРОПОЗИЦІЇ.....</b>	<b>52</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>54</b>

					<i>ДТЕУ 121 07-11.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-орієнтований додаток «Anime Gift-merch» з просування тематичних аніме товарів	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		<i>Зміст</i>	7	54
Керівник		Десятко А.М.		23.12.22		<i>Зміст</i>	<i>Факультет інформаційних технологій 4 курс, 7 група</i>	
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Дьяконов В.О.		23.12.22				

## ВСТУП

Веб додатки це важлива складова сучасного життя, що забезпечують зручне користування різноманітними сервісами та даними через мережу Інтернет. У контексті аніме індустрії, тематичні товари відіграють важливу роль у забезпеченні прибутків у індустрії.

Об'єктом дослідження є аніме індустрія та робота тематичними товарами. Предмет дослідження: розробка та проектування архітектури веб-додатку з реалізації аніме товарів, який дозволить зручно купляти ті товари яких потребують користувачі.

Основною метою проекту є розробка зручного та ефективного веб додатку, що надасть користувачам простий та зрозумілий інтерфейс для покупки тематичних аніме товарів з будь-якої точки країни.

Основними завданнями проекту є: аналіз вимог до додатку, проектування архітектури системи, розробка логіки додатку, тестування та підтримка додатку.

Практична значущість проекту полягає в покращенні ефективності реалізації аніме продукції, забезпеченні доступу до тематичної аніме продукції з будь-якої точки країни.

Тема проекту є актуальною, але досить мало вивченою. Існують різноманітні веб додатки для реалізації аніме продукції, але результатом цього проекту стане унікальний веб додаток у своєму роді. Додаток може бути корисним як фанатам аніме культури, так і не знайомим з цією нішею користувачам.

Дослідження та розробка веб додатку з просування аніме товарів має великий потенціал для практичного використання, а також для подальшого

					<i>ДТЕУ 121 07-11.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		23.12.22	Веб-орієнтований додаток «Anime Gift-merch» з просування тематичних аніме товарів	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		23.12.22		В	8	54
Гарант		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій		
Розробив		Дьяконов В.О.		23.12.22		4 курс, 7 група		
					<i>Вступ</i>			



розвитку і вдосконалення у майбутньому. Задача розробки додатку передбачає вирішення завдань, що знайомить розробника з новими технологіями та методами.

Отже, розробка веб для просування тематичних аніме товарів має великий потенціал для покращення якості та ефективності реалізації товарів.



					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

# РОЗДІЛ 1

## АНАЛІЗ ТЕОРЕТИЧНОЇ ЧАСТИНИ РОЗРОБКИ ВЕБ ДОДАТКІВ

### 1.1. Актуальність вибраної теми

Актуальність теми є важливим елементом будь-якої дослідницької роботи. Для відповідної обґрунтованості необхідно розглянути сучасний стан відповідної сфери, виявити її проблеми та недоліки, а також визначити наскільки вона актуальна.

Мета цього розділу - розглянути актуальність теми "веб-орієнтований додаток з просування тематичних аніме товарів". На сьогоднішній день інтернет-торгівля є дуже популярною і широко поширеною. Однак, існує проблема в тому, що більшість інтернет-магазинів не мають спеціалізованої тематики, а ті, які є, не забезпечують відповідної якості та асортименту. У зв'язку з цим, є необхідність створення веб-орієнтованого додатку з просування тематичних аніме товарів.

Інтернет-магазин з тематичним напрямком аніме є актуальним, оскільки аніме стало дуже популярним серед молоді та підлітків, тому попит на товари цього напрямку постійно зростає. Проте, відсутність відповідних веб-сайтів з сучасним та зручним інтерфейсом може стати причиною того, що багато покупців будуть обирати інші тематики. Однією з ключових переваг веб-орієнтованого додатку є зручність та доступність для покупців, які можуть здійснювати покупки не тільки з комп'ютера, а й з мобільних пристроїв відтак, магазин з тематичним напрямком аніме повинен мати зручний та зрозумілий інтерфейс, де покупець зможе швидко знайти потрібний товар та здійснити його замовлення. Відсутність такого інтерфейсу може стати причиною втрати покупців та негативно вплинути на дохід інтернет-магазину. Крім того, з

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 07-11.БР</i>			
Зав. каф.		Криворучко О.В.		27.01.23	Веб-орієнтований додаток «Аніме Gift-merch» з просування тематичних аніме товарів	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		27.01.23		Р1	10	54
Гарант		Рзаєва С.Л.		27.01.23		Факультет інформаційних технологій		
Розробив		Дьяконов В.О.		27.01.23		4 курс, 7 група		
					Аналіз теоретичної частини розробки веб додатків			

поширенням інтернету та зростанням кількості користувачів, з'являється більше конкуренції на ринку інтернет-торгівлі. Тому, важливо мати переваги, які дозволять виокремитись на ринку та привернути більше покупців. Однією з таких конкурентних переваг є використання інноваційних технологій та підходів, які дозволяють забезпечити більш ефективне просування товарів та підвищити продажі. Саме тому, створення веб-орієнтованого додатку з просування тематичних аніме товарів є актуальним та перспективним напрямком розвитку інтернет-торгівлі. З метою дослідження актуальності теми, проведено аналіз відповідної літератури та ринку інтернет-торгівлі. За результатами аналізу було встановлено, що веб-орієнтований додаток з просування тематичних аніме товарів є досить новим та не дослідженим напрямком. Отже, актуальність теми "веб-орієнтований додаток з просування тематичних аніме товарів" обґрунтована та визначена на сьогоднішній день, що створює підґрунтя для подальшого дослідження даної теми. Дослідження може виявити потенційні можливості використання веб-орієнтованого додатку для просування тематичних аніме товарів, а також виявити можливі переваги та недоліки його використання. Крім того, створення веб-орієнтованого додатку з просування тематичних аніме товарів має потенціал вплинути на розвиток інтернет-торгівлі в цілому, оскільки цей додаток може бути використаний іншими інтернет-магазинами зі схожою тематикою. Тому дослідження даної теми може мати значущий внесок у розвиток інтернет-торгівлі та сприяти підвищенню конкурентоспроможності інтернет-магазинів з тематичним напрямком аніме. Загальний висновок полягає в тому, що тема "веб-орієнтований додаток з просування тематичних аніме товарів" є актуальною та перспективною для дослідження. Детальний аналіз теми дозволить виявити можливості використання веб-орієнтованого додатку для просування тематичних аніме товарів, а також визначити його переваги та недоліки в порівнянні з іншими методами просування. Далі будуть розглянуті деталізовані питання, які допоможуть вирішити дану проблему.

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

## 1.2. Технічне завдання

### 1. Загальні відомості

#### 1.1. Найменування системи

1.1.1. Повне найменування системи – Anime Gift-Merch

1.1.2. Скорочене найменування системи - AMG

#### 1.2. Планові терміни початку та закінчення робіт

Початок робіт - 01.01.2023

Закінчення робіт - 01.05.2023

#### 1.3. Порядок оформлення і пред'явлення результатів робіт

Результати робіт будуть оформлені у вигляді програмного коду, документації з поясненням алгоритмів, принципів роботи та інструкцій щодо встановлення, налаштування та користування додатком. Результати будуть пред'явлені замовнику в електронному вигляді, а також у паперовому вигляді за запитом замовника.

#### 1.4. Головний бенефіціар та потенційні користувачі системи

Головним бенефіціаром системи є студія, що володіє обладнанням та замовник. Потенційні користувачі системи - працівники студії, що мають права доступу до додатку та можуть змінювати налаштування обладнання, зберігати налаштування та отримувати статистику роботи обладнання.

### 2. Мета та призначення створення системи

#### 2.1. Призначення системи

Веб додаток призначений для просування аніме продукції.

#### 2.2. Мета створення системи

Метою створення системи є поліпшення просування тематичних аніме товарів в інтернеті. Додаток повинен бути зручним та простим у користуванні, містити всі необхідні функції та опції для її реалізації товарів.

### 3. Вимоги до системи

#### 3.1. Вимоги до системи в цілому

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

3.1.1. Вимоги до структури та функціонування системи, перелік підсистем

3.1.1.1. Вимоги до способів і засобів інформаційного обміну між компонентами системи

Компоненти системи повинні забезпечувати безперебійний обмін даними між клієнтами та сервером. Для цього використовуються протоколи забезпечення безпеки та стандарти взаємодії, такі як HTTP, HTTPS, TCP/IP. Для передачі даних можуть використовуватися формати JSON.

3.1.1.2. Вимоги до режимів функціонування системи

Система повинна працювати у режимі 24/7, без перебоїв та забезпечувати швидку відповідь на запити користувачів. Також, система повинна забезпечувати безпеку даних користувачів.

3.1.1.4. Вимоги до режимів управління системою

Система повинна мати зручний та простий інтерфейс управління, який дозволить оперативно налаштовувати та контролювати просування тематичних аніме товарів.

3.1.2. Показники призначення

3.1.2.1. Параметри, що характеризують ступінь відповідності системи призначенням:

- максимальна кількість користувачів, що можуть одночасно підключатися до системи;
- максимальний час відповіді системи на запит користувача;
- мінімальна швидкість передачі даних між клієнтами та сервером;
- ступінь зручності та простоти використання системи для користувачів з різним рівнем знань у користуванні інтернетом.

3.1.2.2. Вимоги до пристосовності системи до змін

- можливість додавання нових функцій та модулів до системи без необхідності її повного перепроєктування;

						Аркуш
					ДТЕУ 121 07-11.БР	13
Зм.	Аркуш	№ докум	Підпис	Дата		

3.1.2.3. Вимоги до збереження працездатності системи в різних ймовірних умовах

- можливість працювати в різних операційних системах та на різних пристроях;
- мінімальний час, необхідний для відновлення роботи системи після збою або відключення від мережі Інтернет.

3.1.3. Вимоги до надійності

3.1.3.2. Вимоги до надійності технічних засобів і програмного забезпечення

- Надійність програмного забезпечення повинна бути високою і забезпечувати нормальну роботу системи без відмов протягом тривалого часу;
- Надійність технічних засобів повинна бути також високою, забезпечувати стійкість до зовнішніх впливів і забезпечувати нормальну роботу системи протягом тривалого часу.

3.1.3.3. Вимоги до методів оцінки і контролю показників надійності на різних стадіях створення системи

- Проведення аналізу вимог до надійності та встановлення параметрів надійності системи в цілому і її складових;
- Проведення тестування на різних етапах створення системи з метою виявлення можливих відмов та помилок;
- Перевірка надійності системи в реальних умовах експлуатації.

3.1.4. Вимоги до ергономіки та технічної естетики

Додаток повинен мати зручний та логічно побудований інтерфейс для користувача, що дозволяє з легкістю виконувати всі необхідні функції та мінімізує його зусилля та час на операції. Колірна гамма та фонти повинні бути приємною для сприйняття користувача. Елементи інтерфейсу повинні мати відповідні характеристики для легкої взаємодії з ними.

3.1.6 Вимоги до захисту інформації від несанкціонованого доступу

						Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	

### 3.1.6.1. Вимоги до інформаційної безпеки

Система повинна забезпечувати захист інформації, яка обробляється та зберігається, від несанкціонованого доступу. Система повинна мати механізми аутентифікації та авторизації користувачів. Система повинна забезпечувати конфіденційність, цілісність та доступність даних відповідно до вимог законодавства та політик безпеки інформації компанії. Система повинна забезпечувати моніторинг захисту інформації, виявлення та реагування на можливі інциденти зі злому безпеки.

### 3.1.7. Вимоги до захисту від впливу зовнішніх факторів

Додаток повинен бути захищений від зловмисних дій суб`єктів, які можуть призвести до витоку конфіденційної інформації або пошкодження системи. Dodatok повинен бути захищений від шкідливого ПЗ. Dodatok повинен мати можливість забезпечення захисту від DDoS атак та інших типів кібератак.

### 3.2. Перелік підсистем системи (при наявності підсистем).

Серверна підсистема для зберігання і обробки даних. Клієнтська підсистема для керування просуванням аніме товарів . Підсистема для забезпечення безпеки доступу до даних користувачів. Підсистема для діагностування та моніторингу стану системи. Підсистема для забезпечення комунікації між компонентами системи. Підсистема для забезпечення технічного обслуговування та ремонту системи.

### 3.3. Вимоги до видів забезпечення

#### 3.3.2. Вимоги до інформаційного забезпечення

Система повинна забезпечувати безпечний обмін даними між клієнтом та сервером за допомогою захищеного каналу зв'язку. Система повинна мати можливість зберігати та обробляти дані торгівлі аніме товарами. Система повинна мати можливість забезпечувати авторизацію користувачів. Система повинна мати можливість здійснювати архівування та резервне копіювання

					ДТЕУ 121 07-11.БР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

даних, що забезпечить збереження інформації при можливих випадках втрати даних або несправності системи.

#### 4. Вимоги до програмного забезпечення

Додаток має бути написаний на мові програмування з можливістю підтримки множини одночасних підключень. Додаток повинен мати графічний інтерфейс користувача, який забезпечує зручне інтуїтивне користування.

#### 5. Вимоги до технічного забезпечення

Клієнтський додаток має бути сумісним з операційними системами Windows 10 та MacOS. Серверна частина повинна бути розміщена на сервері з належними характеристиками, забезпечуючи швидкість передачі даних та доступність сервера для клієнтів. Клієнтський чатина веб додатку має вимагати наявності мінімум 2 ГБ оперативної пам'яті. Рекомендована роздільна здатність екрану для роботи з додатком - не менше 1280x768 пікселів. Мінімальна швидкість Інтернет-з'єднання для плавної роботи додатка - 5 Мбіт/с.

### 1.3. Стан дослідження проблеми

Проблема просування товарів у сучасному світі є актуальною та важливою, оскільки залежно від ефективності просування залежить успіх продажу товарів. В інтернеті з'являється все більше та більше інтернет-магазинів з різноманітною тематикою, що зробило конкуренцію на ринку інтернет-торгівлі дуже високою. У зв'язку з цим, актуальність створення веб-орієнтованого додатку з просування тематичних аніме товарів зростає.

Дане дослідження має на меті розробку та впровадження в дію ефективного та зручного інструменту, який дозволить швидко та зручно знайти та купити потрібний товар в обраному магазині. Також, важливо враховувати пандемійну ситуацію, яка змінила спосіб життя багатьох людей та зробила їх

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16



більш залежними від онлайн-сервісів. Покупки в інтернеті втім стали одним з основних засобів придбання товарів, що сприяє популяризації веб-орієнтованих додатків та платформ. Отже, дослідження проблеми забезпечення зручного та ефективного просування тематичних аніме товарів у веб-просторі є актуальним і має великий потенціал в подальшому розвитку онлайн-шопінгу та розширенні ринку аніме-товарів. У світі аніме відбувається бурхливий розвиток, але реалізація товарів, на відміну від телевізійних та кінематографічних продуктів, майже повністю базується на віддаленій інтернет основі . З цього слідує, зростання попиту на створення веб-орієнтованих додатків та інших онлайн-сервісів, які дозволять швидко та зручно знайти та купити потрібні товари. На жаль, на сьогоднішній день, просування аніме-товарів відбувається переважно в ручному режимі, що може призвести до пропуску популярних продуктів та втрати покупців. В той же час, більшість існуючих платформ для продажу аніме-товарів мають деякі недоліки, такі як складний та неінтуїтивний інтерфейс, обмежену кількість товарів, відсутність можливості порівняння цін та відгуків про товари. Одним з можливих рішень проблеми є створення веб-орієнтованого додатку, який буде забезпечувати просування тематичних аніме товарів. Таким чином, ця дипломна робота має на меті провести детальний аналіз проблеми забезпечення зручного та ефективного просування тематичних аніме товарів у веб-просторі, визначити технічні вимоги до створення веб-орієнтованого додатку, а також запропонувати технічну реалізацію. За даними аналітичної компанії Crunchyroll, яка займається ліцензуванням та трансляцією аніме та манги в онлайн-форматі, кількість передплатників їхньої платформи зростає щорічно, а відтак зростає популярність аніме в цілому.

У 2020 році відзначився особливо високий ріст популярності аніме в світі, оскільки у зв'язку з пандемією COVID-19 багато людей залишилися вдома та знайшли для себе нову розвагу у вигляді перегляду онлайн аніме контенту.[1]

					ДТЕУ 121 07-11.БР	Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата		

Попри таку високу популярність аніме, на сьогоднішній день не існує цілеспрямованої онлайн-платформи для продажу аніме-товарів українськими виробниками та продавцями. Такі товари можна придбати лише на кількох інтернет-магазинах, де асортимент дуже обмежений та ціни значно перевищують їх реальну вартість. Таким чином, проблема полягає в відсутності онлайн-платформи з просуванням тематичних аніме-товарів в Україні, що у свою чергу призводить до втрати підприємствами та індивідуальними продавцями можливості збільшення прибутку та популяризації своїх товарів серед широкого кола споживачів. При цьому, у світі існує кілька онлайн-платформ, спеціалізованих на продажу аніме-товарів, такі як Crunchyroll, Right Stuf Anime, Funimation, J-List, Tokyo Otaku Mode та інші. Однак, доступність цих платформ для українських споживачів досить обмежена, або ж у них відсутній регіональний контент, що підходить для українського ринку.

Оскільки популярність аніме в Україні зростає, а відсутність цілеспрямованої онлайн-платформи для продажу таких товарів призводить до втрати прибутку для підприємств та індивідуальних продавців, то ця проблема потребує розв'язання. Також, важливо враховувати тенденції розвитку електронної комерції, зокрема зростаючу кількість покупок через Інтернет та збільшення популярності онлайн-платформ для продажу товарів.

Отже, розробка веб-орієнтованого додатку з просуванням тематичних аніме-товарів може бути актуальною та перспективною для українського ринку. Таким чином, актуальність теми полягає в необхідності створення онлайн-платформи для продажу тематичних аніме-товарів в Україні, що дозволить підприємствам та індивідуальним продавцям збільшити свій прибуток та популяризувати свої товари серед широкого кола споживачів, а також враховувати зростаючу популярність онлайн-платформ для продажу товарів та зміни в споживчому підході до покупок через інтернет. Використання веб-орієнтованих додатків для просування тематичних товарів

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

на сьогоднішній день є досить перспективним та актуальним напрямком маркетингу. В той же час, велика кількість підходів до розробки таких додатків та їхньої оптимізації дозволяє здійснювати успішну рекламну кампанію. Однак, існують проблеми, які потрібно вирішувати в розробці та використанні веб-орієнтованих додатків для просування тематичних товарів. Однією з таких проблем є залежність від пошукових систем. Більшість користувачів Інтернету шукає інформацію про товари в пошукових системах, тому для просування тематичних товарів необхідно оптимізувати сайт або додаток відповідно до вимог пошукових систем. Це може включати у себе написання унікальних текстів, використання ключових слів та інших технік оптимізації, що дозволяють підняти рейтинг сайту в пошукових системах. Іншою проблемою є відмова користувачів від використання різноманітних рекламних програм, таких як блокувальники реклами, які заблоковують відображення реклами на веб-сторінках. Це може привести до того, що рекламна кампанія не буде досягати своєї мети. Для розв'язання цієї проблеми необхідно знайти альтернативні методи просування товарів, які не будуть викликати негативних емоцій. Отже, основні проблеми, що можуть виникнути при розробці веб-орієнтованого додатку для просування тематичних аніме товарів, включають наступне:

- Складність вибору ефективної стратегії маркетингу для аудиторії, яка цікавиться аніме товарами, оскільки ця аудиторія є досить специфічною і має свої власні потреби та пристрасті.
- Недостатня кількість наявних джерел для збору та аналізу даних про попит та пропозицію на ринку аніме товарів.
- Велика кількість конкурентів на ринку, які також активно використовують веб-орієнтовані додатки для просування своїх товарів.
- Технічні труднощі, пов'язані з розробкою та підтримкою веб-орієнтованого додатку, зокрема, забезпечення його безпеки, швидкості роботи та зручного інтерфейсу

						Аркуш
					ДТЕУ 121 07-11.БР	19
Зм.	Аркуш	№ докум	Підпис	Дата		

- Потреба у постійній оновленні контенту та розширенні асортименту товарів, що вимагає залучення великої кількості творчих і технічних ресурсів.

Враховуючи ці проблеми, можна стверджувати, що розробка веб-орієнтованого додатку для просування тематичних аніме товарів є актуальною та важливою задачею, яка може принести значний прибуток та вести до розвитку цієї галузі. Зважаючи на вищенаведене, можна зробити висновок, що розвиток сфери електронної комерції та інтернет-маркетингу стає все більш актуальним і необхідним у сучасному світі. Застосування веб-орієнтованих додатків для просування товарів та послуг набуває все більшої популярності серед підприємців, що прагнуть підвищити ефективність свого бізнесу та залучити більше клієнтів. У контексті тематики дослідження, важливим є розроблення веб-орієнтованого додатку для просування тематичних аніме товарів. Оскільки аніме культура стає все популярнішою у світі, багато підприємців бачать в цьому перспективний напрямок для розвитку свого бізнесу. Проте, існуючі рішення для просування товарів не завжди задовольняють потреби аудиторії, тому розробка нового додатку є вигідним та перспективним рішенням для підприємства. Отже, актуальність теми полягає в необхідності розробки нового веб-орієнтованого додатку для просування тематичних аніме товарів, що забезпечить підвищення ефективності бізнесу та реалізація потреб клієнтів у цій сфері.

Технічне завдання описує основні вимоги до створення веб-орієнтованого додатку для просування тематичних аніме товарів. Цей додаток повинен дозволяти користувачам швидко знаходити та придбати аніме товари.

Опис продукту: Веб-орієнтований додаток для просування тематичних аніме товарів повинен мати наступні функції:

- Реєстрація та вхід користувачів. Користувачі повинні мати змогу створити обліковий запис і входити в систему.

						Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	

- Пошук товарів. Користувачі повинні мати можливість шукати аніме товари за ключовими словами, категоріями, та іншими параметрами.
- Порівняння товарів. Користувачі повинні мати можливість порівнювати різні аніме товари за ціною, описом, рейтингом та іншими параметрами.
- Додавання товарів до списку бажань. Користувачі повинні мати можливість додавати товари до свого списку бажань та отримувати сповіщення про знижки на ці товари.
- Покупка товарів. Користувачі повинні мати можливість здійснювати покупки аніме товарів через забезпечені платіжні системи.
- Аналітика. Адміністратори повинні мати можливість переглядати статистику використання додатку, включаючи кількість користувачів, популярність товарів, та ін.

Маркетингові вимоги:

- Розробити зручний та простий інтерфейс користувача, що відповідає потребам цільової аудиторії.
- Забезпечити можливість відслідковування поточного стану замовлень. Забезпечити можливість відстеження рейтингу популярних аніме.
- Забезпечити можливість оцінювання аніме товарами і послугами.
- Забезпечити можливість коментування та відгукуваня на товари.
- Розробити механізм рекомендацій товарів на основі аналізу покупок користувачів.

Технічні вимоги:

- Додаток повинен забезпечувати збереження даних користувачів на серверах.
- Додаток повинен підтримувати можливість оплати замовлень за допомогою банківських карт.
- Додаток повинен бути захищений від шкідливих впливів, включаючи злом та витік інформації.

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		21

- Додаток повинен забезпечувати швидке завантаження сторінок і функцій.
- Додаток повинен бути орієнтований на роботу з великою кількістю даних.
- Додаток повинен забезпечувати можливість розширення та модифікації функціоналу у майбутньому.

Форма даних: Для зберігання даних про користувачів, продукти, замовлення та інші потрібні дані використовуватимуться бази даних PostgreSQL. Для роботи з базами даних буде використовуватися мова запитів SQL.

Передбачувані обсяги даних: очікується, що кількість користувачів, що зареєструються в системі, буде зростати з часом. На перших етапах розробки передбачається робота зі 100 користувачами, з можливістю масштабування на більші обсяги даних. Очікується, що кожен користувач буде зберігати інформацію про свої покупки, замовлення, відгуки і рейтинги.

Вимоги до надійності: система повинна бути надійною і безпечною, щоб запобігти витоку конфіденційної інформації, такої як паролі користувачів, особисті дані, кредитні картки та інші деталі. Для цього будуть застосовуватися заходи безпеки, такі як шифрування та хешування паролів користувачів, захист баз даних від зламу, резервне копіювання даних тощо.

Вимоги до швидкодії: система повинна працювати швидко і ефективно, щоб забезпечити користувачам максимально комфортний і зручний досвід взаємодії з додатком. Будуть застосовуватися оптимізації швидкодії баз даних, кешування інформації, оптимізація роботи серверів та інші заходи.

Вимоги до масштабованості: система повинна бути здатною до масштабування, щоб забезпечити роботу з більшим обсягом даних.

Мета проекту: розробити веб-орієнтований додаток для просування та продажу тематичних аніме товарів, який забезпечить користувачам зручний і простий інтерфейс, з великою кількістю функцій та можливостей.

						Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	

Тип додатку: веб-орієнтований

Основна функціональність:

- Перегляд асортименту товарів
- Додавання товарів у кошик
- Заовлення товарів
- Керування замовленнями
- Оцінювання та коментування товарів
- Пошук товарів за ключовими словами, жанрами, авторами тощо.
- Реєстрація користувачів та авторизація в системі

Вимоги до функціональності:

- Додавання нових товарів в базу даних
- Можливість редагування та видалення товарів
- Організація оплати замовлень
- Зберігання історії замовлень
- Можливість відстеження статусу замовлень
- Реалізація пошуку товарів за різними параметрами
- Реєстрація користувачів та авторизація в системі
- Можливість додавання опису товарів, зображення, відгуки користувачів
- Захист від зламу та зловживання

Технічне завдання на розробку веб-додатку для просування тематичної аніме продукції містить такі вимоги:

Платформа розробки: веб-додаток буде розроблено з використанням сучасних веб-технологій, таких як HTML5, CSS3, JavaScript, React Js, Express.

Програма також використовуватиме реляційну систему керування базами даних PostgreSQL для зберігання даних.

Дизайн та інтерфейс користувача: додаток матиме адаптивний дизайн, який буде сумісний із настільними комп'ютерами, планшетами та мобільними

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		23

пристроями. Інтерфейс користувача буде зручним та інтуїтивно зрозумілим із зручною навігацією.

Каталог продуктів: програма матиме каталог продуктів, яким керуватиме адміністратор. Каталог міститиме інформацію про товари, включаючи назву продукту, опис, зображення, ціну та категорію.

Керування обліковими записами користувачів: програма матиме систему керування обліковими записами користувачів, яка дозволить користувачам реєструватися та входити в програму. Зареєстровані користувачі зможуть переглядати історію своїх покупок і керувати особистою інформацією. Кошик для покупок: програма матиме функцію кошика для покупок, яка дозволить користувачам додавати продукти до свого кошика та переходити до оформлення замовлення для завершення покупки.

Інтеграція платіжного шлюзу: додаток буде інтегровано з платіжним шлюзом, щоб користувачі могли безпечно здійснювати онлайн-платежі.

Функції пошуку та фільтрації: програма матиме функцію пошуку та фільтрації, яка дозволить користувачам шукати продукти на основі конкретних критеріїв, таких як категорія, діапазон цін і назва продукту.

Безпека та конфіденційність: програма матиме функції безпеки та конфіденційності, які забезпечать безпеку та конфіденційність інформації користувача, включаючи шифрування SSL, безпечну обробку платежів.

Загалом технічне завдання гарантує, що веб-додаток для просування тематичних аніме-продуктів розроблено відповідно до найвищих стандартів якості, з акцентом на зручність використання, функціональність та безпеку.

Терміни виконання: Розробка додатку повинна бути завершена протягом 6 місяців.

#### **1.4. Висновки по першому розділу**

У першому розділі дипломної роботи були розглянуті вимоги до функцій веб-додатку. В результаті аналізу предметної області і вивчення потреб користувачів, було сформульовано основні функції, які має виконувати

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		24



додаток. Ці вимоги включають функції реєстрації та авторизації користувачів, перегляду та пошуку товарів, додавання товарів до кошика, оформлення замовлення та інші. Результати аналізу вимог до функцій становлять основу для подальшої реалізації та розробки додатку. Вони визначають функціональні можливості, які будуть доступні користувачам і визначають спосіб взаємодії з додатком.

Загальний аналіз вимог до функцій допомагає зрозуміти потреби користувачів та створити ефективний та зручний для використання веб-додаток. Детальна специфікація функціональних вимог забезпечує базовий фундамент для подальшого проектування та реалізації додатку.

З урахуванням вимог до функцій, веб-орієнтований додаток з продажу взуття буде розроблено з використанням сучасних технологій та інструментів, що дозволять забезпечити зручне та ефективне користувацьквq досвід.



					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

## РОЗДІЛ 2 АНАЛІЗ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКА ДЛЯ ПРОСУВАННЯ ТЕМАТИЧНИХ АНІМЕ ТОВАРІВ

### 2.1. Огляд сучасних технологій розробки веб-додатків

У сфері розробки веб-додатків існує велика кількість технологій та фреймворків, які використовуються розробниками для створення потужних та функціональних додатків. Далі буде розглянуто деякі найбільш популярні технології та їх роль у розробці. Існує декілька популярних веб-фреймворків та бібліотек таких як: React, Angular, Vue.

**React** - це JavaScript-бібліотека для розробки інтерфейсу користувача. Вона була розроблена компанією Facebook і використовується для створення потужних та масштабованих веб-додатків. React дозволяє розробляти компоненти інтерфейсу користувача, які можна багаторазово використовувати і легко підтримувати. Основна ідея React полягає в тому, щоб створювати веб-модулі, які матимуть змогу оновлюватися незалежно один від одного. Кожен компонент є деяким шматком інтерфейсу користувача, який може мати внутрішній стан і логіку. React ефективно керує оновленням лише тих компонентів, які змінилися, що призводить до покращення продуктивності. React використовує віртуальний DOM (Document Object Model), щоб ефективно оновлювати лише необхідні елементи інтерфейсу користувача. Віртуальний DOM являє собою внутрішнє уявлення реального DOM-дерева, яке React порівнює з поточним станом і обчислює мінімальну кількість змін, необхідних для оновлення інтерфейсу користувача. Це дозволяє уникнути непотрібних операцій та підвищити продуктивність програми. Одна з ключових особливостей React – це односпрямований потік даних. Дані React зазвичай передаються зверху вниз по ієрархії компонентів. Якщо дані в компоненті змінюються, React оновлює лише цей компонент та його нащадків, що спрощує відстеження стану програми. React також надає безліч додаткових

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

інструментів та бібліотек для розробки веб-додатків, таких як React Router для керування маршрутизацією, Redux для керування станом програми та безліч інших сторонніх бібліотек, що дозволяють розширити функціональність React. В цілому, React є потужним і популярним інструментом для розробки веб-додатків, який полегшує створення масштабованих і ефективних інтерфейсів користувача. [2]

**Angular** - це повноцінний веб-фреймворк, розроблений та підтримуваний компанією Google. Він надає інструменти та посібники для розробки масштабованих та високопродуктивних веб-додатків. Angular використовує мову програмування TypeScript, яка додає статичну типізацію та інші можливості JavaScript. В основі Angular лежить концепція компонентів. Компоненти - це будівельні блоки веб-додатків, які включають HTML-шаблон, стилі і логіку. Компоненти можуть бути вкладеними та утворювати деревоподібну структуру, де кожен компонент може мати свій внутрішній стан та обробляти події. Angular надає механізми обробки даних, синхронізації з DOM і управління життєвим циклом компонентів. Angular також забезпечує потужну систему ін'єкції залежностей. Це дозволяє ефективно управляти залежностями та створювати слабопов'язані компоненти. Ін'єкція залежностей у Angular допомагає покращити тестованість та перевикористовуваність коду. Один із ключових елементів Angular - це шаблонізація. Шаблони Angular дозволяють створювати динамічні та інтерактивні інтерфейси користувача з використанням прив'язки даних і директив. Прив'язка даних дозволяє автоматично оновлювати інтерфейс користувача при зміні даних, що спрощує розробку та підтримку додатків. Angular також пропонує маршрутизацію, яка дозволяє керувати навігацією всередині програми. За допомогою модуля маршрутизації Angular можна визначити шляхи та налаштування маршрутизації, що забезпечує навігацію між різними уявленнями та компонентами програми. На додаток до основних функціональних можливостей, Angular має широке співтовариство та екосистему, які надають

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		27

безліч розширень і бібліотек, таких як Angular Material для створення стильних інтерфейсів користувача, NgRx для управління станом програми та інші. Загалом, Angular надає розробникам інструменти та структури для створення складних та масштабованих веб-додатків.[3]

Express - є легковагим та гнучким веб-фреймворком для розробки додатків на основі Node.js. Він дозволяє швидко створювати веб-сервери та API, спрощуючи процес роботи з HTTP-запитами, маршрутизацією та обробкою запитів та відповідей. Основними перевагами Express є його мінімалістичний дизайн та велика кількість плагінів, які дозволяють розширювати його можливості. Фреймворк пропонує простий API для створення сервера та визначення маршрутів. Він дозволяє обробляти HTTP-запити GET, POST, PUT та DELETE, а також забезпечує можливість використання параметрів шляху та запити. Express також підтримує шаблонізацію, що дозволяє генерувати HTML-сторінки за допомогою шаблонів, таких як Pug, EJS та інших. Це дозволяє розробникам створювати динамічні веб-сторінки та додатки. Ще одною перевагою Express є його розширюваність та підтримка сторонніми модулями. Через велику кількість плагінів, які створені спільнотою, розробники можуть легко додавати нові функції до своїх додатків. Наприклад, існують плагіни для автентифікації, обробки форм, роботи з базами даних та інших завдань. В цілому, Express є потужним та популярним веб-фреймворком, який дозволяє розробникам швидко створювати веб-додатки на Node.js. Його простий синтаксис та розширюваність роблять його відмінним вибором для будь-якого проекту, де потрібно побудувати ефективний та масштабований веб-сервер.[4]

## 2.2. Архітектура веб-додатків

Архітектура веб-додатків - це організація компонентів, модулів та взаємодія між ними для створення функціонального та ефективного веб-додатку. Нижче розглянуто основні складові архітектури веб-додатків.

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

Клієнтська частина (Client-Side): Клієнтські технології: HTML, CSS, JavaScript і бібліотеки/фреймворки, такі як React, Angular або Vue.js, використовуються для створення інтерфейсу користувача (UI) та взаємодії з ним. HTML відповідає за структуру сторінки, CSS - за її вигляд, а JavaScript - за динамічну функціональність.

Серверна частина (Server-Side):

Веб-сервер: Це програмне забезпечення, яке обробляє запити від клієнта та надсилає відповіді. Популярні веб-сервери включають Apache, Nginx, Microsoft IIS тощо.

Серверні мови програмування: PHP, Python, Ruby, Java, C# і Node.js - це лише кілька прикладів мов програмування, які використовуються для написання серверної логіки веб-додатків. Вони дозволяють обробляти запити від клієнта, взаємодіяти з базою даних та надсилати відповіді.

Фреймворки, такі як Django, Ruby on Rails, Laravel або ASP.NET, надають структуру та набір інструментів для швидкої розробки веб-додатків. Вони спрощують процес створення маршрутів, роботи з базами даних, аутентифікації, безпеки тощо.

**Архітектура** проекту також включає в себе проектування Uml діаграм, таких як use case або class, за допомогою яких можна спрогнозувати розробку веб-додатка, чи взагалі будь-якого додатка, автором роботи було розроблено дві діаграми за допомогою яких можна детальніше ознайомитись з сутностями веб-додатку.

Use case діаграма показує взаємодію між користувачем та веб-додатком, та допомагає виділити орієнтири подальшої розробки яка буде описана більш детально у третьому розділі.

						Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	

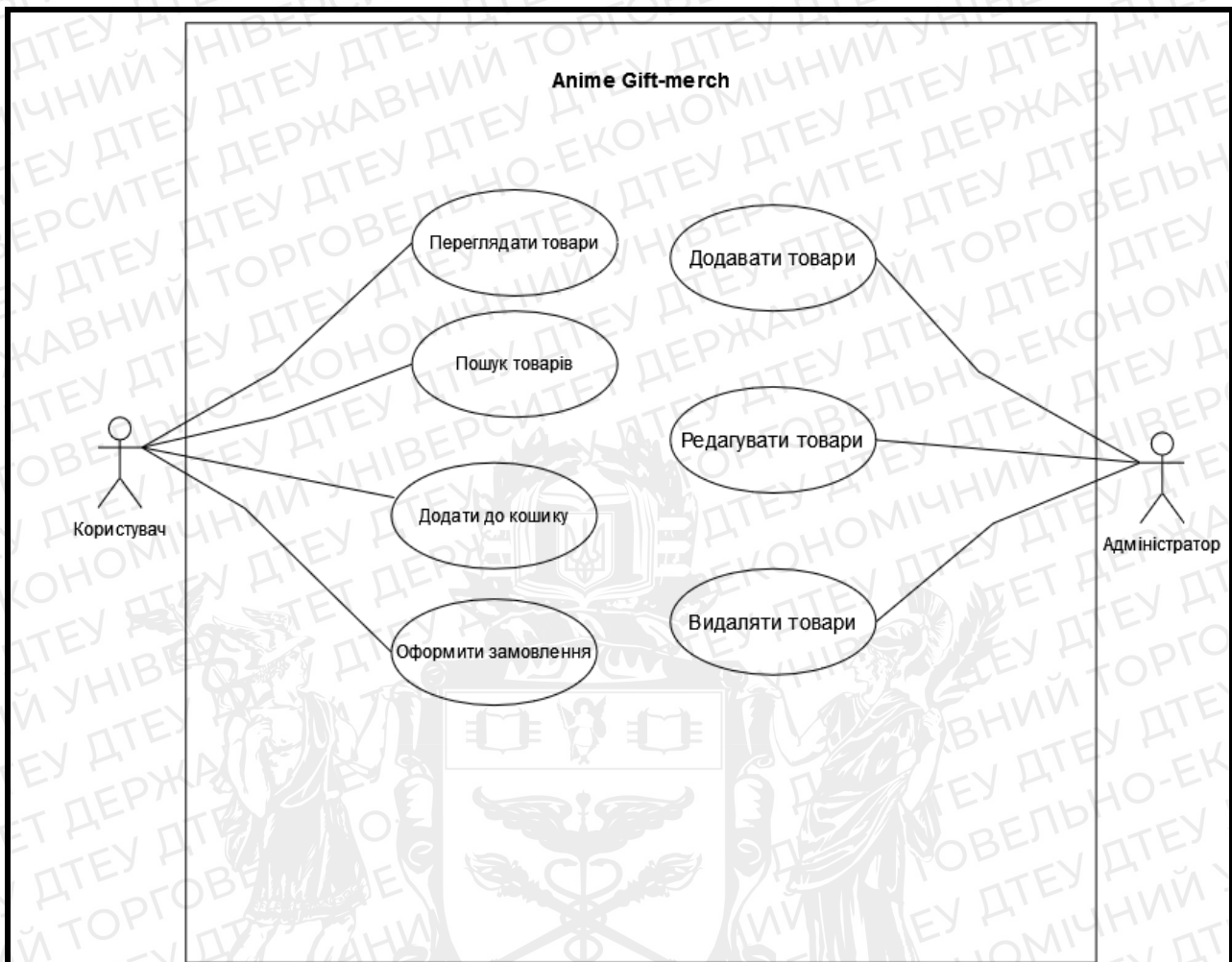


Рис.2.1 Use case діаграма веб-додатку

Джерело: побудовано автором

Діаграма класів – допомагає ознайомитись з взаємодією сутностей між собою, а також виділити порядок реалізації сутностей і їх взаємозв'язок між собою.

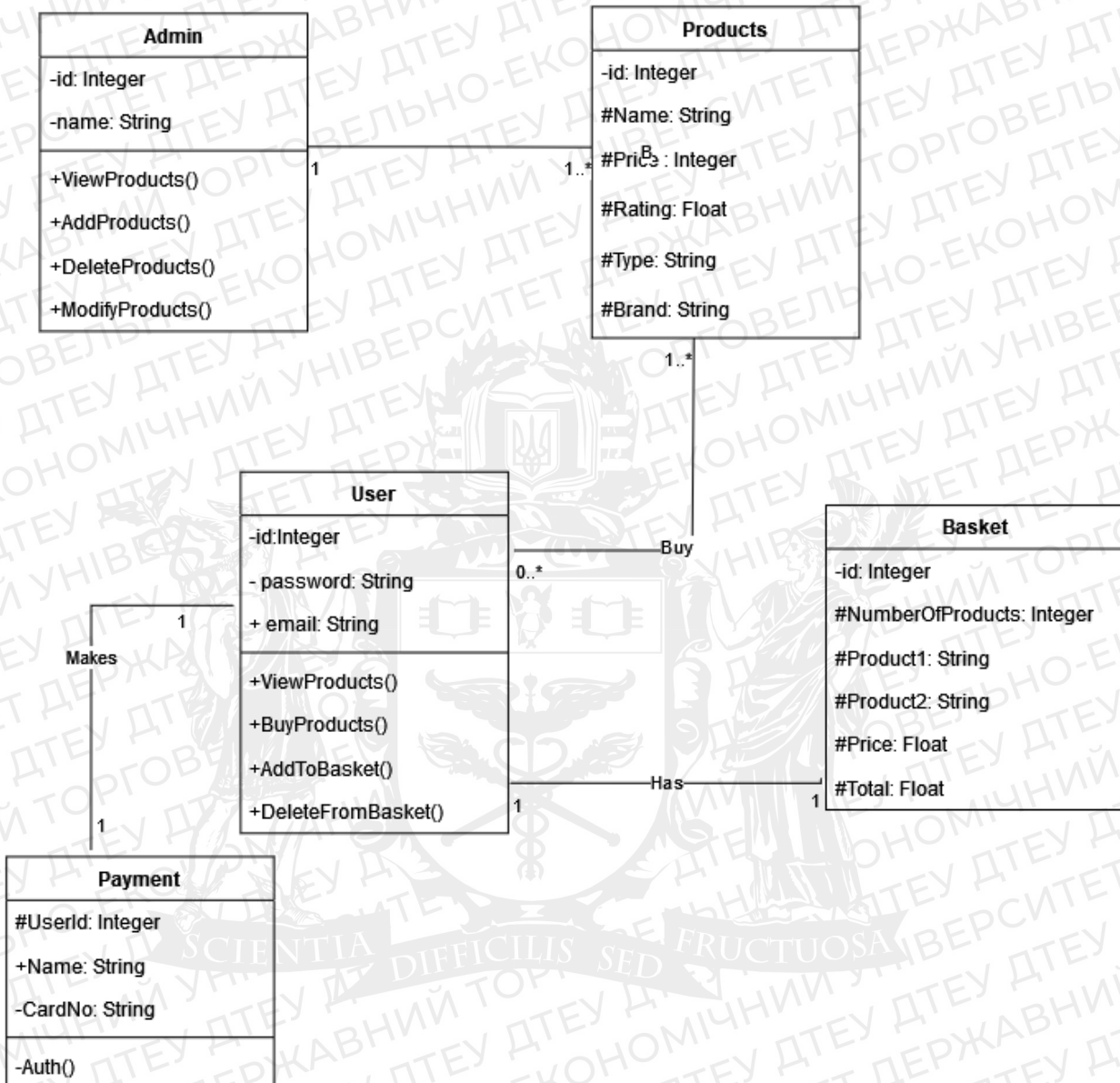


Рис.2.2 діаграма класів веб-додатку

Джерело: побудовано автором

**Бази даних:**

Реляційні бази даних: Наприклад, MySQL, PostgreSQL, Oracle, SQL Server - це системи управління базами даних (СУБД), які забезпечують зберігання даних у вигляді таблиць зі зв'язками між ними. Вони використовуються для збереження та організації даних веб-додатку.

Нереляційні бази даних: Наприклад, MongoDB, CouchDB - це бази даних, які використовують іншу модель зберігання даних, таку як документи, ключ-значення, графи тощо. Вони зазвичай використовуються для зберігання невстановленої або змінної структури даних.

Взаємодія між клієнтом та сервером:

Протоколи: HTTP (Hypertext Transfer Protocol) є основним протоколом, який використовується для комунікації між клієнтом та сервером у веб-додатках. Взаємодія відбувається шляхом обміну HTTP-запитами та відповідями.

API (Application Programming Interface): API визначає набір правил та протоколів, за допомогою яких різні компоненти програмного забезпечення можуть взаємодіяти між собою. Веб-додатки можуть надавати публічні API для взаємодії з іншими додатками або послугами. Така архітектура веб-додатків забезпечує поділ обов'язків та організацію компонентів, що дозволяє розробникам ефективно створювати складні та функціональні веб-додатки. Вибір конкретної архітектури залежить від потреб проекту, масштабу додатку та вимог до його функціональності.

### 2.3. Бази даних для веб-додатків

Бази даних є важливою складовою веб-додатків, оскільки вони забезпечують зберігання, організацію та доступ до даних. У розробці веб-додатків використовуються різні типи баз даних, залежно від вимог проекту. Розглянемо деякі типи баз даних для веб-додатків:

1. Реляційні бази даних (Relational databases): Реляційні бази даних є одним з найпоширеніших типів баз даних для веб-додатків. Вони базуються на реляційній моделі даних, де дані зберігаються у вигляді таблиць зі зв'язками між ними. Деякі популярні реляційні бази даних включають MySQL, PostgreSQL, Oracle та SQL Server. Реляційні бази даних



дозволяють створювати структуровані схеми даних, забезпечують цілісність даних та підтримують потужну мову запитів SQL для роботи з даними.

2. **Нереляційні бази даних (Non-relational databases):** Нереляційні бази даних, також відомі як NoSQL (Not Only SQL), використовують іншу модель зберігання даних, відмінну від традиційної реляційної моделі. Ці бази даних підходять для зберігання невстановленої або змінної структури даних. Деякі типи нереляційних баз даних включають документові бази даних (наприклад, MongoDB), ключ-значення бази даних (наприклад, Redis), стовпчикові бази даних (наприклад, Cassandra) та графові бази даних (наприклад, Neo4j). Нереляційні бази даних підтримують горизонтальне масштабування та забезпечують високу швидкодію при обробці великих обсягів даних. Інші типи баз даних: Крім реляційних та нереляційних баз даних, існують інші типи баз даних, які можуть бути використані в веб-додатках в залежності від потреб проекту. Наприклад:

- **Інтернет-пам'ять (In-Memory Databases):** Ці бази даних зберігають дані безпосередньо в оперативній пам'яті, що забезпечує швидкий доступ до них. Вони особливо ефективні для додатків з високою швидкістю, які вимагають миттєвого доступу до даних.
- **Географічні бази даних:** Ці бази даних спеціалізовані на зберігання та обробку географічних даних, таких як картографічні дані, координати місць тощо.
- **Часові бази даних:** Ці бази даних призначені для зберігання та аналізу даних, пов'язаних зі змінними в часі, такими як журнали, сенсорні дані, історичні дані тощо.

Вибір типу бази даних залежить від потреб веб-додатку, обсягу даних, швидкодії, потреб в масштабуванні та інших факторів. При розробці веб-додатку важливо зрозуміти характеристики різних баз даних та їх

відповідність вимогам проекту, тому було розроблено uml-діаграму реляційної бази даних.

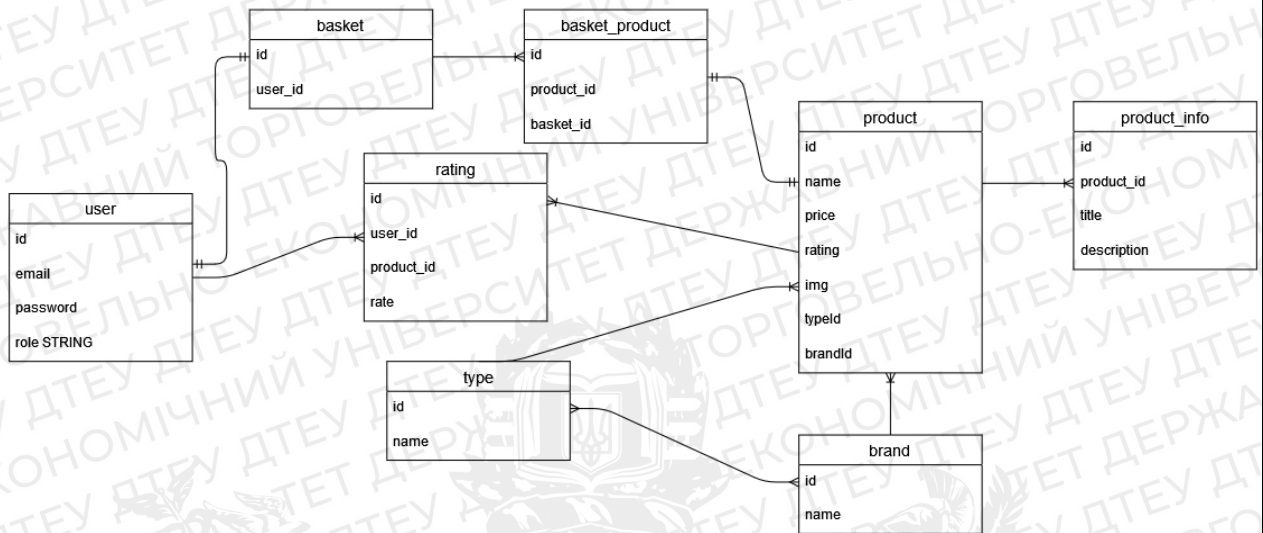


Рис.2.3 Логічна модель бази даних

Джерело: Побудовано автором

## 2.4. Вибір технологій для розробки веб-додатка

Вибір технологій для розробки веб-додатка є важливим етапом, який впливає на успіх проекту. Існує широкий спектр технологій, фреймворків та інструментів, які можна використовувати для розробки веб-додатків. При виборі технологій слід враховувати такі фактори:

1. Вимоги проекту: Розуміння вимог веб-додатка є вирішальним фактором. Потрібно врахувати функціональні вимоги, необхідність інтеграції з іншими системами, масштабованість, безпеку, швидкодію та інші параметри проекту.
2. Тип додатка: Розробка веб-додатків може бути фронтенд-орієнтованою, бекенд-орієнтованою або повномасштабною. Мови програмування: для фронтенд-розробки використовуються HTML, CSS та JavaScript. Для бекенд-розробки можуть використовуватись мови, такі як Python, Ruby, PHP, Java, JavaScript, TypeScript.

3. Фреймворки: спрощують розробку веб-додатків, надаючи готові рішення та інструменти для роботи зі стандартними завданнями. Наприклад, для розробки бекенду можна використовувати фреймворки Django, Ruby on Rails, Laravel, ASP.NET або Express.js. Для фронтенду популярні фреймворки і бібліотеки, такі як React, Angular або Vue.js.
4. Бази даних: Реляційні бази даних, такі як MySQL, PostgreSQL або SQL Server, забезпечують структуроване зберігання даних. Нереляційні бази даних, наприклад MongoDB або Redis, підходять для зберігання невстановленої структури даних.

При виборі технологій важливо збалансувати вимоги проекту, знання та досвід, доступні ресурси та час, які можна вкласти в навчання нових технологій.

## 2.5. Забезпечення безпеки веб-додатків

Забезпечення безпеки веб-додатків є критично важливим аспектом при їх розробці. Незахищені веб-додатки можуть бути піддані атакам, які можуть призвести до витоку конфіденційної інформації, порушення цілісності даних або втрати доступу до системи. Ось деякі аспекти, які слід враховувати при забезпеченні безпеки веб-додатків:

**Аутентифікація і авторизація:** Забезпечення механізмів аутентифікації (підтвердження ідентичності користувача) та авторизації (контроль доступу до ресурсів) є основою безпеки веб-додатка. Використовуйте надійні методи аутентифікації, такі як паролі з хешуванням, двофакторна аутентифікація або використання зовнішніх сервісів аутентифікації. Переконайтеся, що авторизація відбувається на всіх рівнях системи, включаючи фронтенд, бекенд та доступ до бази даних.

**Захист від вразливостей веб-додатків:** При розробці веб-додатка слід уникати вразливостей, таких як введення зламу SQL, міжсайтовий скриптінг (XSS), перекладачі та інші. Потрібно використовувати методи, такі як

					<i>ДТЕУ 121 07-11.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		35

валідація введення, захист від ін'єкцій, обмеження привілеїв, ескейпінг HTML та інші, щоб забезпечити безпеку вашого додатка. Захист від атак на сесії: Забезпечення безпеки сесій користувача є важливим аспектом. Слід використовувати механізми генерації унікальних сесійних ідентифікаторів, встановлювати обмеження на тривалість сесій та перевіряти правильність прав доступу до ресурсів під час кожного запиту.

Шифрування даних: Важливо шифрувати конфіденційні дані, що передаються між клієнтом та сервером або зберігаються в базі даних. Використовувати протоколи шифрування, такі як HTTPS, для захисту комунікації між клієнтом та сервером. Для зберігання паролів та інших конфіденційних даних використовуються алгоритми хешування. Загальноприйнятим підходом до забезпечення безпеки веб-додатків є принцип "захищеність за замовчуванням" (security by default). Це означає, веб-додаток повинен бути спроектованим та розробленим з урахуванням безпеки вже на початкових етапах, використовуючи найкращі практики та стандарти безпеки.

## 2.6 Взаємодія зі сторонніми сервісами

Взаємодія зі сторонніми сервісами є важливим аспектом розробки веб-додатків, оскільки це дозволяє розширити функціональність додатка та отримати доступ до додаткових ресурсів. Основні аспекти взаємодії зі сторонніми сервісами включають наступне:

API-інтеграція: Багато сторонніх сервісів надають API (інтерфейс програмування додатків), що дозволяє взаємодіяти з ними і обмінюватися даними. Залежно від потреб веб-додатка, можливі варіанти взаємодії можуть включати отримання даних, створення, оновлення або видалення ресурсів.

Аутентифікація та авторизація: При взаємодії зі сторонніми сервісами може знадобитися аутентифікація для отримання доступу до конфіденційних даних

					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		36

або забезпечення безпеки. Деякі сервіси можуть вимагати встановлення ключів API, токенів або інших форм ідентифікації. Крім того, слід враховувати права доступу та обмеження, які накладаються сторонніми сервісами, і забезпечувати правильну авторизацію для взаємодії з ними. Обробка помилок та винятків: При взаємодії зі сторонніми сервісами необхідно передбачити обробку помилок та винятків. Помилки можуть включати некоректні запити, відповіді з помилковими статусами або відсутність доступу до сервісу. Важливо налагодити механізми перехоплення та обробки помилок, а також визначити стратегію повторної спроби взаємодії в разі недоступності сервісу.

Оптимізація швидкості та масштабованості: При взаємодії зі сторонніми сервісами слід забезпечити оптимальну швидкість та масштабованість додатка. Це може включати кешування даних, асинхронну обробку запитів, розподілену архітектуру або використання служб кешування і черг. Враховуючи ці аспекти, вибір сторонніх сервісів та правильна інтеграція їх у веб-додаток можуть значно покращити його функціональність, продуктивність та користувацький досвід.

## 2.6. Висновки по другому розділу

У другому розділі дипломної роботи було розглянуто аналіз предмета дослідження, а саме технології розробки веб-додатків. Були розглянуті різні аспекти, які мають важливе значення при виборі технологій для розробки веб-додатків.

У першому підрозділі були представлені популярні веб-фреймворки, такі як Django, Ruby on Rails, Laravel, ASP.NET і React. Кожен з них має свої переваги і недоліки, і вибір залежить від потреб проекту та навичок розробника. Крім того, було розглянуто роль клієнтських технологій (HTML5, CSS3, JavaScript, AJAX) і серверних технологій (PHP, Python, Ruby, Node.js) у розробці веб-додатків.

						Аркуш
					ДТЕУ 121 07-11.БР	37
Зм.	Аркуш	№ докум	Підпис	Дата		

Другий підрозділ присвячений архітектурі веб-додатків. Було розглянуто три рівні архітектури: клієнтський рівень, серверний рівень і рівень бази даних. Кожен рівень виконує певні функції і взаємодіє з іншими рівнями для забезпечення роботи веб-додатка. Описано популярні архітектурні шаблони, такі клієнт-серверна архітектура.

Третій підрозділ присвячений базам даних для веб-додатків. Було розглянуто реляційні бази даних та нереляційні бази даних (NoSQL). Для реляційних баз даних розглянуто SQL-мову та важливі концепції, такі як таблиці, запити, індекси та транзакції. Для NoSQL баз даних розглянуто типи документ-орієнтованих, ключ-значення та стовпчикових баз даних.

Висновки другого розділу вказують на важливість вибору правильних технологій розробки веб-додатків, архітектурного підходу та бази даних. Коректний вибір технологій та архітектури впливає на продуктивність, масштабованість, безпеку та підтримку веб-додатків. Крім того, розуміння особливостей роботи з базами даних дозволяє забезпечити ефективне управління даними в веб-додатках

						Аркуш
					ДТЕУ 121 07-11.БР	38
Зм.	Аркуш	№ докум	Підпис	Дата		

### РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКА ДЛЯ ПРОСУВАННЯ ТЕМАТИЧНИХ АНІМЕ ТОВАРІВ

#### 3.1. Вибір архітектури веб-додатка з просування аніме товарів

Вибір архітектури веб-додатка є критичним етапом розробки, оскільки від цього залежить його функціональність, масштабованість, ефективність та зручність у підтримці. При розгляді варіантів архітектур для розробки веб-додатка з просування тематичних аніме товарів, розглянемо наступні підходи:

**Монолітна архітектура:** Це традиційний підхід, де весь додаток розгортається як єдиний модуль. У цьому випадку усі компоненти додатка, такі як користувацький інтерфейс, бізнес-логіка та база даних, знаходяться в одному кодовому базі. Монолітна архітектура проста у розробці та деплої, але може бути обмежена в масштабованості та можливості розподіленого розвитку.[5]

**Клієнт-серверна архітектура:** У цьому підході клієнтська частина (користувацький інтерфейс) та серверна частина (бізнес-логіка, база даних) розташовуються на різних фізичних серверах. Клієнтська частина взаємодіє з сервером через мережу. Цей підхід дозволяє розділити відповідальності та полегшує масштабування.[6]

**Мікросервісна архітектура:** У цьому підході додаток розбивається на невеликі незалежні сервіси, кожен з яких виконує окрему функціональну одиницю. Кожен сервіс може бути розгорнутий та масштабований незалежно. Взаємодія між сервісами відбувається через API. Мікросервісна архітектура

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 07-11.БР			
Зав. каф.	Криворучко О.В.			14.04.23	Веб-орієнтований додаток «Аніме Gift-merch» з просування тематичних аніме товарів	Стадія	Аркуш	Аркушів
Керівник	Десятко А.М.			14.04.23		РЗ	39	54
Гарант	Рзаєва С.Л.			14.04.23		Факультет інформаційних технологій		
Розробив	Дьяконов В.О.			14.04.23		4 курс, 7 група		

дозволяє забезпечити гнучкість, масштабованість та розподілений розвиток, але вимагає складнішого управління та підтримки.[7]

При виборі конкретної архітектури для веб-додатка з просування тематичних аніме товарів необхідно враховувати наступні фактори:

- Функціональні вимоги проекту: Які функції має виконувати додаток і які компоненти потрібні для їх реалізації.
- Нелінійність розвитку: Якщо передбачається поступове розширення та додавання нового функціоналу, мікросервісна архітектура може бути більш гнучкою.
- Вимоги до продуктивності: Якщо додаток передбачає велику кількість користувачів або обробку великого обсягу даних, необхідно вибрати архітектуру, яка забезпечує швидку та ефективну роботу системи.
- Вимоги до безпеки: Якщо безпека є критичним аспектом проекту, слід вибрати архітектуру, яка надає високий рівень захисту даних та доступу.

Після аналізу наведених типів архітектур для веб-додатка з просування тематичних аніме товарів, було прийняте рішення про використання мікросервісної архітектури. Цей підхід надає гнучкість, масштабованість та можливість розподіленого розвитку, що відповідає вимогам проекту. Кожен сервіс може виконувати окрему функціональну одиницю, таку як каталог аніме товарів, кошик покупок, оплата тощо. Взаємодія між сервісами здійснюється за допомогою API, що забезпечує гнучкість та розширюваність системи. Обраний підхід до архітектури дозволить розробити веб-додаток, який відповідає потребам користувачів, забезпечує швидку та ефективну роботу, а також зручну підтримку та розвиток.

### **3.2. Функціональні вимоги до веб-додатка з просування аніме товарів**

Функціональні вимоги до веб-додатка для просування тематичних аніме товарів включають детальний опис його функціональних можливостей та

						Аркуш
					ДТЕУ 121 07-11.БР	40
Зм.	Аркуш	№ докум	Підпис	Дата		



основних функціональних модулів, які необхідно реалізувати. Давайте розглянемо ці аспекти детальніше.

Детальний опис функціональних можливостей:

- Реєстрація та авторизація користувачів: Веб-додаток має надавати можливість користувачам створити обліковий запис і авторизуватись для доступу до функцій.
- Пошук та перегляд аніме товарів: Користувачі повинні мати можливість шукати аніме товари за різними критеріями (назва, жанр, рейтинг тощо) і переглядати детальну інформацію про них.
- Додавання аніме товарів до кошика: Користувачі повинні мати можливість додавати бажані аніме товари до кошика покупок.
- Керування кошиком покупок: Користувачі мають змогу переглядати та редагувати вміст свого кошика покупок, додавати або видаляти товари.
- Рейтинг та відгуки: Користувачі можуть залишати рейтинги та відгуки про придбані аніме товари, що допомагає іншим користувачам при виборі.

Основні функціональні модулі:

- Модуль користувача: Включає функції реєстрації, авторизації, керування обліковим записом та профілем користувача.
- Модуль аніме товарів: Включає функції пошуку, перегляду та додавання аніме товарів до кошика.
- Модуль кошика покупок: Забезпечує функції керування вмістом кошика, додавання та видалення товарів.
- Модуль відгуків та рейтингів: Дозволяє користувачам залишати відгуки та оцінки аніме товарів.

Обговорені функціональні вимоги та основні функціональні модулі веб-додатка з просування тематичних аніме товарів визначають його можливості та функціонал. Враховуючи ці вимоги, буде легше побудувати систему, що

						Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	

задовольняє потреби користувачів та пропонує зручний та ефективний інтерфейс для взаємодії.

### 3.3. Архітектура бази даних веб-додатка з просування аніме товарів

Архітектура бази даних веб-додатка для просування тематичних аніме товарів включає визначення сутностей та взаємозв'язків, вибір типу бази даних та обґрунтування вибору, а також опис схеми бази даних та структури таблиць. Розглянемо кожен аспект детальніше:

Визначення сутностей та взаємозв'язків. При проектуванні бази даних необхідно визначити сутності, які будуть представлені в базі даних. У веб-додатку для просування аніме товарів можуть бути такі сутності:

- Користувачі: Зберігає інформацію про користувачів, таку як ідентифікатор, ім'я, електронну пошту, пароль тощо.
- Аніме товари: Містить дані про аніме товари, такі як назва, опис, жанр, рейтинг, ціна тощо.
- Кошик покупок: Записує відношення між користувачем і аніме товарами, які знаходяться в його кошику.
- Рейтинг: містить дані про рейтинг та відгуки

Вибір типу бази даних та обґрунтування вибору: Для розробки веб-додатка можуть бути використані різні типи баз даних, такі як реляційні бази даних наприклад, MySQL, PostgreSQL, NoSQL бази даних наприклад, MongoDB, CouchDB або гібридні рішення. Вибір типу бази даних залежить від ряду факторів, таких як обсяг та складність даних, швидкодія, потреби в масштабованості та зручність роботи з даними. На основі попередніх пунктів було вибрано модель управління реаліційними базами даних PostgreSQL.

						Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	

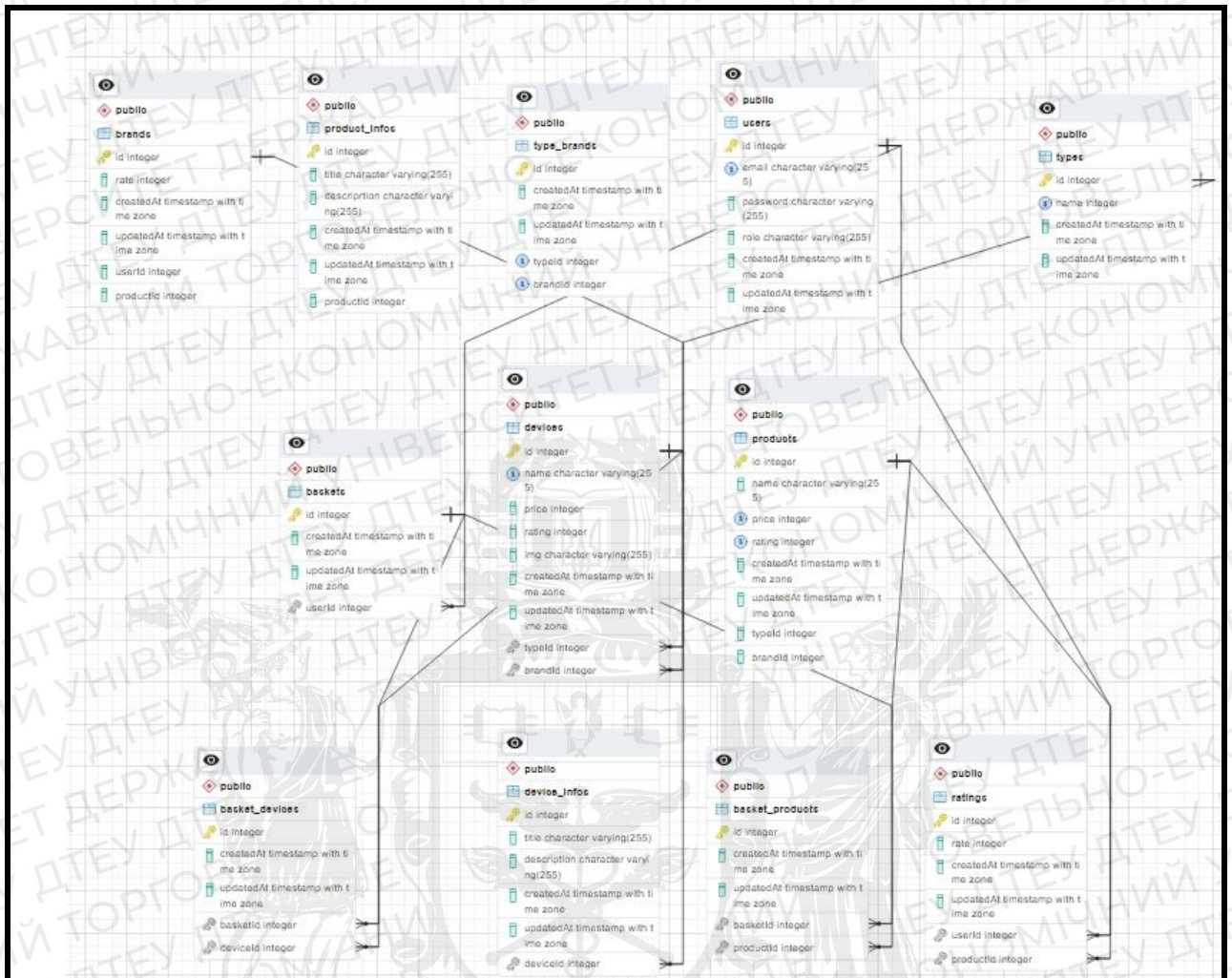


Рис.3.1 Фізична модель бази даних

Джерело: Побудовано автором

Опис схеми бази даних та структури таблиць: Після визначення сутностей та взаємозв'язків необхідно створити схему бази даних, що включає структуру таблиць та їх зв'язки. Наприклад, для сутності "Користувачі" можна створити таблицю "users" з колонками, що відповідають атрибутам користувачів, таким як id, name, email, password тощо. Аналогічно, для інших сутностей створені відповідні таблиці з необхідними колонками та зв'язками.

Опис зв'язків таблиць у базі даних:

- Зв'язок "User.hasOne(Basket); Basket.belongsTo(User);":

Користувач (User) має одну кошик (Basket). Кошик (Basket) належить до одного користувача (User).

- Зв'язок "User.hasMany(Rating); Rating.belongsTo(User);":  
Користувач (User) має багато рейтингів (Rating). Рейтинг (Rating) належить до одного користувача (User).
- Зв'язок "Basket.hasMany(BasketProduct); BasketProduct.belongsTo(Basket);":  
Кошик (Basket) має багато товарів кошика (BasketProduct). Товар кошика (BasketProduct) належить до одного кошика (Basket).
- Зв'язок "Type.hasMany(Product); Product.belongsTo(Type);":  
Тип (Type) має багато продуктів (Product). Продукт (Product) належить до одного типу (Type).
- Зв'язок "Brand.hasMany(Product); Product.belongsTo(Brand);":  
Бренд (Brand) має багато продуктів (Product). Продукт (Product) належить до одного бренду (Brand).
- Зв'язок "Product.hasMany(Rating); Rating.belongsTo(Product);":  
Продукт (Product) має багато рейтингів (Rating). Рейтинг (Rating) належить до одного продукту (Product).
- Зв'язок "Product.hasMany(BasketProduct); BasketProduct.belongsTo(Product);":  
Продукт (Product) має багато товарів кошика (BasketProduct). Товар кошика (BasketProduct) належить до одного продукту (Product).
- Зв'язок "Product.hasMany(ProductInfo, { as: 'info' }); ProductInfo.belongsTo(Product);":  
Продукт (Product) має багато інформаційних даних (ProductInfo) під псевдонімом 'info'. Інформаційні дані (ProductInfo) належать до одного продукту (Product).
- Зв'язок "Type.belongsToMany(Brand, { through: TypeBrand }); Brand.belongsToMany(Type, { through: TypeBrand });":

Тип (Type) належить до багатьох брендів (Brand) через проміжну таблицю TypeBrand. Бренд (Brand) належить до багатьох типів (Type) через проміжну таблицю TypeBrand.

Ці зв'язки визначають залежності між різними сутностями бази даних та дозволяють встановлювати зв'язки між ними для збереження та отримання даних.



```
76 User.hasOne(Basket);
77 Basket.belongsTo(User);
78
79 User.hasMany(Rating);
80 Rating.belongsTo(User);
81
82 Basket.hasMany(BasketProduct);
83 BasketProduct.belongsTo(Basket);
84
85 Type.hasMany(Product);
86 Product.belongsTo(Type);
87
88 Brand.hasMany(Product);
89 Product.belongsTo(Brand);
90
91 Product.hasMany(Rating);
92 Rating.belongsTo(Product);
93
94 Product.hasMany(BasketProduct);
95 BasketProduct.belongsTo(Product);
96
97
98 Product.hasMany(ProductInfo, { as: 'info' });
99 ProductInfo.belongsTo(Product);
100
101
102 Type.belongsToMany(Brand, { through: TypeBrand });
103 Brand.belongsToMany(Type, { through: TypeBrand });
```

Snipped

Рис 3.2 Зв'язки таблиць бази даних у кодї  
Джерело: Розроблено автором

### 3.4. Розробка користувацького інтерфейсу веб-додатку

Розробка користувацького інтерфейсу веб-додатка для просування тематичних аніме товарів включає опис дизайну та структури інтерфейсу, розгляд інтерфейсних елементів та застосування принципів UX-дизайну. Розглянемо кожен аспект детальніше:

Опис дизайну та структури користувацького інтерфейсу: Дизайн інтерфейсу повинен бути привабливим та залучати користувачів. Розробка дизайну включає вибір кольорової палітри, типографіки, компоновання елементів та загальну структуру сторінок. Важливо забезпечити зручну навігацію, зрозумілу ієрархію інформації та привабливе візуальне оформлення.

Розгляд інтерфейсних елементів: Користувацький інтерфейс веб-додатка має включати різноманітні інтерфейсні елементи для зручної взаємодії з користувачами. Наприклад, меню навігації допомагає користувачам переходити між різними розділами додатка. Форми використовуються для введення та обробки даних. Списки товарів демонструють перелік доступних товарів з відповідними деталями. Кнопки використовуються для виконання певних дій, наприклад, додавання товару до кошика або збереження змін.

Застосування принципів UX-дизайну: UX (User Experience) - це взаємодія користувача з додатком та його враження від цієї взаємодії. При розробці користувацького інтерфейсу важливо враховувати принципи UX-дизайну, які сприяють зручності, ефективності та задоволенню користувачів. Наприклад, розміщення інтерфейсних елементів повинно бути логічним та інтуїтивно зрозумілим. Забезпечення швидкості реакції додатка на дії користувача, а також розміщення найважливішої інформації на видному місці допоможуть покращити UX. Враховуючи описані аспекти, розробка користувацького інтерфейсу веб-додатка для просування тематичних аніме товарів забезпечує зручну навігацію, привабливий дизайн та використання принципів UX-

						Аркуш
						46
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	

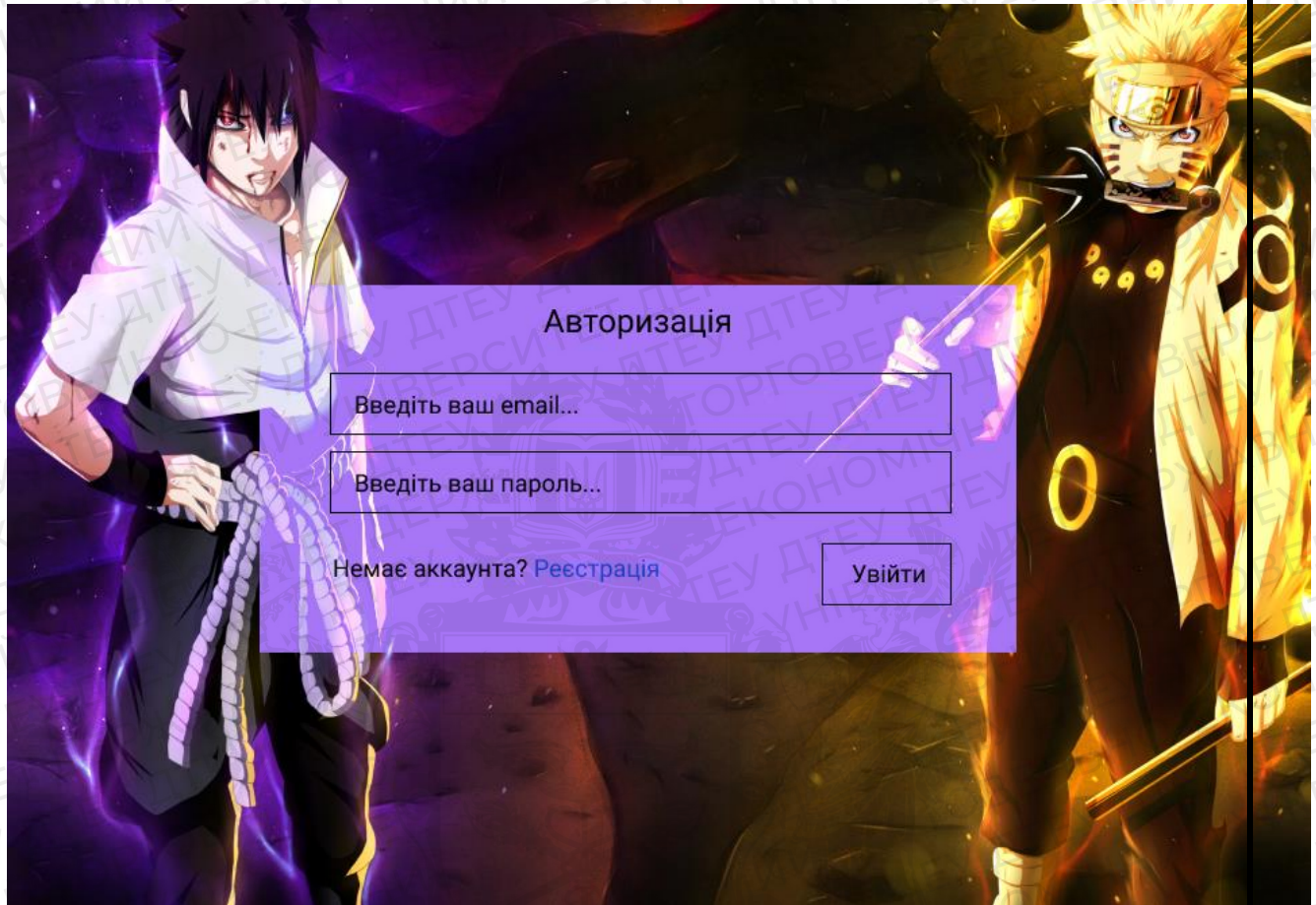


Рис.3.3 Вікно Авторизації веб-додатку

Джерело: Розроблено автором

Сторінка вибору товарів є важливою частиною веб-додатка, яка дозволяє користувачам переглядати доступні товари і здійснювати вибір з різних варіантів основні елементи та функціонал, які присутні на сторінці вибору товарів.

Список товарів: Сторінка містить список товарів, які доступні для вибору. Кожен товар представлений зображенням, назвою, описом та ціною.

Фільтри: Для полегшення вибору товарів надані фільтри, які дозволяють користувачам обмежити результати за певними критеріями. Наприклад, фільтри включують категорію товару, бренд.

Деталі товару: Користувачі можуть отримати детальну інформацію про товар, натиснувши на його зображення або назву. Це включає розширений опис, характеристики, відгуки покупців, фотографії тощо.

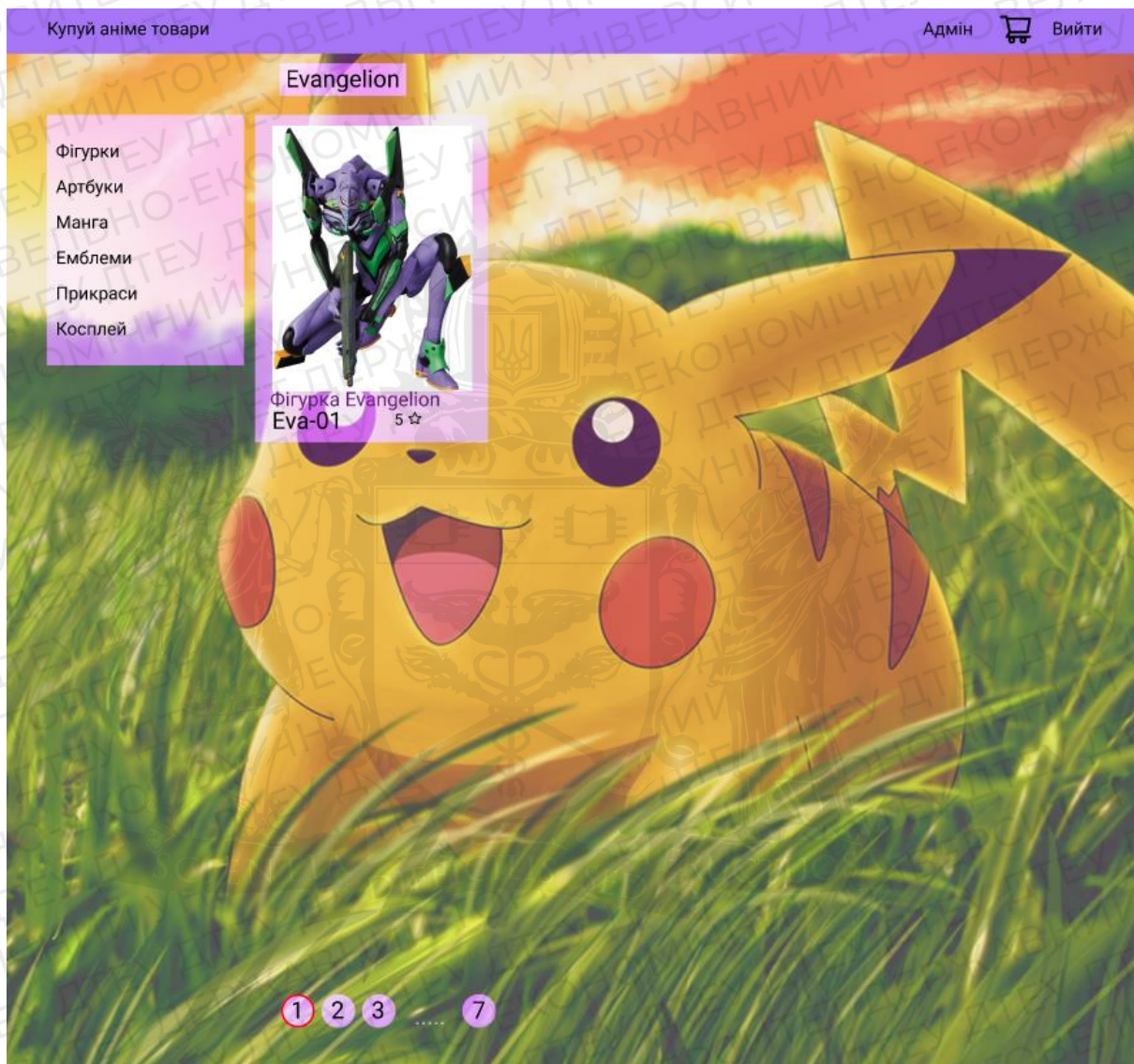


Рис.3.4 Сторінка вибору товарів веб-додатку

Джерело: розроблено автором

### 3.5.Реалізація функціональних модулів веб-додатку

Реалізація функціональних модулів веб-додатка для просування тематичних аніме товарів включає опис процесу реалізації та використання визначених підходів та технологій. Розглянемо кожен аспект детальніше:

						Аркуш
						48
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-11.БР	



Опис процесу реалізації основних функціональних модулів: Реалізація функціональних модулів передбачає розробку програмного коду та інтеграцію з іншими компонентами веб-додатка. Для кожного модуля було визначено основні функціональні вимоги та розробити відповідний програмний код для їх виконання. Процес реалізації включає створення базових класів або об'єктів, розробку функцій, взаємодію з базою даних та інші необхідні операції.

```

16  const Auth = observer(() => {
17    const navigate = useNavigate();
18    const { user } = useContext(Context);
19    let location = useLocation();
20    const isLogin = location.pathname === LOGIN_ROUTE;
21    const [email, setEmail] = useState('');
22    const [password, setPassword] = useState('');
23
24    const click = async () => {
25      try {
26        let userData;
27        if (isLogin) {
28          userData = await login(email, password);
29        } else {
30          userData = await registration(email, password);
31        }
32        user.setUser(userData);
33        user.setIsAuth(true);
34        navigate(SHOP_ROUTE);
35      } catch (error) {
36        alert(error.response.data.message);
37      }
38    };
  
```

Рис.3.5 Приклад реалізації реєстрації у кодї

Використання підходів та технологій: Підходи та технології, що були визначені у попередніх розділах, використовуються під час реалізації функціональних модулів. Наприклад, використання вибраної бібліотеки React js дозволяє швидко розробляти та інтегрувати різні функціональні модулі. Також, використання мов програмування, які були вибрані для серверної та клієнтської сторінок JavaScript, допомагає ефективно реалізувати функціональність додатка. Під час реалізації функціональних модулів веб-додатка для просування тематичних аніме товарів було дотримано вимоги

проекту та використовувати визначені підходи та технології. Це дозволило створити функціональності, які відповідають потребам користувачів та забезпечують ефективну роботу додатка.

### 3.6. Висновки по третьому розділу

Висновки по третьому розділу "Реалізація веб-додатка для просування тематичних аніме товарів" такі: У цьому розділі була розглянута реалізація веб-додатка для просування тематичних аніме товарів згідно з визначеними вимогами та обраною архітектурою. Були використані сучасні технології та інструменти для розробки та реалізації функціональності додатка.

Були реалізовані основні функціональні модулі, такі як реєстрація користувачів, перегляд та пошук товарів, додавання товарів до кошика та оформлення замовлення. Кожен модуль був ретельно протестований для переконання в його правильному функціонуванні та відповідності вимогам. Під час реалізації була забезпечена безпека веб-додатка шляхом застосування відповідних заходів, таких як хешування паролів, перевірка прав доступу та обмеження ризиків вразливостей.

Було проведено тестування додатка з метою перевірки його функціональності, безпеки та продуктивності. Виявлені помилки та недоліки були виправлені, що забезпечило стабільну та надійну роботу веб-додатка.

Загальною висновкою є те, що реалізація та тестування веб-додатка для просування тематичних аніме товарів були успішно виконані. Додаток відповідає вимогам та надає користувачам зручний та безпечний спосіб отримання інформації про товари та оформлення замовлень. Дані результати відкривають можливості для подальшого розвитку та вдосконалення веб-додатка, таких як розширення функціональності, покращення інтерфейсу користувача та вдосконалення системи безпеки. Таким чином, реалізація та тестування веб-додатка для просування тематичних аніме товарів були

					Аркуш
					50
Зм.	Аркуш	№ докум	Підпис	Дата	

успішно завершені, що дозволить користувачам отримувати якісний сервіс та задовольняти свої потреби у придбанні аніме товарів



					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		51

## ВИСНОВКИ І ПРОПОЗИЦІЇ

У ході виконання дипломної роботи було проведено детальне дослідження та розробка веб-орієнтованого додатку з просування тематичних аніме товарів. Застосування веб-технологій у сфері електронної комерції, зокрема в контексті просування та продажу аніме товарів, має великий потенціал і може принести значні переваги.

У першому розділі дипломної роботи була описана проблема, яка полягає в нестачі зручного та ефективного рішення для просування тематичних аніме товарів. Було встановлено актуальність теми та визначено мету та завдання дослідження.

У другому розділі було проведено аналіз предмета дослідження, зокрема технологій розробки веб-додатків. Були розглянуті різні фреймворки, клієнтські та серверні технології, а також архітектура та бази даних для веб-додатків. Цей аналіз дозволив обрати оптимальні технології для розробки веб-орієнтованого додатку з просування аніме товарів.

У третьому розділі було запропоновано технічне рішення для веб-орієнтованого додатку з просування тематичних аніме товарів. Були розроблені основні функціональні та нефункціональні вимоги, структура бази даних та архітектура додатку. У процесі роботи над дипломною роботою було проведено дослідження, розроблено технічне завдання та виконано детальний аналіз предмета дослідження. Було обрано оптимальні технології та розроблено архітектуру веб-додатку. Також було описано підхід до забезпечення безпеки та взаємодії зі сторонніми сервісами. Завершуючи, виконання даної дипломної роботи дало змогу поглибити знання про розробку веб-додатків та їх використання у сфері електронної комерції. Результатом роботи є розроблене технічне завдання та детальний аналіз предмета

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 07-11.БР			
Зав. каф.		Криворучко О.В.		28.04.23	Веб-орієнтований додаток «Anime Gift-merch» з просування тематичних аніме товарів	Стадія	Аркуш	Аркушів
Керівник		Десятко А.М.		28.04.23		ВП	52	54
Гарант		Рзаєва С.Л.		28.04.23		Факультет інформаційних технологій		
Розробив		Дьяконов В.О.		28.04.23		4 курс, 7 група		
						Висновки та пропозиції		

дослідження, що можуть бути використані для подальшої реалізації веб-орієнтованого додатку з просування аніме товарів.



					ДТЕУ 121 07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		53

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ріст популярності аніме на стрімінгових сервісах  
 URL: <https://www.economist.com/business/2021/06/05/streaming-and-covid-19-have-entrenched-animes-global-popularity> (дата звернення 03.04.2023)
2. Інформація про бібліотеку React  
 URL: <https://uk.legacy.reactjs.org/tutorial/tutorial.html#what-is-react> (дата звернення 07.04.2023)
3. Інформація про веб-фреймворк Angular URL: <https://angular.io/docs> (дата звернення 15.04.2023)
4. Інформація про веб фреймвор Express URL: <https://expressjs.com/> (дата звернення 20.04.2023)
5. Інформація про монолітну архітектуру веб-додатків  
 URL: <https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith> (дата звернення 25.04.2023)
6. Інформація про клієнт-серверну архітектуру веб-додатків  
 URL: <https://www.britannica.com/technology/client-server-architecture> (дата звернення 02.05.2023)
7. Інформація про мікросерверну архітектуру веб-додатків  
 URL: <https://microservices.io/> (дата звернення 04.05.2023)
8. Книга "JavaScript: The Definitive Guide" авторства David Flanagan
9. Книга "RESTful Web Services" авторства Leonard Richardson та Sam Ruby

<i>ДТЕУ 121 07-11.БР</i>																	
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>													
Зав. каф.		Криворучко О.В.		23.12.22	Веб-орієнтований додаток «Аніме Gift-merch» з просування тематичних аніме товарів  <i>Список використаних джерел</i>												
Керівник		Десятко А.М.		23.12.22													
Гарант		Рзаєва С.Л.		23.12.22													
Розробив		Дьяконов В.О.		23.12.22													
					<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: center;"><i>Стадія</i></td> <td style="width: 20%; text-align: center;"><i>Аркуш</i></td> <td style="width: 50%; text-align: center;"><i>Аркушів</i></td> </tr> <tr> <td style="text-align: center;"><i>СВД</i></td> <td style="text-align: center;">54</td> <td style="text-align: center;">54</td> </tr> <tr> <td colspan="3" style="text-align: center;">Факультет інформаційних технологій</td> </tr> <tr> <td colspan="3" style="text-align: center;">4 курс, 7 група</td> </tr> </table>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	<i>СВД</i>	54	54	Факультет інформаційних технологій			4 курс, 7 група		
<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>															
<i>СВД</i>	54	54															
Факультет інформаційних технологій																	
4 курс, 7 група																	

## ДОДАТКИ

### Додаток А

```
import sequelize from '../db.js';
import { DataTypes } from 'sequelize';

const User = sequelize.define('user', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
  email: { type: DataTypes.STRING, unique: true },
  password: { type: DataTypes.STRING },
  role: { type: DataTypes.STRING, defaultValue: 'USER' },
});

const Basket = sequelize.define('basket', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
});

const BasketProduct = sequelize.define('basket_product', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
});

const Product = sequelize.define('product', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
  // allowNull false - не может быть равно нулю
  name: { type: DataTypes.STRING, unique: true, allowNull: false },
  price: { type: DataTypes.INTEGER, allowNull: false },
  rating: { type: DataTypes.INTEGER, defaultValue: 0 },
  img: { type: DataTypes.STRING, allowNull: false },
});

const Type = sequelize.define('type', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
  name: { type: DataTypes.STRING, unique: true, allowNull: false },
});
```

```
const Brand = sequelize.define('brand', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
  name: { type: DataTypes.STRING, unique: true, allowNull: false },
});

const Rating = sequelize.define('rating', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
  rate: { type: DataTypes.INTEGER, allowNull: false },
});

const ProductInfo = sequelize.define('product_info', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
  title: { type: DataTypes.STRING, allowNull: false },
  description: { type: DataTypes.STRING, allowNull: false },
});

// сполучна таблиця між типом та брендом
const TypeBrand = sequelize.define('type_brand', {
  id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
});

// Опис зв'язків моделей
// Всі зовнішні ключі sequelize додасть сам при зв'язуванні
// // один запис девайса в бд має багато записів з характеристиками
// Product.hasMany(ProductInfo)

// // для ProductInfo вказуємо, що ця сутність належить Product
// ProductInfo.belongsTo(Product);

// зв'язок один до одного
User.hasOne(Basket);
Basket.belongsTo(User);

User.hasMany(Rating);
Rating.belongsTo(User);
```



```

Basket.hasMany(BasketProduct);
BasketProduct.belongsTo(Basket);

Type.hasMany(Product);
Product.belongsTo(Type);

Brand.hasMany(Product);
Product.belongsTo(Brand);

Product.hasMany(Rating);
Rating.belongsTo(Product);

Product.hasMany(BasketProduct);
BasketProduct.belongsTo(Product);

Product.hasMany(ProductInfo, { as: 'info' });
ProductInfo.belongsTo(Product);

Type.belongsToMany(Brand, { through: TypeBrand });
Brand.belongsToMany(Type, { through: TypeBrand });

export { User, Basket, BasketProduct, Brand, Product, ProductInfo, Rating, Type,
TypeBrand };

export default User;
// зв'язок багато хто до багатьох
// необхідно вказати сполучну таблицю - сполучну модель
// info - назва поля, яка буде у масиву характеристик

```

Додаток Б

```

import Modal from 'react-bootstrap/Modal';
import Button from 'react-bootstrap/Button';
import Form from 'react-bootstrap/Form';
import { useState } from 'react';
import { createBrand } from '../http/productAPI';

const CreateBrand = ({ show, onHide }) => {
  const [value, setValue] = useState("");
  const addBrand = () => {
    createBrand({ name: value })
      .then((data) => {
        setValue("");

```

```

    onHide();
  })
  .catch((error) => alert(error.response.data.message));
});
return (
  <Modal
    size="lg"
    aria-labelledby="contained-modal-title-vcenter"
    centered
    show={ show }
    onHide={ onHide }
  >
    <Modal.Header closeButton>
      <Modal.Title id="contained-modal-title-vcenter">Додати новий
бренд</Modal.Title>
    </Modal.Header>
    <Modal.Body>
      <Form>
        <Form.Control
          value={ value }
          onChange={(e) => {
            setValue(e.target.value);
          }}
          placeholder="Введіть назву бренда"
        >>/Form.Control</Form>
      </Modal.Body>
      <Modal.Footer>
        <Button
          variant={'outline-danger'}
          onClick={ onHide }
        >
          Закрити
        </Button>
        <Button
          variant={'outline-success'}
          onClick={ addBrand }
        >
          Додати
        </Button>
      </Modal.Footer>

```

```

</Modal>
);
};

export default CreateBrand;

import React, { useContext } from 'react';
import { Context } from './Providers';
import Nav from 'react-bootstrap/Nav';
import Navbar from 'react-bootstrap/Navbar';
import Container from 'react-bootstrap/Container';
import { Link } from 'react-router-dom';
import { ADMIN_ROUTE, BASKET_ROUTE, LOGIN_ROUTE, SHOP_ROUTE
} from './utils/consts';
import Button from 'react-bootstrap/Button';
import { observer } from 'mobx-react-lite';
import { useNavigate } from 'react-router-dom';

//observer - щоб mobx відстежував зміни

const Header = observer(() => {
  const navigate = useNavigate();
  const { user } = useContext(Context);
  const { basket } = useContext(Context);

  const logOut = () => {
    user.setUser({});
    user.setIsAuth(false);
    basket.clearProducts();
    localStorage.removeItem('token');
    navigate(LOGIN_ROUTE);
  };
  return (
    <Navbar
      bg="dark"
      variant="dark"
    >
      <Container>
        <Link
          style={{ color: 'white' }}
          to={SHOP_ROUTE}

```

Додаток В

```
>
  Назва магазину
</Link>
  {user.isAuth ? (
    <Nav
      style={{ color: 'white' }}
      className="ml-auto"
    >
      <Button
        variant={'outline-light'}
        onClick={() => navigate(ADMIN_ROUTE)}
      >
        Адмін панель
      </Button>

      <Button
        variant={'outline-light'}
        onClick={() => navigate(BASKET_ROUTE + '/' + user.user.id)}
      >
        Корзина
      </Button>
      <div className="ms-4">{user.user.email}</div>
      <Button
        variant={'outline-light'}
        onClick={() => logOut()}
      >
        Вийти
      </Button>
    </Nav>
  ) : (
    <Nav
      style={{ color: 'white' }}
      className="ml-auto"
    >
      <Button
        variant={'outline-light'}
        onClick={() => navigate(LOGIN_ROUTE)}
      >
        Авторизація
      </Button>
    </Nav>
  )
}
```

```

    )}
  </Container>
</Navbar>
);
});

export default Header;

```

Додаток Г

```

import React from 'react';
import { Outlet } from 'react-router-dom';
import Header from '../components/Header';

const MainLayout = () => {
  return (
    <>
      <Header />
      <main>
        <Outlet />
      </main>
    </>
  );
};

export default MainLayout;

```

Додаток Г

```

import Container from 'react-bootstrap/Container';
import Card from 'react-bootstrap/Card';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import Row from 'react-bootstrap/Row';
import Col from 'react-bootstrap/Col';
import { Link } from 'react-router-dom';
import { useLocation } from 'react-router-dom';
import { LOGIN_ROUTE, REGISTRATION_ROUTE, SHOP_ROUTE } from
'../utils/consts';
import { login, registration } from '../http/userAPI';
import { useContext, useState } from 'react';
import { observer } from 'mobx-react-lite';
import { Context } from '../components/Providers';
import { useNavigate } from 'react-router-dom';

```

```

const Auth = observer(() => {
  const navigate = useNavigate();
  const { user } = useContext(Context);
  let location = useLocation();
  const isLogin = location.pathname === LOGIN_ROUTE;
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const click = async () => {
    try {
      let userData;
      if (isLogin) {
        userData = await login(email, password);
      } else {
        userData = await registration(email, password);
      }
      user.setUser(userData);
      user.setIsAuth(true);
      navigate(SHOP_ROUTE);
    } catch (error) {
      alert(error.response.data.message);
    }
  };

  return (
    <Container
      className="d-flex justify-content-center align-items-center"
      style={{ height: window.innerHeight - 54 }}
    >
      <Card
        style={{ width: 600 }}
        className="p-5"
      >
        <h2 className="m-auto">{isLogin ? 'Авторизація' : 'Реєстрація'}</h2>
        <Form className="d-flex flex-column">
          <Form.Control
            className="mt-3"
            placeholder="Введіть email"
            value={email}
            onChange={(e) => setEmail(e.target.value)}
          />
        </Form>
      </Card>
    </Container>
  );
});

```

```

/>
<Form.Control
  className="mt-3"
  placeholder="Введіть пароль"
  value={password}
  onChange={(e) => setPassword(e.target.value)}
  type="password"
/>
<Row>
  <Col>
    {isLoggedIn ? (
      <div>
        Немає аккаунта? <Link
to={REGISTRATION_ROUTE}>Зареєструйтесь</Link>
      </div>
    ) : (
      <div>
        Є аккаунт? <Link to={LOGIN_ROUTE}>Авторизуйтесь</Link>
      </div>
    )}
  </Col>
  <Col className="col-12 col-sm-6">
    <Button
      variant={'outline-success'}
      className="align-self-end"
      onClick={click}
    >
      {isLoggedIn ? 'Увійти' : 'Зареєструватись'}
    </Button>
  </Col>
</Row>
</Form>
</Card>
</Container>
);
});
export default Auth;

```

```
import Container from 'react-bootstrap/esm/Container';
```

Додаток Д

```
import Button from 'react-bootstrap/Button';
import CreateProduct from '../components/modals/CreateProduct';
import CreateType from '../components/modals/CreateType';
import CreateBrand from '../components/modals/CreateBrand';
import { useState } from 'react';
```

```
const Admin = () => {
  const [brandVisible, setBrandVisible] = useState(false);
  const [typeVisible, setTypeVisible] = useState(false);
  const [productVisible, setProductVisible] = useState(false);
  return (
```

```
    <Container className="d-flex flex-column">
```

```
      <Button
```

```
        variant={'outline-dark'}
```

```
        className="mt-4 p-3"
```

```
        onClick={() => setTypeVisible(true)}
      >
```

```
        Додати тип
```

```
    </Button>
```

```
    <Button
```

```
      variant={'outline-dark'}
```

```
      className="mt-4 p-3"
```

```
      onClick={() => setBrandVisible(true)}
    >
```

```
      Додати бренд
```

```
    </Button>
```

```
    <Button
```

```
      variant={'outline-dark'}
```

```
      className="mt-4 p-3"
```

```
      onClick={() => setProductVisible(true)}
    >
```

```
      Додати пристрій
```

```
    </Button>
```

```
    <CreateBrand
```

```
      show={brandVisible}
```

```
      onHide={() => setBrandVisible(false)}
    />
```

```
    <CreateProduct
```

```
      show={productVisible}
```

```
      onHide={() => setProductVisible(false)}
    />
```



```
<CreateType
  show={typeVisible}
  onHide={() => setTypeVisible(false)}
/>
</Container>
);
};
export default Admin;
```

