

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ	Економіки та бізнес-освіти
Кафедра	Економіки та цифрового бізнесу
Спеціальність	Комп'ютерні науки
Форма навчання	Денна

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Зацеркляної Тетяни Олегівни

(прізвище, ім'я, по батькові здобувача)

на тему Розробка інтерактивного книжкового маркетплейса
з обмінним фондом

(повна назва теми)

за матеріалами

(повна назва бази дослідження)

науковий керівник

(наук. ступінь, вчене звання)

(підпис)

Шокотько Л.М.

(прізвище, ініціали)

Робота допущена до захисту в ЕК

Протокол засідання кафедри

від 09.06.2025р. № 12

Завідувач кафедри

(підпис)

к.е.н., доцент

наук. ступень, вчене звання

Радько В.М.

прізвище, ініціали

ЗАТВЕРДЖЕНО

Наказ Міністерства освіти і науки, молоді та спорту України
29 березня 2012 року № 384

Форма № Н-9.01

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ ТА БІЗНЕС-ОСВІТИ
(повне найменування вищого навчального закладу)

Кафедра економіки та цифрового бізнесуОсвітній ступінь бакалаврСпеціальність комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ **В.М. Радько**

“07” квітня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА ЗДОБУВАЧУ
Зацеркляній Тетяні Олегівні

1. Тема роботи «Розробка інтерактивного книжкового маркетплейса з обмінним фондом»

науковий керівник роботи Шокотько Людмила Миколаївна

затверджені наказом вищого навчального закладу від «04» квітня 2025 р. № 224-ст

2. Строк подання здобувачем роботи 31.05.2025р.

3. Зміст кваліфікаційної роботи бакалавра, об'єкт, предмет та мета дослідження:

Розділ 1 Теоретичний аналіз предметної області та особливостей веб-розробки.

Розділ 2 Проектування онлайн-платформ для продажу та обміну книг.

Розділ 3 Програмна реалізація проекту.

Об'єкт дослідження – процес проектування та розробки онлайн-платформ для продажу та обміну книг.

Предмет дослідження – методи та засоби розробки інтерактивного веб-додатку з функцією взаємодії між користувачами, використанням сучасних технологій (React, Node.js, PostgreSQL, Amazon S3)

Мета кваліфікаційної роботи бакалавра – розробка інтерактивного веб-додатку – книжкового маркетплейса з можливістю обміну книгами, що забезпечує користувацьку взаємодію, управління контентом, авторизацію, модерацію та історію дій.

4. Дата видачі завдання 04.04.2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	до 28.04.2025р.	25.04.2025
2	Підготовка розділу 2	до 16.05.2025р.	15.05.2025
3	Підготовка розділу 3	до 30.05.2025р.	29.05.2025
4	Реєстрація завершеної дипломної роботи	до 31.05.2025р.	30.05.2025
5	Отримання відгуку від наукового керівника	03-04.06.2025р.	04.06.2025
6	Отримання зовнішньої рецензії	05-06.06.2025р.	06.06.2025
7	Перевірка кваліфікаційної роботи на плагіат	02-09.06.2025р.	04.06.2025
8	Попередній захист кваліфікаційної роботи на кафедрі	03.06.2025р.	03.06.2025
9	Допуск кафедрою кваліфікаційної роботи до захисту	09.06.2025р.	09.06.2025
10	Підготовка студента до захисту в ЕК	до 17.06.2025р.	17.06.2025

Завдання підготував науковий керівник

_____ Шокотько Л.М.

Завдання одержав здобувач

_____ Зацеркляна Т.О.

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 76 сторінок, 58 малюнків, 3 таблиці, 27 використаних джерел, 4 додатки.

Об'єкт дослідження: процеси проектування та розробки онлайн-платформ для продажу та обміну книг.

Мета: розробка інтерактивного веб-додатку –маркетплейсу з можливістю обміну книгами, що забезпечує користувацьку взаємодію, управління контентом, авторизацію, модерацію та історію дій.

У результаті дослідження було проаналізовано сучасні веб-технології, функціональні вимоги до маркетплейса, особливості побудови веб-архітектури та UX-інтерфейсів. Розроблено структуру бази даних, реалізовано серверну логіку, створено систему авторизації та захищеного доступу, інтегровано хмарне зберігання файлів. Також був розроблений механізм обміну книгами, модерації контенту та реалізації сповіщень між користувачами.

Результати роботи можуть бути використані для створення онлайн-маркетплейсів, платформ обміну цифровими або фізичними товарами, систем з керуванням контентом та взаємодією між користувачами.

Ключові слова: авторизація, база даних, веб-додаток, електронна комерція, купівля, користувацький інтерфейс, маркетплейс, модерація, обмін книгами, продаж книг, сповіщення, хмарне зберігання, BACKEND, FRONTEND, JWT, NODE, REACT, REST API

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	5
ВСТУП	6
РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОСОБЛИВОСТЕЙ ВЕБ-РОЗРОБКИ	8
1.1. Теоретичні аспекти веб-розробки. Огляд сучасних технологій ...	8
1.2. Аналіз особливостей розробки систем електронної комерції	15
1.3. Аналіз механізмів продажу та обміну книг в онлайн-сервісах ...	18
1.4. Вимоги до функціоналу маркетплейса	22
Висновки до розділу 1	23
РОЗДІЛ 2. ПРОЕКТУВАННЯ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ТА ОБМІНУ КНИГ	25
2.1. Проектування архітектури веб-сайту	25
2.2. Проектування інтерфейсу користувача	32
2.3. Проектування інтерфейсу обмінного фонду	39
2.4. Проектування механізму запитів на обмін	44
2.5. Проектування бази даних маркетплейса	45
Висновки до розділу 2	52
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЄКТУ	54
3.1. Реалізація серверної частини та налаштування бази даних	54
3.2. Реалізація функціональності маркетплейса	58
3.3. Програмна реалізація інтерфейсу користувача	62
3.4. Тестування веб-сайту	68
Висновки до розділу 3	74
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	78
ДОДАТКИ	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

TCP – Transmission Control Protocol

IP – Internet Protocol

WWW – WorldWide Web

HTTP – HyperText Transfer Protocol

HTML – HyperText Markup Language

URL – Uniform Resource Locator

CSS – Cascading Style Sheets

SPA – Single Page Application

JWT – JSON Web Tokens

SSL – Secure Sockets Layer

ВСТУП

Книжкова індустрія є важливою частиною культурного та економічного життя суспільства. В умовах розвитку цифрових технологій і глобалізації вона зазнала значних змін. Сучасний читач, маючи доступ до різноманітних форматів книг, таких як паперові, електронні та аудіокниги, обирає зручність і швидкість. Це спричинило появу нових тенденцій у книжковій галузі, зокрема зростання популярності електронних платформ для читання та інтеграцію новітніх рішень електронної комерції.

Попри виклики, пов'язані з воєнним станом, книжковий ринок України демонструє зростання. Станом на лютий 2024 року в країні працює близько 461 книгарні, хоча ще у вересні 2023 року їх було лише 422. У 2023 році випуск книжкової продукції зріс на 73% за кількістю видань і на 203% за тиражем порівняно з попереднім роком. Такий приріст свідчить не лише про адаптацію ринку до нових умов, але й про зміну споживчих уподобань – попит на аудіо- та електронні книги постійно зростає, що відображає глобальні тренди.

Електронна комерція відіграє ключову роль у трансформації книжкового бізнесу. Використання штучного інтелекту для створення персоналізованих рекомендацій та впровадження розширеної реальності дозволяють значно покращити клієнтський досвід. У книжковій індустрії це проявляється в таких інноваціях, як персоналізований підбір літератури, підписки на необмежений доступ до каталогів, інтеграція цифрових платіжних систем і широке використання аналітики великих даних. Ці рішення сприяють популяризації електронних форматів і зміцнюють позиції книжкової індустрії в умовах цифровізації.

У сучасних умовах цифровізації торгівлі та поширення онлайн-сервісів індустрія електронної комерції стрімко розвивається, змінюючи способи взаємодії між продавцями та покупцями. Одну з багатьох ніш в цій сфері займають книжкові онлайн-платформи. Поряд із традиційною моделлю купівлі-продажу все більшого поширення набуває концепція обміну книгами, що

дозволяє користувачам купувати і обмінюватися літературою безпосередньо між собою.

Amazon, Google, Rakuten та інші, активно розробляють нові функції для електронної комерції, включаючи персоналізовані рекомендації, адаптивні інтерфейси та інтеграцію з хмарними сервісами.

Актуальність даної роботи зумовлена потребою у створенні зручного веб-застосунку, який би поєднував у собі можливості електронної комерції та обміну книгами. В умовах воєнного стану та економічної нестабільності в Україні такі сервіси дозволяють оптимізувати витрати користувачів, сприятимуть сталому використанню ресурсів та популяризації читання.

Метою роботи є проектування та створення інтерактивного веб-додатку книжкового маркетплейса з функцією обміну книгами між користувачами. Основні завдання дослідження включають:

- аналіз предметної області та сучасних підходів до побудови маркетплейсів;
- розробку архітектури та інтерфейсів користувача;
- реалізацію серверної частини з підтримкою авторизації, обміну, замовлень та системи сповіщень;
- впровадження механізмів модерації, фільтрації книг та зберігання зображень у хмарному середовищі AWS;
- тестування функціоналу та перевірка його відповідності вимогам.

Об'єктом дослідження є процеси проектування та розробки онлайн-платформ для продажу та обміну книг.

Предметом дослідження виступають методи та засоби розробки інтерактивного веб-додатку з функцією взаємодії між користувачами, використанням сучасних технологій (React, Node.js, PostgreSQL, Amazon S3).

Інформаційну базу дослідження склали практичні матеріали, отримані під час розробки власного веб-застосунку, а також приклади архітектур маркетплейсів, що активно використовуються у сучасній веб-розробці. Додатково враховано приклади реалізації авторизації (JWT), організації REST

API, обробки користувацького контенту та інтеграції з хмарними сервісами на зразок Amazon S3.

Практичне значення роботи полягає у створенні прототипу платформи, яка може бути використана як основа для подальшого впровадження подібних сервісів у сфері книгорозповсюдження, буккросингу та цифрової бібліотечної ініціативи.

РОЗДІЛ 1

ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОСОБЛИВОСТЕЙ ВЕБ-РОЗРОБКИ

1.1 Теоретичні аспекти веб-розробки. Огляд сучасних технологій

У сучасному світі інтернет став невід'ємною частиною повсякденного життя, забезпечуючи доступ до інформації, комунікацію та взаємодію з різноманітними сервісами. Базові принципи роботи інтернету та веб-технологій формувалися ще з кінця 1960-х років, коли Агентство перспективних дослідницьких проєктів (DARPA) створило першу експериментальну мережу для передачі даних. Удосконалення протоколів TCP/IP стало фундаментом для подальшого розвитку інтернету, а в 1983 році поняття "Інтернет" офіційно увійшло в обіг.

Розвиток інтернету привів до створення Всесвітньої павутини (World Wide Web), яка базується на протоколі HTTP та мові розмітки HTML. Завдяки цим технологіям виникла можливість створювати веб-сторінки – одиниці контенту, які користувачі можуть переглядати через веб-браузери. Сукупність веб-сторінок, що мають спільне доменне ім'я, формує веб-сайти. Веб-сайти забезпечують зручний спосіб організації інформації, пропонуючи користувачам доступ до контенту у зрозумілому форматі.

Проте розвиток технологій і зростання потреб користувачів висунули нові вимоги до веб-ресурсів. З'явилася необхідність створення платформ, які дозволяють не лише переглядати інформацію, а й активно взаємодіяти з нею. Такі платформи отримали назву веб-додатків.

Робота веб-додатків заснована на взаємодії клієнтської частини (веб-браузера) із серверною частиною, забезпечуючи доступ до динамічного контенту. Важливим є те, що взаємодія клієнта та сервера базується на використанні протоколів HTTP/HTTPS. На клієнтській стороні веб-додатку використовуються технології HTML, CSS і JavaScript для створення, стилізації

та забезпечення інтерактивності контенту. Серверна частина зазвичай розробляється за допомогою мов програмування, таких як Node.js, PHP, Python або Java, які забезпечують обробку запитів, збереження даних та логіку додатка. Динамічність та інтерактивність веб-додатків забезпечується за рахунок використання асинхронних запитів (AJAX), що дозволяє оновлювати лише частину сторінки без повного перезавантаження. Схема, представлена на рисунку 1.1, ілюструє етапи взаємодії користувача з веб-додатком:

- користувач вводить запит у веб-браузері;
- браузер відправляє HTTP-запит до веб-сервера;
- веб-сервер обробляє запит та, за необхідності, звертається до бази даних або сторонніх API;
- згенерована відповідь повертається клієнту, де вона обробляється браузером і відображається користувачеві.

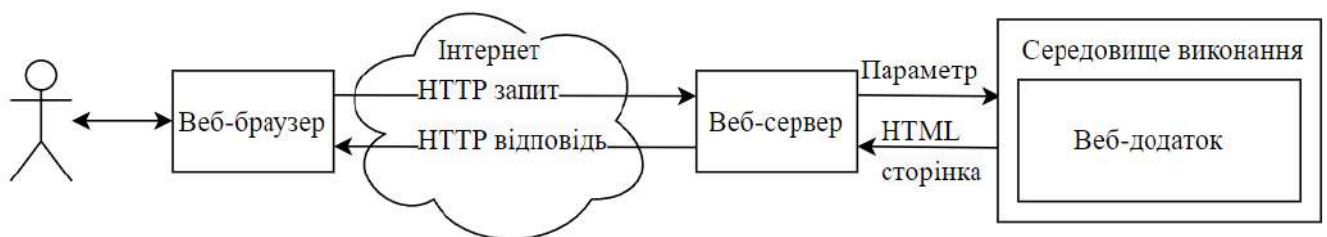


Рис. 1.1. Загальна схема взаємодії користувача з веб-додатком

Джерело: Розроблено із використанням [4]

На основі цього процесу веб-додатки відрізняються від традиційних веб-сайтів. Веб-сайти, як правило, мають статичний характер, пропонуючи інформацію без можливості взаємодії з нею. У свою чергу, веб-додатки дозволяють користувачам виконувати динамічні дії в реальному часі. Наприклад, веб-додаток може дозволити створення акаунтів, заповнення форм, обробку транзакцій або отримання персоналізованих даних. Ці особливості роблять їх ідеальними для застосування у складних системах, таких як електронна комерція, онлайн-банкінг чи соціальні мережі.

Таблиця 1.2 надає порівняння між веб-сайтами та веб-додатками, виокремлюючи їхні ключові характеристики. Основні відмінності стосуються функціональності, взаємодії з користувачами та способу обробки даних. Наприклад, для веб-додатків характерна інтерактивність, обробка даних у реальному часі та авторизація користувачів, тоді як веб-сайти здебільшого зосереджені на наданні інформації у статичній формі.

Таблиця 1.1

Відмінності між веб-додатком та веб-сайтом

Параметр	Веб-додаток	Веб-сайт
Призначення	Веб-додаток створено для інтерактивної взаємодії з кінцевим користувачем.	Веб – сайт переважно складається зі статичного контенту і доступний для всіх відвідувачів
Взаємодія з користувачем	Користувач не тільки переглядає інформацію, але й може взаємодіяти з даними на веб-додатку.	Веб-сайт забезпечує користувачам доступ до візуального та текстового контенту для перегляду.
Автентифікація	Для веб-додатків потрібна автентифікація, оскільки вони надають розширені можливості.	автентифікація не є обов'язковою для доступу до загальнодоступного контенту на сайті.
Завдання та складність	Веб-додатки часто мають більш складні функції та взаємодії, порівняно зі звичайними веб-сайтами.	Веб-сайти призначені для відображення інформації та даних на певних сторінках.
Тип ПЗ	Веб-додаток є частиною веб-сайту, але функціонує як окремий програмний компонент.	Веб-сайт є самостійним продуктом, доступним через веб-браузер.
Компіляція	Веб-додаток необхідно компілювати перед розгортанням на сервері.	Веб-сайт не вимагає попередньої компіляції перед публікацією.
Розгортання	Всі зміни у веб-додатку вимагають перекомпіляції та повторного розгортання на сервері.	Зміни на веб-сайті можна вносити без повного перекомпілювання; достатньо оновити HTML – код.

(розроблено автором)

Завдяки прогресу веб-технологій веб-додатки також отримали можливість інтеграції з іншими сервісами через API, наприклад, з платіжними системами чи геолокаційними платформами. Інший важливий аспект – безпека, яка включає використання SSL/TLS для захисту передачі даних, а також впровадження сучасних методів автентифікації, таких як OAuth чи JWT, які забезпечують

високий рівень захисту даних користувачів. Це дозволяє веб-додаткам підтримувати сучасні стандарти безпеки та забезпечувати захищений досвід для користувачів.

Важливим досягненням у розвитку веб-додатків стало використання прогресивних технологій, таких як PWA (прогресивні веб-додатки), що поєднують функціональність звичайних веб-додатків із зручністю мобільних додатків. Веб-додатки з PWA можуть працювати офлайн, забезпечують швидке завантаження і мають адаптивний дизайн для різних пристроїв.

Таким чином, сучасні веб-додатки стали інструментом, який поєднує функціональність програмного забезпечення з доступністю та зручністю використання через браузер. Їхня поширеність у різних галузях – від електронної комерції до соціальних мереж зумовлює необхідність глибшого розуміння функціонування і технологій, що лежать в основі їх розробки.

У цьому контексті важливим є аналіз сучасних інструментів та методів веб-розробки, які дозволяють реалізовувати такі рішення. Розробка веб-додатків вимагає підходу, який охоплює визначення мети проєкту, вибір архітектури, побудову інтерфейсу користувача та організацію взаємодії між клієнтом і сервером. Усе це неможливо без використання відповідних мов програмування, бібліотек, фреймворків та платформ, які забезпечують гнучкість, масштабованість і безпеку веб-продукту.

Процес створення веб-продукту починається з чіткого розуміння його мети. Це може бути платформа для соціальної взаємодії, онлайн-магазин, інформаційний портал, стрімінгова платформа або інструмент для онлайн-навчання. Кожен тип додатку вимагає різного підходу до проектування і вибору технологій, від яких залежить, наскільки зручним буде продукт для кінцевого користувача. Зазвичай, веб-розробка поділяється на дві основні частини: фронтенд та бекенд. Кожна з цих частин має свої інструменти і фреймворки, які допомагають розробникам створювати ефективні, функціональні та привабливі веб-продукти.

Фронтенд – це те, що безпосередньо бачить користувач у браузері. Фронтенд включає у себе дизайн інтерфейсу, розташування елементів, кольорову гаму, а також реакцію інтерфейсу на дії користувача. Основні мови для розробки фронтенду – це HTML, CSS та JavaScript. HTML забезпечує структуру сторінки, CSS – її стилізацію, а JavaScript додає інтерактивність.

HTML – мова гіпертекстової розмітки та технологія створення веб-додатків, яка використовується для створення та структурування вмісту сторінки. В основі лежить концепція розмітки, яка дає змогу вказати браузеру, як вміст буде відображатися на екрані. HTML визначає основну структуру сторінки: заголовки, абзаци, списки, посилання, зображення та інші елементи. Також забезпечується можливість вбудовувати інші технології на кшталт CSS і JavaScript для додаткової стилізації, надання інтерактивності.

Структура HTML – документа схожа на "каркас" сторінки, що допомагає браузеру правильно відобразити і організувати інформацію. HTML складається з різних елементів, кожен з яких має своє призначення. Основні компоненти HTML-документа включають:

- `<html>` – кореневий елемент, що обгортає всю розмітку сторінки;
- `<html>` – розділ, де розміщуються метадані документа, наприклад, заголовок, підключення CSS стилів, мета-теги, тощо;
- `<body>` – головний розділ сторінки, що містить усі елементи видимі користувачу: заголовки, текстові блоки, зображення, відео, посилання, тощо.

У середині `<body>` документ будується за допомогою вкладених елементів (або тегів), які формують ієрархічну структуру сторінки, відому як DOM (Document Object Model). Кожен елемент має свої атрибути та стилі, а також містити інші елементи, що дозволяє створювати складні, багаторівневі інтерфейси.

Після того, як структура сторінки створена, на черзі йде застосування CSS та JS. CSS – це мова стилів, яка визначає відображення HTML-документів. Наприклад, CSS працює зі шрифтами, кольором, полями, рядками, висотою,

шириною, фоновими зображеннями, позиціонуванням елементів і багатьма іншими речами. Використання CSS економить багато роботи. При цьому описи стилю зазвичай зберігаються у зовнішніх файлах з розширенням. CSS. За допомогою такого зовнішнього файлу таблиці стилів можна змінити зовнішній вигляд всього сайту, змінивши лише один файл CSS.

JS – це мова програмування, за допомогою якою можна додавати та створювати інтерактивність. Використовуючи JS можна обробляти події, здійснювати валідацію форм, динамічно змінювати контент сторінки, виконувати запити до сервера без перевантаження сторінки. Діапазон можливостей залежить від того, хто користується цим.

Для складних та багатофункціональних веб-додатків часто використовують фреймворки та бібліотеки, які допомагають ефективно керувати компонентами та взаємодіяти з DOM. Три популярніших вибору серед розробників це: React, Angular, Vue.

React – це бібліотека для побудови інтерфейсів користувача, яка дозволяє створювати компоненти – невеликі, незалежні частини інтерфейсу, через це їх можна використовувати один раз і потім в різних місцях. Цей підхід дає змогу працювати інтерфейс в рази швидше. Angular – це фреймворк для розробки додатків і платформа для створення складних односторінкових додатків, він дозволяє синхронізувати дані і відображення на екрані, має готові інструменти для обробки форм, налаштування маршрутів між сторінками, тощо. Vue – фреймворк, який працює на JS, створений для розробки інтерфейсів та є свого роду сумішшю React та Angular. Спочатку можна помітити, що це бібліотека для роботи з DOM, а потім, що це фреймворк, комфортний для великий проєктів. Нижче, представлено порівняння (табл. 1.3) між React, Angular, Vue.

Зі зростанням популярності веб-додатків виникла потреба у зберіганні великої кількості мультимедійного контенту: зображень, аудіо, електронних документів тощо. Це сприяло широкому впровадженню хмарних сервісів, які дозволяють ефективно зберігати, обробляти та масштабувати дані поза межами локального середовища.

Таблиця 1.2

Порівняння хмарних платформ

Параметр	Amazon Web Services (AWS)	Microsoft Azure	Google Cloud Platform (GCP)
Рік запуску	2006	2010	2008
Позиціонування	Найбільш зріла та масштабована платформа	Добре інтегрується з продуктами Microsoft	Орієнтація на машинне навчання та аналітику
Сервіси для зберігання	Amazon S3 – об'єктне сховище з підтримкою API, резервного копіювання, версіонування	Azure Blob Storage – добре працює з екосистемою Windows	Google Cloud Storage – оптимізовано для Big Data
Інтеграція	Підтримує Node.js, Python, Java, а також CI/CD, Docker, Kubernetes	Тісно пов'язаний із Visual Studio, .NET, Windows Server	Сильна підтримка Kubernetes, інтеграція з BigQuery
Безпека	Гнучка система прав доступу (IAM), шифрування, політики контролю	Підтримка Active Directory, шифрування на рівні файлової системи	Надійна система контролю доступу, автоматичне шифрування
Ціноутворення	Оплата за використання, гнучкі моделі	Безкоштовний рівень, тарифи з фокусом на підприємства	Прозора модель з безкоштовними лімітами для розробників
Підходить для	Високонавантажених систем, стартапів, великих продуктів	Компаній, які вже використовують Microsoft	Проектів з акцентом на аналітику та ШІ

(розроблено автором)

Хмарні платформи забезпечують стабільну роботу застосунків навіть при великій кількості запитів, підтримують автоматичне масштабування, резервне копіювання та гнучке керування доступом. Наприклад, під час масового завантаження зображень система зберігає продуктивність завдяки розподіленій інфраструктурі.

Найпоширенішими хмарними платформами у веб-розробці є Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP). Їхні ключові особливості наведено в таблиці 1.4.

Таблиця 1.3

Порівняння хмарних платформ

Параметр	Amazon Web Services (AWS)	Microsoft Azure	Google Cloud Platform (GCP)
Рік запуску	2006	2010	2008
Позиціонування	Найбільш зріла та масштабована платформа	Добре інтегрується з продуктами Microsoft	Орієнтація на машинне навчання та аналітику
Сервіси для зберігання	Amazon S3 –об'єктне сховище з підтримкою API, резервного копіювання, версіонування	Azure Blob Storage – добре працює з екосистемою Windows	Google Cloud Storage – оптимізовано для Big Data
Інтеграція	Підтримує Node.js, Python, Java, а також CI/CD, Docker, Kubernetes	Тісно пов'язаний із Visual Studio, .NET, Windows Server	Сильна підтримка Kubernetes, інтеграція з BigQuery
Безпека	Гнучка система прав доступу (IAM), шифрування, політики контролю	Підтримка Active Directory, шифрування на рівні файлової системи	Надійна система контролю доступу, автоматичне шифрування
Ціноутворення	Оплата за використання, гнучкі моделі	Безкоштовний рівень, тарифи з фокусом на підприємства	Прозора модель з безкоштовними лімітами для розробників
Підходить для	Високонавантажених систем, стартапів, великих продуктів	Компаній, які вже використовують Microsoft	Проектів з акцентом на аналітику та ШІ

(розроблено автором)

Сучасні веб-проекти все частіше потребують не лише серверної логіки, а й зручного механізму для роботи з файлами – зображеннями, уривками, додатковими матеріалами. Хмарні сервіси стали стандартним рішенням: вони знімають обмеження локального зберігання, спрощують доступ і дозволяють зосередитися на функціональності застосунку замість інфраструктурних питань.

1.2 Аналіз особливостей розробки систем електронної комерції

У сучасному світі електронна комерція стала частиною багатьох галузей, у тому числі й книжкового бізнесу. Більшість споживачів надають перевагу

зручності онлайн-покупок, що зумовлює потребу у якісних веб-платформах для продажу книг. Інтернет-магазин для книжкового бізнесу – це не лише вітрина для каталогу товарів, але й комплексний інструмент для взаємодії з клієнтами, просування асортименту та забезпечення ефективного процесу покупки.

Розробка таких веб-сайтів має свої особливості, які відрізняють її від створення інших типів онлайн-платформ. Це обумовлено необхідністю працювати з великими обсягами текстової інформації та забезпечувати користувачам простий і зрозумілий доступ до каталогу. Важливість інтеграції зручних інструментів для пошуку, сортування та навігації є ключовою, оскільки вони впливають на загальний досвід використання платформи. З огляду на це, можна виділити кілька основних аспектів, які визначають специфіку розробки інтернет-магазинів для книжкової індустрії:

- широкий каталог і фільтрація за характеристиками (магазини книг часто мають величезний асортимент, тому потребують ретельної системи категоризації та фільтрації. Наприклад, можливість шукати книги за жанром, автором, серією, видавництвом, форматом);
- додаткові деталі товару: (важливі такі характеристики, як опис сюжету, фрагмент для ознайомлення, рецензії, рейтинг. На відміну від загальних товарів, де інформація може бути мінімальною, для книг такі деталі мають суттєве значення для покупців);
- рецензії та рейтинги від читачів (важлива функція для книжкових сайтів – це розділи з рецензіями, де користувачі можуть залишати власні відгуки та оцінки);
- різноманітність форм подачі контенту (на відміну від звичайних товарів, книги можуть мати різні формати, як-от електронний варіант, аудіо чи паперове видання, що вимагає спеціальних опцій для вибору формату та відповідного способу доставки).

Проте окремо варто виділити формат онлайн-торгівлі, яким користувався кожен – маркетплейси. Якщо класичний онлайн-магазин представляє одну компанію – продавця, то маркетплейс – це багатостраноння платформа, яка

об'єднує кілька продавців, брендів або компаній. У такій моделі власник платформи не обов'язково володіє товарами, а виступає як посередник, який забезпечує технічну інфраструктуру.

Нерідко маркетплейси плутають зі звичайним інтернет-магазином, хоча між ними є принципові відмінності. У маркетплейсах усі товари виставляють сторонні продавці, а сам сервіс надає лише інструменти: особистий кабінет, систему публікації товарів, методи оплати або і методи доставки [5]. Ще однією відмінністю є те, що звичайні користувачі мають значно ширший вибір, оскільки продукція надходить від різних власників. Один і той самий товар може бути представлений кількома продавцями, але умови покупки – ціна, швидкість доставки, якість обслуговування – залежать виключно від конкретного продавця. Це створює здорову конкуренцію і дозволяє покупцю обирати найбільш вигідний варіант та породжує велику конкуренцію [6].

Залежно від бізнес-моделі, маркетплейси поділяють на кілька основних типів. Найпоширенішими є моделі B2C (business-to-consumer), де компанії продають товари кінцевим споживачам (приклад – Amazon); C2C (consumer-to-consumer), у яких приватні особи торгують між собою (наприклад, OLX); та B2B (business-to-business), де одна компанія продає товари або послуги іншим організаціям (як-от Alibaba). Також існують вертикальні маркетплейси, які спеціалізуються на конкретному сегменті (наприклад, книги, одяг, техніка), і горизонтальні – що охоплюють широкий асортимент у різних категоріях [7].

Як писалося вище, на відміну від звичайного онлайн-магазину, де всі товари належать одній компанії, маркетплейс функціонує як посередник, який організовує всі процеси – від пошуку товару до його доставки. Завдяки цьому модель охоплює більше типів користувачів і вимагає розширеної логіки. Це зумовлює необхідність складнішої структури кабінетів, багаторівневої модерації, а також аналітики по товарах і транзакціях. До особливостей таких систем можна віднести:

- підтримку великої кількості товарів і продавців – це вимагає швидкого пошуку, розумної навігації та стабільної структури категорій;

- гнучке оновлення цін і наявності – оскільки кожен продавець самостійно керує своїм асортиментом, платформа повинна реагувати на зміни миттєво;
- підтримку кількох способів оплати й варіантів доставки – усе має бути зручно з обох сторін: як для покупця, так і для продавця;
- прозору систему оцінок і коментарів, яка формує довіру між учасниками платформи;
- гарантії безпеки та врегулювання конфліктів – маркетплейс часто виконує роль арбітра і має забезпечувати повернення коштів, компенсації, захист особистих даних.

У реальних проєктах добре показала себе стратегія поетапної побудови: спочатку чітко окреслюється, що саме буде на платформі, які ролі матимуть користувачі, як відбуватиметься монетизація, які зовнішні сервіси будуть підключені. Уже на цьому етапі важливо закладати гнучкість: система має зростати разом з потребами проєкту, не ламаючись при кожному оновленні. Це стосується і архітектури, і структури бази даних, і того, як працюють окремі модулі. Успішні маркетплейси зазвичай мають незалежні блоки – каталог, аналітика, оплати, служба підтримки – які легко масштабуються або замінюються без ризику для всієї системи [8].

Незалежно від формату, кожна платформа – це набір рішень, який має відповідати реальним очікуванням користувачів і бізнесу. Саме такий підхід дозволяє закласти фундамент для стабільної роботи й подальшого розвитку, що є важливим у конкурентному середовищі цифрової комерції.

1.3 Аналіз механізмів обміну книг в онлайн-сервісах

Окрім традиційної моделі купівлі-продажу, в останні роки набули поширення сервіси, що дозволяють користувачам обмінюватися книгами напряму або через посередництво платформи [9]. Такі системи формують новий рівень взаємодії між читачами, сприяють розвитку читацьких спільнот та дають

змогу дати книгам друге життя. Сутність обміну полягає у переході від одностороннього споживання до двостороннього діалогу між учасниками платформи, де кожен водночас виступає і як отримувач, і як донор книжкового ресурсу.

Найпростішою та найбільш інтуїтивною моделлю є прямий обмін типу "книга на книгу". У такому форматі двоє користувачів домовляються про рівнозначну заміну одного видання на інше. Сервіс, що підтримує цю модель, зазвичай надає інструменти для створення особистої бібліотеки, а також фільтрації потенційних обмінів. Попри свою очевидну зручність, ця модель вимагає високого рівня взаємної згоди, а тому не завжди ефективна для великих спільнот, де інтереси читачів не завжди співпадають.

У відповідь на цю обмеженість виникла модель "Хочу / Пропоную", яка дозволяє розділити запити користувача на два незалежні списки: бажаних книг і тих, які готовий передати. Це відкриває можливості для непрямих обмінів – так званих "ланцюгових", коли книга передається через кількох посередників, але зрештою кожен отримує бажане. Деякі сервіси навіть реалізують автоматизоване зіставлення побажань із наявними пропозиціями, що значно спрощує процес.

Крім того, є ще один підхід – кредитна система, в якій користувачі отримують внутрішні бали або "кредити" за передачу книг іншим. Ці кредити згодом можна обміняти на книги, які пропонують інші учасники. Перевагою цієї моделі є гнучкість: не потрібно шукати прямого партнера для обміну. Водночас система вимагає технічної реалізації електронного балансу, механізму накопичення балів та контролю за дотриманням домовленостей.

Окремий тип ініціатив становить BookCrossing – модель безадресного обміну, заснована на ідеї публічного розповсюдження книг із можливістю відстеження їх переміщення. Книга, зареєстрована на спеціалізованому сайті, отримує унікальний ідентифікатор, за допомогою якого її подальший шлях можна зафіксувати у системі. Такий формат не передбачає адресного отримувача, а також не вимагає реєстрації з боку того, хто книгу знайшов, однак дозволяє спостерігати динаміку циркуляції книжкового ресурсу в просторі.

Також у низці міст діють локальні ініціативи – пункти безкоштовного обміну або "книжкові шафи", які часто мають онлайн-мапи з відмітками місць. Їхній функціонал не завжди має цифрову платформу в традиційному сенсі, однак вони інтегруються у загальну екосистему обміну. Користувач може дізнатись про наявність точки, перевірити залишені книги або навіть зарезервувати екземпляр онлайн.

Також, деякі онлайн-маркетплейси включають обмін як додаткову функцію. У таких системах поряд із опцією купівлі або продажу є можливість вказати готовність до обміну. Часто це реалізується через позначки "обмін можливий" у товарних оголошеннях, із подальшим листуванням між учасниками.

Одним із прикладів українського онлайн-сервісу, який реалізує обмін книгами поряд із продажем, є платформа Glodibi Marketplace. Цей додаток дозволяє користувачам переглядати книги, доступні до обміну, сортувати їх за жанрами, станом, ціною, а також ініціювати взаємодію через кнопки дій на сторінці товару.

Прикладом реалізації механізму обміну книг у межах українського онлайн-маркетплейса є платформа Glodibi (рис. 1.2), де обмін інтегровано як окрему форму взаємодії поряд із продажем. Суть реалізації полягає в тому, що користувач під час додавання книги до системи може обрати варіант «Обмін». У такому випадку оголошення відображається в загальному каталозі з відповідною позначкою, а сама книга стає доступною для пошуку іншими користувачами.

Запит на обмін ініціюється в момент, коли інший користувач обирає чужу книгу та вказує власну як зустрічну пропозицію. Після цього система формує замовлення з типом «обмін», яке надходить до власника книги. Він має два дні на підтвердження або відхилення. У разі прийняття замовлення книга автоматично вилучається з доступного каталогу, а статус змінюється на «Готово до відправки». Відправник зазначає обрану службу доставки та трек-номер.

Маркетплейс
Glodibi

Головна Книги Сторінки

Головна > Книги

Всього: 588

Сортувати за: Нещодавно додані Показати: Усі книги

 Емоційний Шантаж Сюзанн Форвард Емоційний Шантаж Як працює Google (м'як... Девід Сімон Як працює Google Ерік Шмідт Девід Сімон Як працює Google Лео Дашвар #Галубезголови Лео Дашвар Ідеальний шторм Джек Райгардт Тінь над Інсмутом Говард Лаварафт Симпатик Вет Тіам Нгуєн	 Чарівна ти Чак Поланік Чарівна ти Страви за 10 хвилин. 10... Гордон Рамзі Страви за 10 хвилин 48 законів влади Роберт Грін 48 законів влади Роберт Грін Суджені Девід Марр Суджені Девід Марр Пес на ім'я Мані, або А... Б. Шефер Пес на ім'я Мані, або А... Б. Шефер Розмови з донькою про ЕКОНОМІКУ Валуфакіс Яніс
--	--

Рис. 1.2. Каталог обмінних книг на платформі Glodibi Marketplace

Джерело: Розроблено із використанням [10]

Унікальність механізму Glodibi полягає в тому, що платформа дозволяє реалізувати неспарений обмін, тобто користувач не зобов'язаний вести переговори напряму, як у моделі «хочу/пропоную», натомість пропонує книгу з власного списку і чекає на відповідь. Водночас обидва учасники отримують повну відповідальність за обмін: Glodibi не виступає гарантом, а лише технічно фіксує угоду. Сервіс прямо зазначає, що не гарантує безпечного обміну, тому вся відповідальність за дотримання домовленостей лежить на користувачах.

Вимоги до успішного обміну формалізовано через ряд обмежень. Зокрема, користувач повинен мати хоча б одну книгу в активному статусі на обмін, а книга, на яку він робить запит, не повинна бути з його ж акаунту або зі списку

вже прочитаних ним екземплярів. Це дозволяє уникнути конфліктів інтересів і забезпечити мінімальну логіку перевірки умов.

Крім обміну, маркетплейс Glodibi підтримує також прямий продаж книг, що дозволяє користувачам купувати товари без взаємного погодження. У процесі додавання книги продавець вказує ціну, опис, стан. Покупець додає товар до кошика, оформлює замовлення, після чого воно надходить продавцю, який може його прийняти або відхилити. У випадку підтвердження діє та сама логіка доставки, як і при обміні.

Отже, Glodibi реалізує функціональність, близьку до класичної C2C-моделі з опцією обміну та продажу, де платформа виступає інструментом пошуку, реєстрації та фіксації запитів, але не втручається в логістику або контроль дій. Це наближає сервіс до логіки, застосовуваної в peer-to-peer-системах, де ефективність залежить від довіри й активності спільноти.

Загалом, попри різні підходи до організації обміну, ключовими викликами залишаються питання модерації, контролю достовірності пропозицій, а також організація процесу передачі книг, особливо у випадку міжміського або міжнародного обміну. Водночас, інтеграція зазначених механізмів у веб-застосунки книжкового спрямування сприяє розширенню функціоналу платформи, активізації користувацької взаємодії та підтримці сталого використання друкованої продукції.

1.4 Вимоги до функціоналу маркетплейса

На етапі визначення вимог до функціональності майбутнього веб-сайту книжкового маркетплейса слід враховувати специфіку предметної області, яка передбачає як класичну модель електронної комерції, так і можливість обміну книгами між користувачами. Система повинна відповідати вимогам безпеки та зручності, задовольняючи потреби як користувачів, так і адміністратора.

Зареєстрованим користувачам має надаватися персоніфікований доступ до розширених функцій. Після авторизації вони зможуть редагувати профіль,

додавати книги на обмін або продаж, шукати літературу, надсилати запити на обмін і переглядати історію взаємодій. Авторизація відбуватиметься через токени доступу для захисту персональних даних.

Каталог повинен забезпечувати перегляд книг із можливістю фільтрації за автором, назвою, форматом і типом пропозиції. Кожна книга матиме окрему сторінку з описом, обкладинкою та інформацією про доступність. Книги без модерації не мають відображатися у публічному доступі.

Функція обміну передбачає надсилання запиту з вибором однієї зі своїх книг. Інший користувач може підтвердити або відхилити запит. У разі згоди обидві книги деактивуються, а сторонам відкривається доступ до контактних даних для організації доставки.

Для інформування користувачів система повинна включати модуль сповіщень про ключові події: нові запити, зміну статусу, результат взаємодії. Сповіщення зберігатимуться в особистому кабінеті.

Продаж книг реалізовуватиметься через класичну модель: додавання до кошика, вибір формату, оформлення замовлення, після чого інформація потрапляє в історію покупок.

Адміністративна частина має містити модуль модерації, що дозволить адміністратору перевіряти нові книги, керувати публікацією, розглядати запити та працювати з даними користувачів.

До технічних вимог належить адаптивна верстка, взаємодія клієнта і сервера через REST API з підтримкою JSON, захист через HTTPS і дотримання конфіденційності. Також важливо передбачити оптимізацію роботи з мультимедійними файлами.

Висновки до розділу 1

У першому розділі було проведено теоретичний аналіз особливостей веб-розробки для створення сучасного книжкового маркетплейса. Розглянуто базові поняття та принципи функціонування веб-додатків, зокрема архітектуру клієнт-

сервер, особливості фронтенду і бекенду, а також роль HTML, CSS, JavaScript і фреймворків, таких як React у формуванні зручного інтерфейсу. Також було обговорено хмарні платформи зберігання, які стали частиною масштабованих веб-систем.

На прикладі книжкових інтернет-магазинів та маркетплейсів описано вимоги до функціоналу: підтримка великого каталогу, рецензії, вибір формату, модерація товарів, підтримка кількох продавців. Визначено відмінності між класичними інтернет-магазинами та платформами з моделлю С2С.

Також детально проаналізовано моделі обміну книг, що дедалі частіше інтегруються у книжкові онлайн-сервіси. Розглянуто прямий обмін, формат «хочу/пропоную», кредитну систему, а також кейс української платформи Glodibi, яка поєднує продаж і обмін книг в єдиному цифровому середовищі.

Отримані результати формують базу для подальшої роботи над проектом, зокрема для розробки книжкового інтернет-маркетплейса, який буде поєднувати технічну досконалість із врахуванням потреб користувачів і тенденцій ринку.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ТА ОБМІНУ КНИГ

2.1 Проектування архітектури веб-сайту

Проектування архітектури веб-сайту є важливим етапом у розробці будь-якого веб-додатку. Це забезпечує структурованість, ефективність і масштабованість проекту, дозволяючи кожному компоненту системи виконувати свою роль без збоїв. В архітектурі веб-сайту можна виділити три основні компоненти: frontend, backend та база даних (збереження інформації). Опис цих компонентів надає зрозуміти, як функціонують зв'язки між ними. Архітектура проекту також буде складатися з трьох частин.

Проаналізувавши дані про середовища, призначення та сумісність для проекту книжкового маркетплейса було обрано бібліотека React, за допомогою якої буде реалізована клієнтська частина. У контексті книжкового магазину саме цей компонент буде відповідати за відображення каталогу книг, кошика, форми реєстрації та інших елементів інтерфейсу, з якими взаємодіє користувач. Перевагою React є можливість розбиття інтерфейсу на компоненти, що спрощує розробку, підтримку та масштабування.

Для серверної частини було обрано Node.js через його високу продуктивність і сумісність із JS мовою, яка використовується у React. Node.js дозволяє швидко та ефективно обробляти клієнтські запити, виконувати бізнес-логіку (наприклад, перевірка замовлень або авторизація користувачів) та передавати дані до, та з бази даних.

Для збереження даних була обрана PostgreSQL – реляційна база даних, яка визначається надійністю та багатим функціоналом. У базі буде зберігатися записи про книги (назва, автор, жанр, опис), інформація про користувачів (логіни, паролі) дані про замовлення та їх статус. Також PostgreSQL була обрана

через те, що забезпечує високу продуктивність при роботі з великими обсягами даних, що є важливим для магазину з великим асортиментом книг.

Крім того, під час проектування архітектури веб-сайту передбачається використання хмарної платформи Amazon Web Services (AWS), зокрема сервісу Amazon S3, для зберігання зображень книжок. Такий підхід дозволить організувати розділення файлів за статусами (наприклад, «очікує модерації», «схвалено», «відхилено») та уникнути навантаження на основний сервер. Файли зберігатимуться у хмарному середовищі, а в базі даних зберігатимуться лише посилання на відповідні об'єкти.

У процесі проектування архітектури веб-сайту також важливо розуміти, які дії можуть виконувати користувачі системи та як саме вони взаємодіють із різними компонентами. За допомогою використання діаграм варіантів (Use Case Diagram) можна візуалізувати ці взаємодії, визначити основні ролі користувачів та їх можливості в рамках функціональності системи. Це забезпечує чітке розуміння функціональних вимог, що спрощує розробку як frontend, так і backend компонентів.

У контексті розробки веб-сайту книжкового маркетплейса діаграма варіантів використання (рис. 2.1) демонструє можливості трьох основних категорій користувачів: гостьового користувача, авторизованого користувача та адміністратора. Кожна з цих ролей має набір завдань, які вони можуть виконувати у системі. Наприклад, гостьовий користувач може переглядати каталог книг та шукати товари, у той час як авторизований користувач має доступ до більш розширеного функціоналу, такого як додавання книг у кошик, оформлення замовлення та перегляд історії покупок, обміну. Адміністратор має можливість редагувати контент каталогу та управляти заявками на додавання книг.

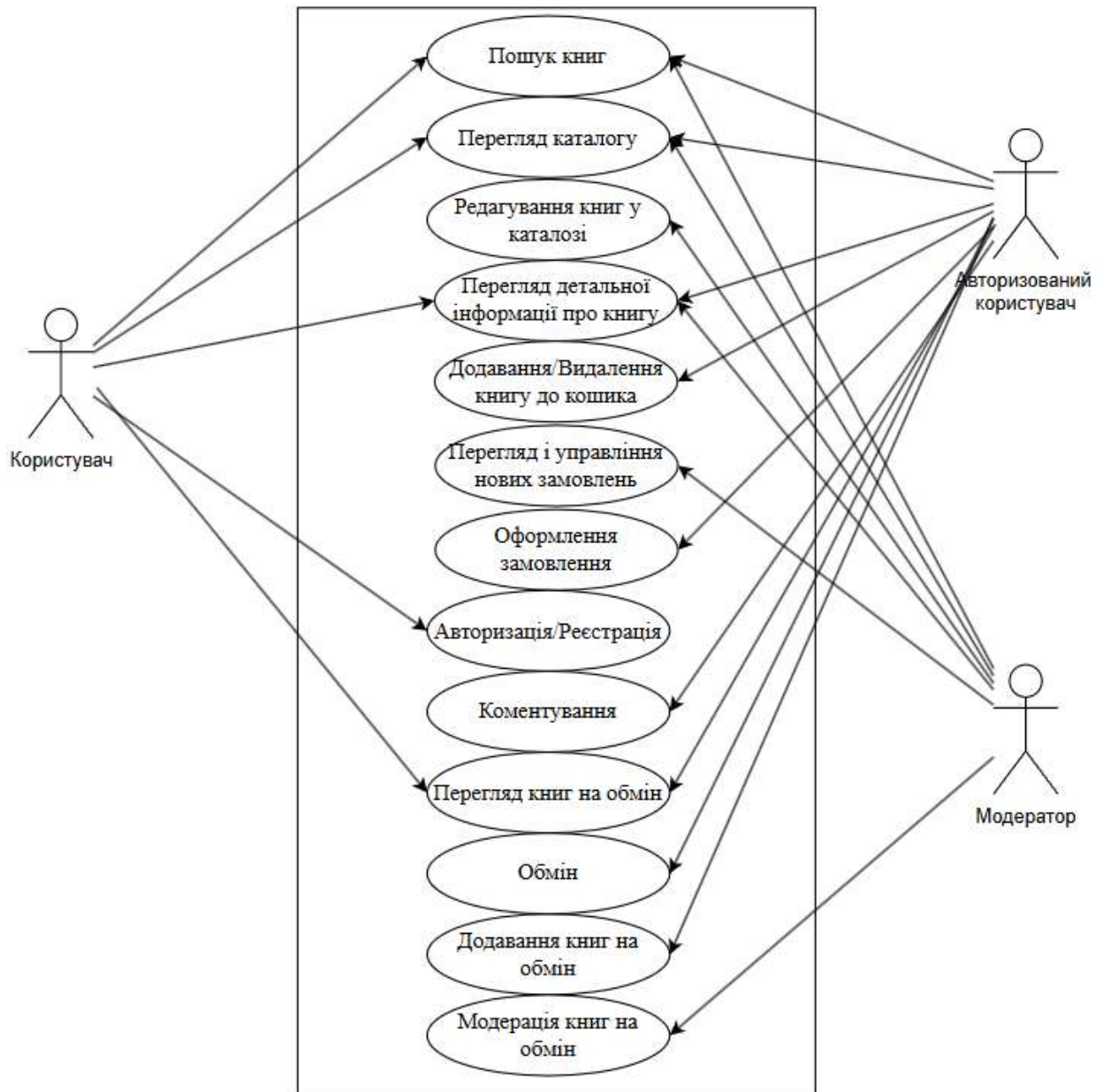


Рис. 2.1. Діаграма варіантів використання

(розроблено автором)

Діаграма варіантів використання, наведена вище, ілюструє функціональні можливості для кожного типу користувача:

1. Користувач (гостьовий доступ). Основні функції включають:
 - 1.1. пошук книг;
 - 1.2. перегляд каталогу;
 - 1.3. перегляд детальної інформації про книгу (автор, рік видання, кількість сторінок, розмір, опис книги);

- 1.4. авторизація;
- 1.5. перегляд каталогу з книг для обмін-продажу.
2. Авторизований користувач. Основні функції включають:
 - 2.1. пошук книг;
 - 2.2. перегляд каталогу;
 - 2.3. перегляд детальної інформації про книгу (автор, рік видання, кількість сторінок, розмір, опис книги);
 - 2.4. додавання/Видалення книг в кошику;
 - 2.5. оформлення замовлення;
 - 2.6. авторизація/Реєстрація;
 - 2.7. коментування.
 - 2.8. перегляд каталогу з книг для обмін-продажу;
 - 2.9. обмін книжками;
 - 2.10. додавання книг на обмін.
3. Адміністратор. Основні функції включають:
 - 3.1. пошук книг;
 - 3.2. перегляд каталогу;
 - 3.3. редагування книг у каталозі (Додавання, видалення, зміна інформації про книгу);
 - 3.4. перегляд і управління нових замовлень.

Після визначення функціональних можливостей для кожної ролі користувачів варто детальніше описати їх взаємодію із системою. Для цього створено діаграму діяльності (Activity Diagram), яка ілюструє процеси для кожної ролі. У даній діаграмі (рис 2.2) відображено послідовність дій, які може виконувати користувач, починаючи з перегляду каталогу книг і закінчуючи можливістю створити обліковий запис для доступу до розширеного функціоналу.

Діаграма діяльності для користувача, наведена вище, ілюструє такі можливі дії гостьового користувача:

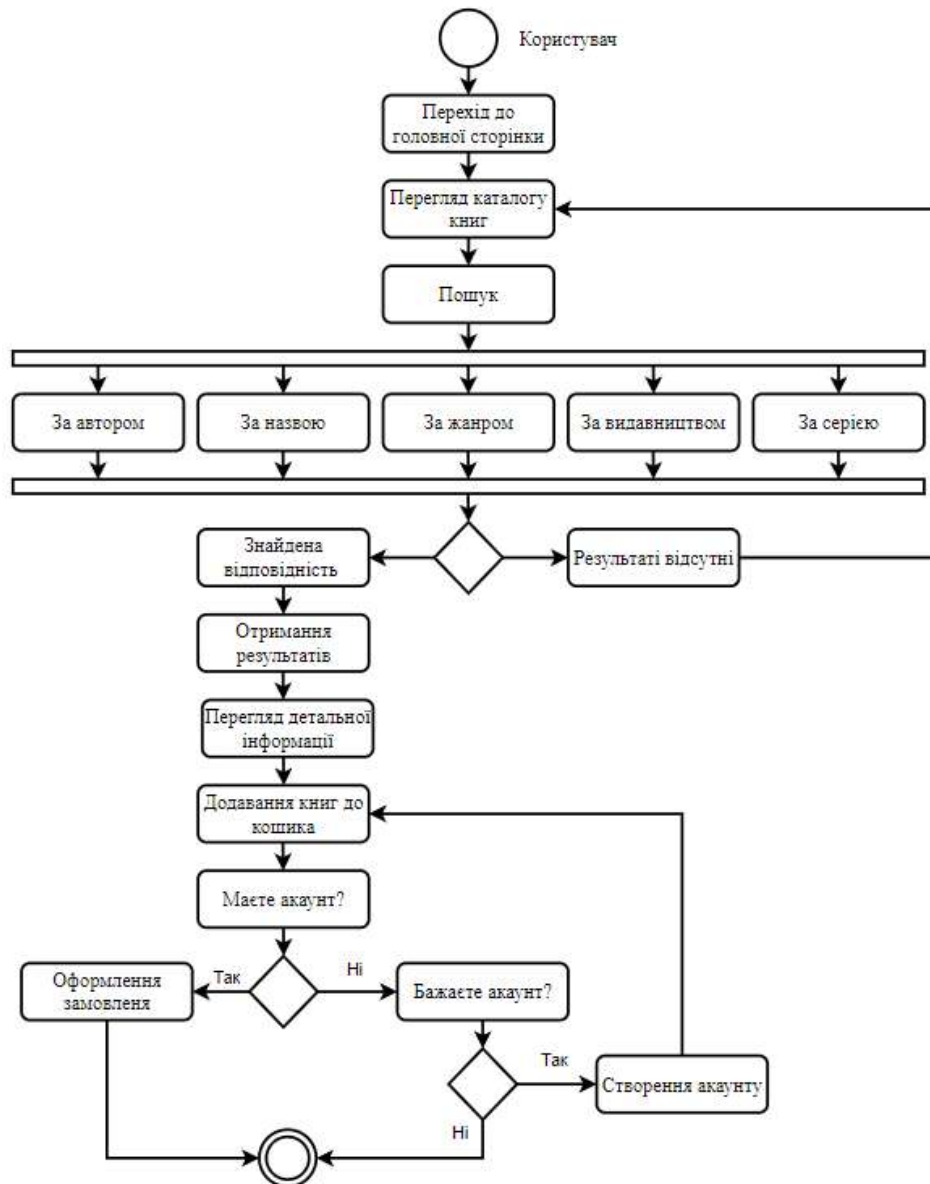


Рис. 2.2. Діаграма діяльності користувачів

(розроблено автором)

- користувач переходить на головну сторінку сайту;
- має можливість переглядати каталог книг, шукати конкретні книги за різними критеріями (автор, назва, жанр тощо);
- після знаходження книги, гість може ознайомитися з детальною інформацією про неї;
- для оформлення замовлення чи додавання книги в кошик користувачеві буде запропоновано створити обліковий запис;

- якщо користувач не хоче зареєструватися, він може закінчити роботу на сайті.

Для користувачів, які вже зареєстровані, мають такі можливості:

- перегляд кошику та додавання або видалення книг з нього;
- оформлення замовлення через інтерактивну форму, яка вимагає заповнення основних даних для доставки;
- залишити коментар на сторінці з книгою.

Крім користувача, в системі є права адміністратора, основні функції наведені на рисунку 2.3. Адміністратор має особливий статус у системі, який надає права виконувати дії, недоступні звичайним користувачам.

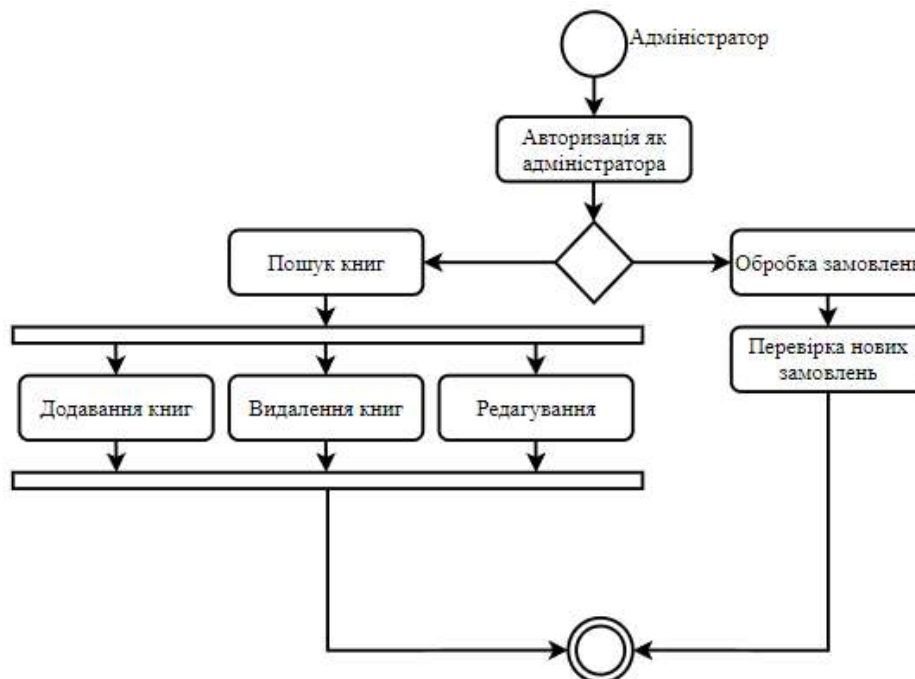


Рис. 2.3. Діаграма діяльності для адміністратора

(розроблено автором)

Основні функції адміністратора включають:

- редагування каталогу. Адміністратор може додавати нові книги з повним набором даних (назва, автор, опис, жанр, видавництво тощо)
- змінювати вже існуючу інформацію книг;
- видаляти книги з каталогу;

- адміністратор може переглядати замовлення в активному списку;
- може змінювати статус, наприклад, від «очікує обробки», до «виконано».

Оскільки розроблюваний веб-сайт має розширену функціональність маркетплейса, важливо окремо спроектувати процеси, які стосуються взаємодії між користувачами у форматі обміну книгами. Зокрема, одним із етапів є розміщення книги на обмін, що потребує погодження з боку адміністратора.

Для зрозумілості було створено блок-схему в нотації BPMN (рис. 2.4), яка ілюструє етапи від моменту подання запиту до фінального рішення адміністратора.

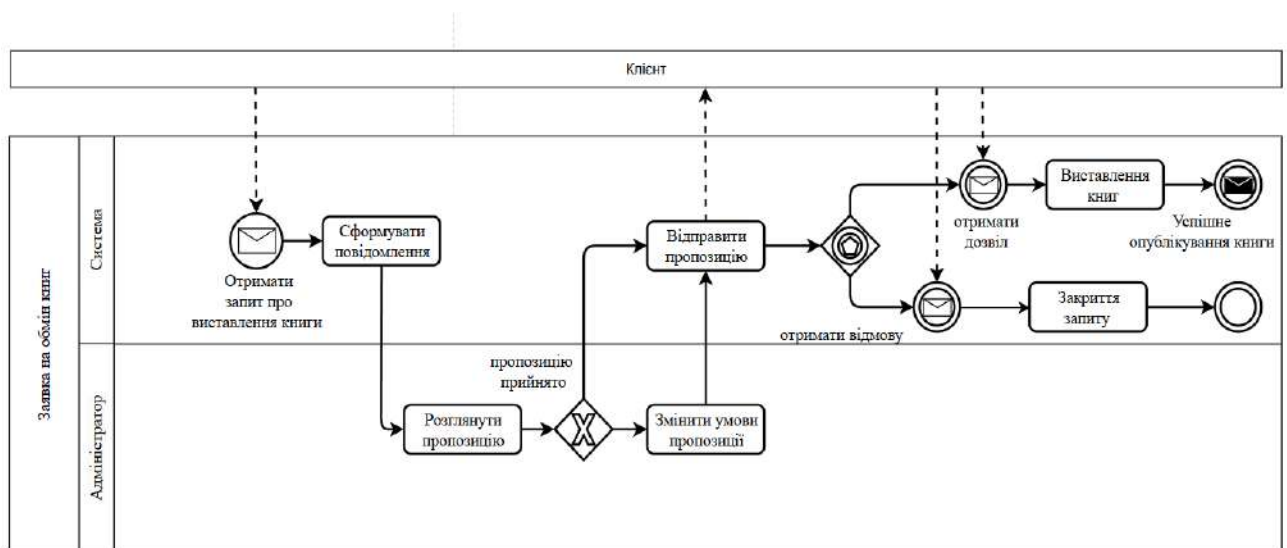


Рис. 2.4. Блок-схема заявки на обмін

(розроблено автором)

Система автоматично формує сповіщення, після чого адміністратор розглядає заявку: у разі потреби може запропонувати зміни або схвалити одразу. Якщо пропозицію ухвалено – книга публікується, якщо ж ні – заявка закривається з відповідним повідомленням.

Після публікації книги стає можливою друга фаза – безпосередній обмін між користувачами. Вона описана на блок-схемі (рис. 2.5), яка охоплює основні дії: вибір книги, надсилання запиту, розгляд іншою стороною, ухвалення або

відмова та обмін контактними даними для доставки. Схема відображає двосторонню взаємодію і враховує обидва можливі варіанти відповіді.

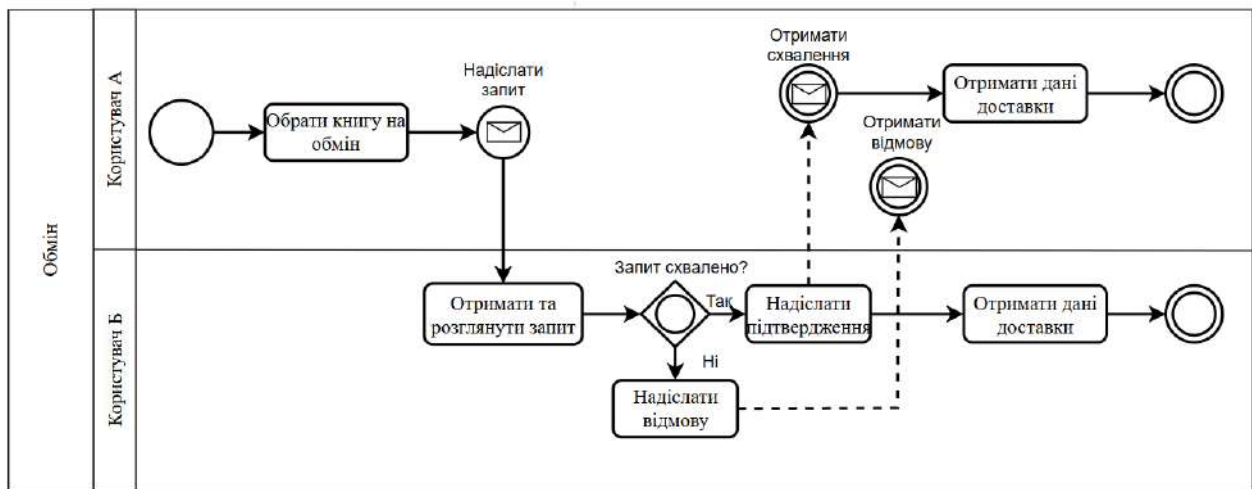


Рис. 2.5. Блок-схема обміну між користувачами

(розроблено автором)

Проектування архітектури веб-сайту книжкового маркетплейса дозволило зрозуміти основні компоненти системи: клієнтську частину, серверну логіку, базу даних, тощо. Завдяки поєднанню класичної моделі торгівлі з механізмом обміну книгами було створено архітектуру, що враховує потреби як звичайних користувачів, так і адміністратора. Візуалізація через діаграми варіантів використання та діяльності допомогла описати ролі користувачів і ключові сценарії взаємодії, включаючи модерацию, обміни та систему сповіщень.

2.2 Проектування інтерфейсу користувача

Книжкова індустрія активно адаптується до онлайн-середовища, і успішність веб-платформи значною мірою залежить від якості її інтерфейсу. Інтерфейс має бути інтуїтивно зрозумілим, зручним у навігації та естетично привабливим.

Проектування UI – це створення візуального середовища для взаємодії користувача з додатком. Основні елементи: головна сторінка, каталог, профіль

користувача та сторінка замовлення формують загальне враження про сайт. Зокрема, головна сторінка виконує роль "вітрини", демонструючи асортимент і функціональні можливості.

Під час розробки дизайну враховується специфіка книжкового ринку: потреба у швидкій навігації за жанрами, зручному пошуку, фільтрах за автором чи видавництвом. Усе це має сприяти зручності та позитивному користувацькому досвіду.

Першим етапом є створення макетів сторінок, вони допомагають логічно розмістити елементи, визначити структуру та продумати послідовність дій користувача. Узагальнені макети дозволяють сформувати інтерфейс, що відповідає очікуванням і підвищує ефективність використання сайту. Перший такий макет наведено на рисунку 2.6.

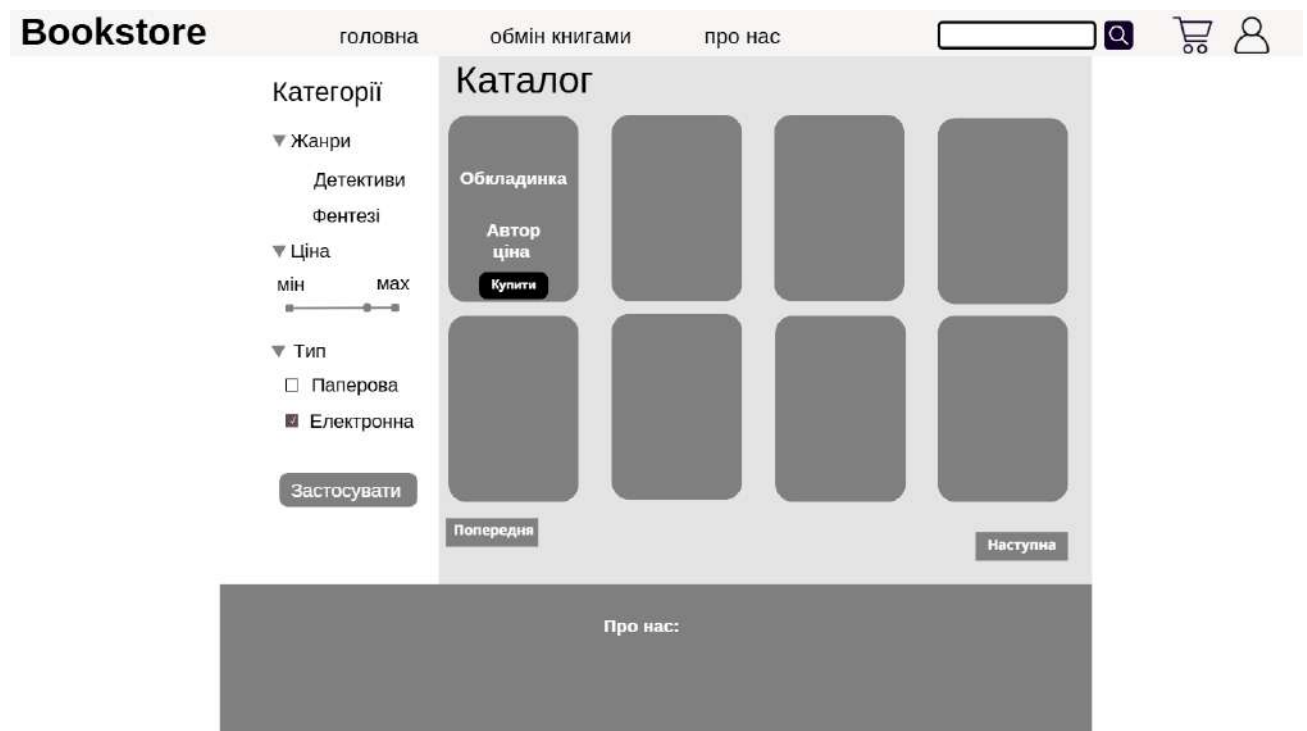


Рис. 2.6. Головна сторінка

(розроблено автором)

Головна сторінка є центральним елементом взаємодії користувача із сайтом і створює перше враження про платформу. Від її структури,

функціональності та візуальної привабливості залежить, чи залишиться користувач на сайті і чи виконає цільові дії, такі як пошук, перегляд або покупка книг.

Макет головної сторінки, представлений на зображенні, демонструє зрозумілу структуру можливих майбутніх елементів:

- шапка сторінки. У верхній частині розташовані: логотип, що виконує роль навігаційного елементу для повернення на головну сторінку; кнопки головної сторінки, обміну та інформації про компанію, рядок пошуку для швидкого доступу до потрібних книг; навігаційне меню для переходу до каталогу, авторизації чи інших розділів сайту; кнопки авторизації або реєстрації, які дозволяють користувачам увійти у свій профіль.
- категорії книг. Нижче розташовані інтерактивні кнопки або списки жанрів, які дозволяють швидко знайти літературу потрібної тематики. Інтуїтивний підхід до їх розташування спрощує навігацію сайтом. У центральній частині сторінки представлений перелік книг. Кожна картка містить обкладинку, назву, автора та ціну. Картки інтерактивні: при кліку користувач переходить на сторінку конкретної книги.
- футер. У нижній частині буде розташовані контактні дані магазину, посилання на соціальні мережі, політику конфіденційності або допоміжну інформацію.

Після знайомства з головною сторінкою наступним важливим елементом є сторінка книжок, яка виконує роль детального представлення кожного продукту каталогу. Ця сторінка повинна надавати всю необхідну інформацію про книгу, допомагати користувачам прийняти рішення щодо покупки та заохочувати до додаткової взаємодії із сайтом, наприклад, перегляду рецензій чи рекомендацій.

Детальна сторінка книги є важливим елементом будь-якого книжкового онлайн-магазину. Її мета інформувати користувача, сприяти ухваленню рішення про купівлю завдяки зручному дизайну, візуальній привабливості та функціональності. Макет такої сторінки представлено на рисунку 2.7.

У верхній частині сторінки розміщено велике зображення обкладинки, що формує перше враження про товар і дозволяє краще розглянути його оформлення. Поруч розташовано назву книги, ім'я автора, ціну (за паперову та електронну версію) та кнопку «Купити». Нижче подано коротку характеристику з основною інформацією: жанр, кількість сторінок, мова тощо.



Рис. 2.7. Книжкова сторінка

(розроблено автором)

Опис книги представлено окремим блоком, де користувач може ознайомитися з анотацією та зрозуміти, чи відповідає книга його інтересам. Додатково передбачено розділ з повною характеристикою, що може містити такі дані, як ISBN, видавництво, рік видання та інші технічні параметри..

Модальне вікно кошика (рис. 2.8) є проміжним етапом між додаванням товарів та оформленням замовлення. Воно відкривається при натисканні на іконку кошика у хедері сайту та дозволяє швидко переглянути обрані книги без переходу на окрему сторінку.

У макеті вікно містить заголовок і перелік товарів у вигляді карток. Кожна картка включає обкладинку, назву книги, автора, тип (електронна/паперова), ціну за одиницю, кількість з можливістю редагування («+»/«-») та підсумкову вартість.



Рис. 2.8. Модальне вікно кошика

(розроблено автором)

У нижній частині розташовано блок з підрахунком загальної суми та кнопки: «Очистити» (видалення всіх товарів), «Продовжити» (закрити вікно без втрати даних) та «Оформити» (перехід до оформлення замовлення).

Модальне вікно підтримує динамічне оновлення без перезавантаження, інтуїтивну взаємодію та збереження стану. Якщо користувач не авторизований, при спробі оформити замовлення відкривається вікно входу або реєстрації (рис. 2.9).

Вікно об'єднує обидва режими в єдиному інтерфейсі з можливістю швидкого перемикання. У режимі входу вводиться e-mail і пароль, у режимі

реєстрації також номер телефону. Всі поля проходять клієнтську валідацію перед відправленням даних на сервер.

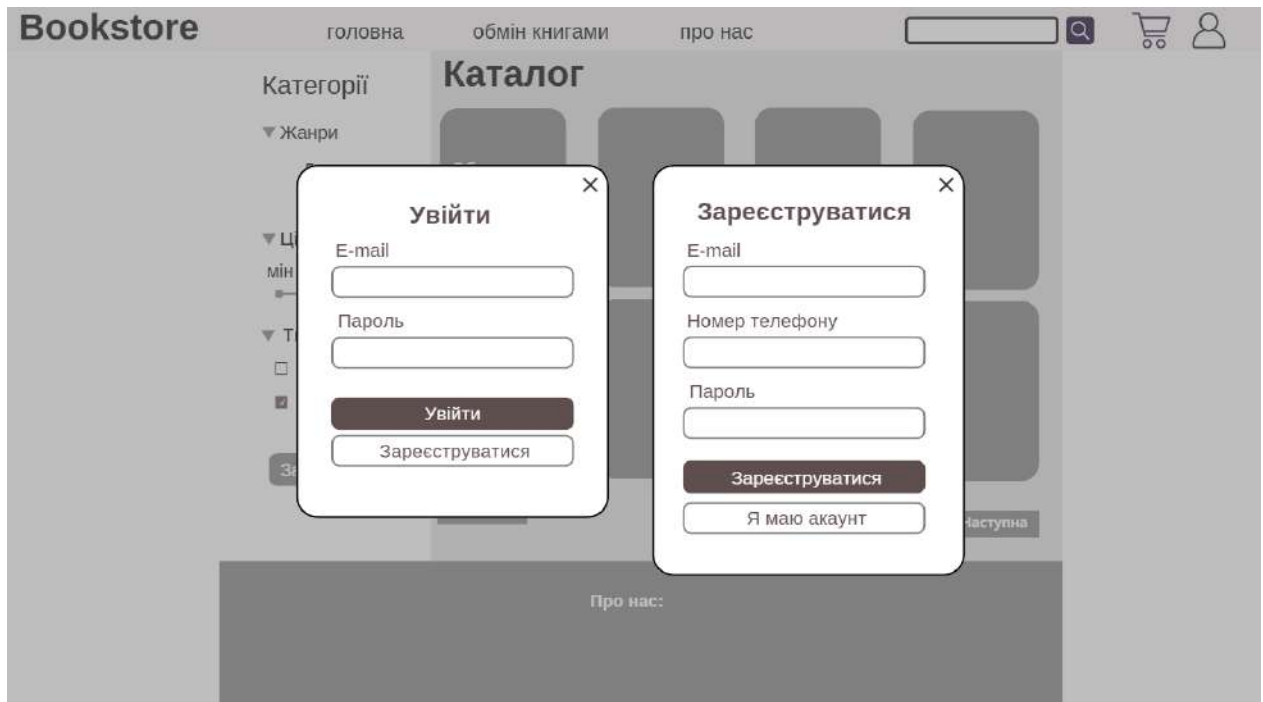


Рис. 2.9. Модальне вікно реєстрації та авторизації

(розроблено автором)

Після успішної авторизації або реєстрації користувач повертається до попередньої дії, наприклад, до кошика. Якщо після цього він натискає кнопку «Оформити замовлення», система перенаправляє його на відповідну сторінку (рис. 2.10).

Верхня частина містить заголовок «Огляд замовлення», що позначає поточний етап процесу. Нижче відображено список товарів, які були додані до кошика: кожна позиція подається у вигляді картки з назвою, типом книги, ціною, кількістю та загальною сумою.

У нижній частині сторінки розташовано блок із підрахунком загальної вартості та кнопка «Сплатити». Після її натискання дані замовлення надсилаються на сервер. У разі успішного оформлення система повідомляє користувача та перенаправляє його на головну сторінку.

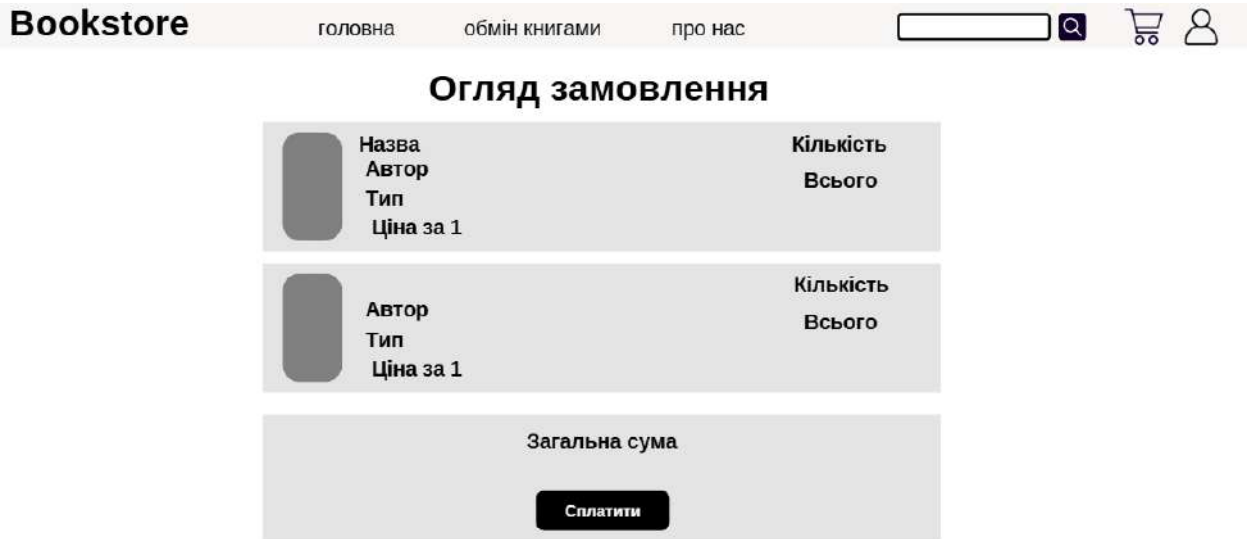


Рис. 2.10. Сторінка замовлення

(розроблено автором)

Сторінка адміністративної панелі (рис. 2.11) призначена для керування книжковим каталогом. На макеті представлено інтерфейс додавання нової книги, який дозволяє адміністратору швидко внести необхідну інформацію, що одразу зберігається у базі даних і стає доступною для користувачів.

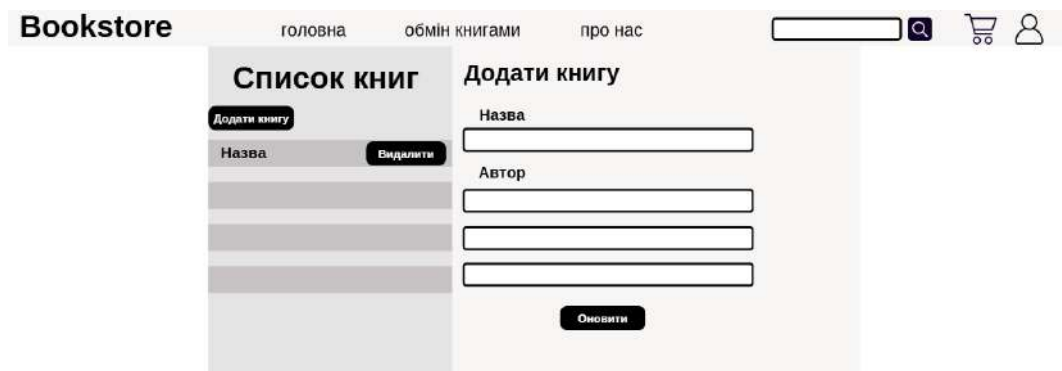


Рис. 2.11. Сторінка адміністративної панелі

(розроблено автором)

Центральне місце займає форма з полями для введення основних характеристик книги: назви, автора, ISBN, жанру, опису, року видання, формату (паперовий або електронний), ціни та обкладинки. Поруч із полями можуть відображатися підказки щодо формату введення (наприклад, для ISBN або зображення).

Також реалізовано можливість редагування вже доданих книг: адміністратор може обрати запис зі списку, змінити дані та натиснути кнопку «Оновити». Кнопка «Додати книгу» активує порожню форму для заповнення. Після підтвердження всі дані надсилаються на сервер, і, у разі успіху, книга автоматично з'являється в загальному каталозі.

2.3 Проектування інтерфейсу обмінного фонду

Розширення функціональності веб-сайту книжкового маркетплейса передбачало інтеграцію модуля обміну книгами між користувачами. Якщо початково система була орієнтована переважно на класичну модель електронної комерції, то із впровадженням механізму обміну основний акцент поступово змістився на розвиток взаємодії між користувачами.

Реалізація можливості обміну вимагала проектування окремих елементів інтерфейсу, які забезпечують простоту навігації, інтуїтивність процесу вибору книг для обміну та відстеження історії взаємодій. Інтерфейс мав охопити весь шлях користувача: від додавання власної книги до каталогу обміну до успішного погодження й здійснення обміну.

Кабінет користувача (рис. 2.12) створений для зручного керування основними діями на платформі. Через нього можна оновлювати особисті дані, додавати книги для обміну, переглядати історію замовлень, обмінів, коментарів і отримувати сповіщення.

З лівого боку розташоване меню, за допомогою якого можна переходити між основними розділами: "Профіль", "Додати книгу", "Історія замовлень", "Сповіщення", "Історія обміну" та "Історія коментарів". Така структура допомагає швидко знаходити потрібні функції без зайвих переходів.

У правій частині сторінки відображаються особисті дані користувача: ім'я, прізвище, електронна пошта, номер телефону, місто та місце відправлення. Ця інформація потрібна для оформлення замовлень і організації обміну книгами між користувачами після підтвердження запиту.

Однією з основних можливостей у кабінеті користувача є функція додавання книги для участі в обміні або продажу. Перехід до цієї функції здійснюється через натискання на кнопку "Додати книгу" у лівому меню.

На сторінці додавання книги (рис. 2.13) користувачеві пропонується заповнити форму, яка включає обов'язкові дані для створення нової пропозиції.

Рис. 2.12. Макет кабінету

(розроблено автором)

Поля для заповнення:

- «Назва» – вказується повна назва книги;
- «Автор» – ім'я автора книги;
- «Стан» – короткий опис фізичного стану книги (наприклад, "як нова", "гарний", "зі слідами використання");
- «Опис» – додаткова інформація про книгу або умови обміну/продажу;
- «Тип пропозиції» – вибір між варіантами "Продаж" або "Обмін";
- «Ціна» – вказується лише у разі вибору варіанту "Продаж";
- «Обкладинка» – можливість завантажити зображення книги.

Після заповнення всіх полів користувач натискає кнопку "Надіслати на модерацию", що автоматично створює заявку для адміністратора. Книга стає доступною для перегляду іншими користувачами тільки після проходження модерации.

Для забезпечення якості контенту на платформі книжкового маркетплейса було реалізовано ще один модуль модерации. Його основне завдання це перевірка усіх книг, які користувачі додають до обмінного фонду, перш ніж вони стануть доступними для перегляду іншими учасниками платформи.

The screenshot shows a web interface for adding a book. At the top, there is a navigation bar with the logo 'Bookstore' and links for 'головна', 'обмін книгами', and 'про нас'. To the right of the navigation bar are icons for search, shopping cart, and user profile. The main content area is titled 'Додати книгу' and contains several input fields: 'Назва', 'Автор', 'Стан', 'Опис', 'Ціна', and 'Завантажити файл'. There are also radio buttons for 'Продаж' and 'Обмін'. A large grey placeholder box labeled 'Обкладинка' is positioned to the right of the input fields. At the bottom of the form is a button labeled 'Надіслати на модерацию'.

Рис. 2.13. Макет додавання книги для обміну

(розроблено автором)

Після того як користувач додає книгу до розділу обміну, ця книга автоматично потрапляє до переліку заявок на модерацию. Адміністратор має змогу переглядати список усіх нових заявок у зручному форматі: кожна книга відображається у вигляді компактної картки з мінімально необхідною інформацією: обкладинкою, назвою, автором та унікальним ідентифікатором заявки (рис. 2.14).



Рис. 2.14. Сторінка перегляду заявок на обмінні книги
(розроблено автором)

Для перегляду повної інформації про заявку адміністратор натискає на відповідну картку. Після натискання відкривається модальне вікно з детальним описом книги: зображенням обкладинки, автором, назвою, типом пропозиції (обмін чи продаж), описом та станом книги (рис. 2.15).

У вікні модерації адміністратор має змогу прийняти або відхилити заявку:

- прийняття заявки призводить до автоматичного переміщення зображення книги з тимчасової папки у сховище підтверджених зображень, а також робить книгу доступною для інших користувачів у каталозі обміну;
- відхилення заявки супроводжується видаленням книги з бази даних та автоматичним очищенням відповідного зображення з тимчасового сховища.

Після реалізації особистого кабінету користувача, наступним логічним етапом стало створення інтерфейсу для роботи з каталогом книг на обмін та продаж.

Сторінка обмінних книг (рис. 2.16) є публічно доступною для всіх користувачів платформи. Вона призначена для перегляду всіх активних пропозицій книг, виставлених іншими учасниками спільноти. Основна мета

даної сторінки забезпечити доступ до доступних варіантів обміну, спростити процес пошуку потрібної книги.



Рис. 2.15. Модальне вікно модерації обмінної книги

(розроблено автором)

Інтерфейс сторінки побудовано у форматі сітки карток як і на основній сторінці. Кожна картка книги містить обкладинку, назву, ім'я автора та коротку інформацію про тип пропозиції ("на обмін" або "на продаж").



Рис. 2.16. Сторінка обмінних книг

(розроблено автором)

Натискання на картку відкриває детальну сторінку книги (рис. 2.17), де користувач може ближче ознайомитися з вибраною пропозицією. Тут подано

збільшене зображення обкладинки, назву, автора, опис книги, а також вказано фізичний стан і тип пропозиції: обмін або продаж.

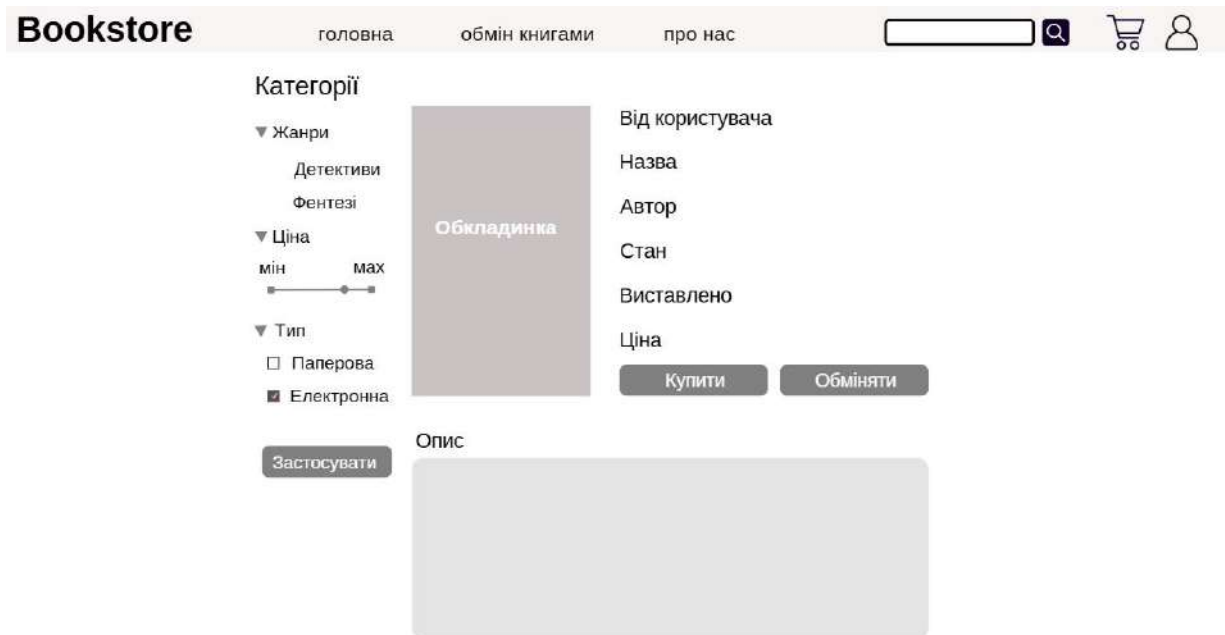


Рис. 2.17. Детальна інформація книги

(розроблено автором)

На детальній сторінці також відображається ім'я користувача, який виставив книгу. Це допомагає створити більш персоналізовану взаємодію і дає змогу орієнтуватися у пропозиціях під час вибору книги для обміну.

2.4 Проектування механізму запитів на обмін

Після проектування інтерфейсної частини особистого кабінету та каталогу обмінних книг, наступним кроком стало створення системи обміну між користувачами. Треба було спроектувати механізм, який дозволив би двом користувачам здійснити обмін книгами за взаємною згодою.

Логіка обміну буде побудована на основі кількох етапів: спершу користувач переглядає доступні книги на сторінці обмінного фонду та зацікавившись конкретною пропозицією, може ініціювати запит на обмін. Для цього йому пропонується обрати одну зі своїх книг, які попередньо були додані

до обмінного каталогу. Запит надсилається власнику іншої книги, який отримує відповідне сповіщення у своєму кабінеті.

Після розгляду запиту власник має змогу підтвердити або відхилити обмін. У разі підтвердження обидві книги автоматично стають неактивними, а обом сторонам тимчасово відкривається доступ до контактних даних один одного (місто, пункт відправлення), що дає змогу організувати доставку. У разі відмови обмін не відбувається, книги залишаються доступними для інших запитів.

Все починається після того, як користувач натискає кнопку "Обміняти" на детальній сторінці книги, користувачу відкривається модальне вікно з переліком його власних книг, які вже були додані до обмінного фонду. На цьому вікні користувач може обрати ту книгу, яку він готовий запропонувати в обмін (рис. 2.18).

Інтерфейс модального вікна досить простий: кожна книга представлена обкладинкою, іменем автора та назвою. Таким чином, користувач має змогу зорієнтуватися серед своїх книг і швидко зробити вибір. Оскільки у користувача можуть бути додані кілька книг, буде реалізовано можливість прокрутки або розширення списку, якщо їх більше, ніж вміщується на екрані.

Після вибору, системою виводиться ще одне модальне вікно-підтвердження (рис. 2.19).

У ньому відображаються обидві книги: зліва та, яку користувач пропонує, справа та, яку він хоче отримати. Таке рішення дає змогу ще раз перевірити інформацію перед надсиланням запиту.

Користувачеві залишається лише натиснути кнопку підтвердження або скасувати дію. Запит фіксується в системі лише після підтвердження.

Після підтвердження запиту система автоматично створює сповіщення для власника книги, до якої наді. Це сповіщення з'являється у відповідному розділі особистого кабінету користувача. У ньому відображається, яка саме книга була запропонована в обмін, яку хочуть обміняти, хто надіслав запит, автори та назви книг, а також надається можливість прийняти або відхилити його (рис. 2.20).

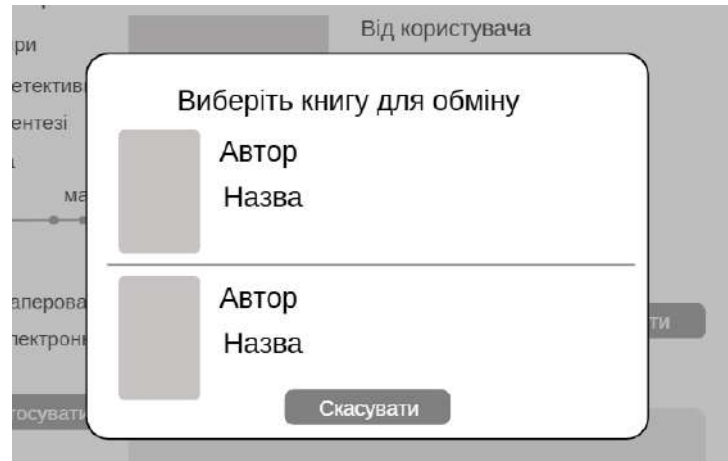


Рис. 2.18. Вибір книги на обмін

(розроблено автором)

У разі, якщо власник книги погоджується на обмін, обидві книги стають неактивними й зникають із загального каталогу.



Рис. 2.19. Підтвердження книги на обмін

(розроблено автором)

Якщо користувач відхиляє запит, система фіксує це рішення: книги залишаються активними, а в сповіщенні зазначено «Ви відхилили обмін». Відправник запиту у своєму кабінеті бачить повідомлення «Ваш запит на обмін було відхилено». У разі схвалення – «Ваш запит на обмін було прийнято».

Обидва учасники отримують сповіщення зі статусом «Ви підтвердили обмін» та бачать основну інформацію про іншу сторону: ім'я, номер телефону, місто й місце відправлення (рис. 2.21). Це дозволяє самостійно організувати доставку.



Рис. 2.20. Сповіщення про обмін

(розроблено автором)

Якщо рішення ще не прийнято, запит позначається як «очікує». Таким чином, система сповіщень дає обом сторонам бачити результат взаємодії, зберігаючи повну історію запитів.

Окрім обміну, на платформі буде реалізована також можливість прямої покупки книги з обмінного фонду. Якщо користувача цікавить книга, яка доступна до продажу, він зможе натиснути кнопку «Купити» на її детальній сторінці. На відміну від обміну, при покупці не потрібно обирати власну книгу у відповідь, замість цього одразу формується заявка на придбання.

Після підтвердження дії створюється новий запис у системі зі статусом «прийнято», типом «покупка», а також зазначеною вартістю книги. Так само, як і у випадку обміну, книга автоматично стає неактивною і зникає з публічного каталогу.



Рис. 2.21. Результат заявки

(розроблено автором)

Логіка покупки базується на тому ж механізмі обробки, що й обмін, однак передбачає лише односторонню дію без взаємного погодження. Завдяки цьому користувач зможе придбати бажану книгу, якщо вона виставлена як доступна для продажу

2.5 Проектування бази даних маркетплейса

Проектування функціональної частини веб-сайту, включаючи розробку макетів і визначення основних сценаріїв користувацької взаємодії неможливо реалізувати без чіткої структури бази даних. Тому що вона відповідає за збереження всієї інформації, яка проходить крізь інтерфейс: книги, користувачі, обміни, повідомлення, історія запитів тощо. У межах обмінного фонду база даних стає основою, яка забезпечує зв'язність між усіма діями користувачів і внутрішніми процесами системи.

На основі описаних раніше вимог і макетів було відокремлено декілька таблиць, які стосуються функціоналу маркетплейса в частині обміну книгами.

Першою з основних таблиць є users (рис. 2.22). У ній зберігається інформація про всіх зареєстрованих користувачів платформи, включаючи як звичайних відвідувачів, так і адміністраторів.

id [PK] integer	email character varying (255)	phone character varying (15)	password character varying (255)	role character varying (50)	registration_date timestamp without time zone	first_name character varying (100)	last_name character varying (100)	delivery_address text	city_id integer
1	user1@gmail.com	+380728219681	\$2b\$10\$gJ2n.x1eq0P1qYq5Fh2L...	user	2024-12-15 09:10:31.929387	Софін	Галицька	Ukrposhta 50019	29
2	admin1@gmail.com	+380266794537	\$2b\$10\$EeL4C0E172UjvMPrmM...	user	2024-12-15 14:52:11.823902	Антон	Каркоич	[null]	2
3	user2@gmail.com	+380646494560	\$2b\$10\$86J7VdVz3EUvAna42T9L...	user	2024-12-16 04:50:36.999461	Дарія	Нікорона	[null]	187
4	admin@gmail.com	+380427994782	\$2b\$10\$3tyMv/5.cAl.ugh.Nb6oj.#7...	admin	2024-12-16 23:25:23.320663	[null]	[null]	[null]	[null]
5	user12211@gmail.com	+380213132131	\$2b\$10\$e3kfc0q7NdotikVC.rbcO...	user	2025-04-08 21:58:41.260282	Тарасова	Ачрова	[null]	52
6	user31@gmail.com	+380746590472	\$2b\$10\$2WW2kUQn#TKMAEXIy...	user	2025-04-11 17:38:35.110448	Анна	Далицька	НоваПошта 20127	50
7	amihai127@gmail.com	+380946775658	\$2b\$10\$ptZ4ymnEX.LmZ0bpwE...	user	2025-04-12 00:38:51.691829	Ріта	Гушко	[null]	[null]

Рис. 2.22. Таблиця users

(розроблено автором)

Таблиця містить технічні та персональні дані: адресу електронної пошти, номер телефону, ім'я, прізвище, дату реєстрації, роль користувача (наприклад, admin або user), а також поле з паролем, який зберігається у зашифрованому вигляді.

Додатково передбачено можливість вказати місто проживання через зовнішній ключ city_id, а також місце відправлення або доставки у вигляді текстового поля. Ці дані згодом використовуються при організації доставки після погодження обміну. Роль користувача (role) визначає, який функціонал йому буде доступний: наприклад, користувач може додавати книги на обмін, переглядати пропозиції, надсилати заявки або підтверджувати запити інших осіб. Адміністратор має доступ до модерації книг та управління контентом.

Наступною таблицею є exchange_books (рис. 2.23). Вона відповідає за збереження інформації про книги, які користувачі виставляють на обмін або продаж. Кожен запис у таблиці пов'язаний із конкретним користувачем через зовнішній ключ user_id, що дозволяє ідентифікувати власника книги.

Таблиця містить основні текстові поля: назву title, автора (author), опис (description) і стан книги (condition). Зображення обкладинки зберігається як URL-шлях у полі image_url. Додатково передбачено два булеві поля: is_for_sale та is_for_exchange, які дозволяють вказати, в якому форматі доступна книга.

Поле `price` є обов'язковим лише для випадків продажу, а в інших випадках може залишатися порожнім.

17	11	Усі вірші. 1993...	Сергій Жадан	Новий	Вірші є сенс вид...	https://bookclub-c...	false	true	[null]	true	2025-04-09 ...	true
18	11	Моя війна	Валерій Залужний	Новий	Ця книжка – ос...	https://bookclub-c...	true	true	320.00	true	2025-04-09 ...	false
19	11	Поклик. Хронік...	Валерія В. Растет	Новий	«Поклик» – пер...	https://bookclub-c...	true	false	200.00	true	2025-04-09 ...	true
20	11	Балада про нед...	Стефані Гарбер	Новий	Не кожному кох...	https://bookclub-c...	false	true	[null]	true	2025-04-09 ...	true
21	16	Присяжник. Хр...	Брендон Сандерсон	Вживаний	Не сподобалося	https://bookclub-c...	false	true	[null]	true	2025-04-09 ...	true
22	16	Трохи ненавист...	Джо Аберкромбі	Новий	Не цікавить	https://bookclub-c...	false	true	[null]	true	2025-04-09 ...	true
23	16	Багряний рейд	Андрій Кокотюха	Новий	Максим Коломі...	https://bookclub-c...	false	true	[null]	true	2025-04-11 ...	true
24	16	Хороша дівчин...	Майкл Роботам	Вживаний	Іві Кормак · дівч...	https://bookclub-c...	true	true	230.00	true	2025-04-11 ...	true

Рис. 2.23. Таблиця `exchange_books`
(розроблено автором)

Для контролю публікації передбачено поле `approved`, яке позначає, чи була книга схвалена модератором, а також `active`, що визначає, чи книга наразі є доступною. Це дозволить деактивувати пропозиції, які вже були використані в обміні, без необхідності їх видалення з бази. Дата додавання фіксується автоматично в полі `created_at`.

Оскільки під час підтвердження обміну користувачам відкривається інформація про місто та пункт відправлення, важливим елементом бази даних є таблиця `cities` (рис. 2.24).

id [PK] integer	name character varying (100)
29	Березань
30	Березівка
31	Березне
32	Берестечко
33	Берестин
34	Берислав
35	Бершадь
36	Бібрка
37	Біла Церква

Рис. 2.24. Таблиця `city`
(розроблено автором)

У ній зберігається перелік населених пунктів, серед яких користувач обирає своє місце проживання під час реєстрації або редагування профілю.

Ще однією таблицею в бази даних є notifications (рис. 2.25). Вона використовується для зберігання всіх системних сповіщень, пов'язаних із обміном книгами між користувачами. Сповіщення дозволяють інформувати користувача про нові події: надходження запиту, підтвердження чи відмову, а також відкриття контактних даних для доставки.

id [PK] integer	user_id integer	message text	type text	related_exchange_id integer	is_read boolean	created_at timestamp with time zone	payload json
176	18	Користувач хоче обмінати свою книгу н...	exchange_request	43	true	2025-04-30...	[null]
177	17	Ваш запит на обмін очікує відповіді кор...	exchange_pending	43	true	2025-04-30...	[null]
178	17	Ваш запит на обмін було прийнято	exchange_result	43	true	2025-04-30...	[null]
179	18	Ви підтвердили обмін	exchange_result	43	true	2025-04-30...	[null]
180	19	Дані для відправки книги користувачу	delivery_info	43	false	2025-04-30...	{"name":"Анна Далицька","city":"Болград","address":"НоваПошта 20127"}
181	17	Дані для відправки книги користувачу	delivery_info	43	true	2025-04-30...	{"name":"Ріта Гушко","city":"Новий Роздін","address":"Укр.Пошта.205948"}

Рис. 2.25. Сповіщення

(розроблено автором)

Основний текст повідомлення зберігається у полі message. Тип сповіщення (type) вказує, до якого етапу обміну воно належить, наприклад, запит на обмін, результат обміну або повідомлення з контактними даними. Для того щоб визначити, до якого конкретного обміну належить повідомлення, використовується поле related_exchange_id. Час створення фіксується автоматично в полі created_at, а булеве поле is_read дозволяє відстежити, чи користувач уже переглянув повідомлення, що зручно при побудові логіки відображення у профілі.

У деяких випадках (наприклад, для передачі адреси доставки) додається додаткове поле payload, що зберігає структуру даних у форматі JSON. Це надає змогу передати змінний набір даних у межах одного сповіщення без зміни структури таблиці.

Останньою таблицею є exchanges (рис. 2.26). Вона фіксує всі підтверджені та очікуючі обміни або покупки між користувачами. Кожен запис у таблиці відображає взаємодію між двома користувачами щодо конкретної пари книг.

Таблиця містить ідентифікатори обох учасників: `requester_id` (користувач, який надіслав запит) і `owner_id` (власник книги). Через зовнішні ключі `requested_book_id` та `offered_book_id` встановлюється зв'язок із таблицею `exchange_books`, що дозволяє визначити, яка книга була запропонована, а яка отримана. Поле `status` показує поточний стан обміну: `pending`, `accepted` або `declined`. Якщо обмін має тип покупки, то в полі `amount` зазначається вартість книги. Тип взаємодії фіксується у полі `type`, де вказується `exchange` або `purchase`. Дата підтвердження або створення запису автоматично фіксується у полі `exchange_date`.

id [PK] integer	requester_id integer	owner_id integer	requested_book_id integer	offered_book_id integer	status text	amount numeric (10,2)	exchange_date timestamp without t	type text
35	17	18	41	[null]	pending	100.00	2025-04-16 20:1...	purchase
37	17	11	18	[null]	accepted	320.00	2025-04-17 01:3...	purchase
39	17	11	105	108	accepted	[null]	2025-04-17 22:1...	exchange
43	17	18	111	109	accepted	[null]	2025-04-30 23:0...	exchange

Рис. 2.26. Таблиця exchanges

(розроблено автором)

Описані вище таблиці формують основу обмінного функціоналу маркетплейса. Саме на їх основі буде реалізовано усі дії користувача: від додавання книги до каталогу обміну до оформлення та підтвердження обміну, обміну повідомленнями і завершення взаємодії.

Висновки до розділу 2

У цьому розділі було спроектовано структуру та логіку роботи веб-платформи, яка об'єднує класичний книжковий магазин і систему обміну книгами між користувачами. На основі аналізу вимог було визначено основні компоненти архітектури: клієнтську частину (React), серверну логіку (Node.js), базу даних (PostgreSQL) та хмарне сховище зображень. Такий підхід дозволив розділити функціональність за ролями та забезпечити масштабованість системи.

Спочатку було спроектовано архітектуру, яка включає клієнтську частину, серверну логіку та базу даних. Для кращого розуміння процесів було створено діаграми варіантів використання та діаграми діяльності, як для користувачів, так і для адміністратора.

В окремих підрозділах описано інтерфейс користувача, включаючи головну сторінку, каталог книг, модальні вікна та сторінку оформлення замовлення. Було також окремо виділено інтерфейс обмінного фонду: функцію, яка відрізняє книжковий маркетплейс від звичайного інтернет-магазину. Також описано процес додавання книг, проходження модерації, перегляду обмінних книг і детальної сторінки, на якій користувач може запропонувати обмін або здійснити покупку. Крім того, було спроектовано логіку обміну між користувачами: від вибору книги до підтвердження чи відмови, а також реалізовано сповіщення, які дозволяють обом сторонам отримати всю необхідну інформацію.

У кінці, описано таблиці, які зберігають інформацію про користувачів, книги, заявки на обмін, покупки і сповіщення. Завдяки цьому можна простежити весь шлях дій користувача від додавання книги до завершення взаємодії.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЄКТУ

3.1 Реалізація серверної частини та налаштування бази даних

Щоб забезпечити повноцінне функціонування веб-застосунку, необхідно реалізувати серверну частину, яка відповідатиме за збереження, обробку та передачу даних між клієнтом і базою даних. Сервер виступає головною частиною, що приймає запити користувачів, виконує необхідні дії (наприклад, додавання книги, створення замовлення, тощо) і повертає відповідь у зручному форматі, зазвичай у вигляді JSON.

Розробка бекенду була здійснена з використанням Node.js та фреймворку Express. Для зберігання даних використовувалась система керування базами даних PostgreSQL, що дозволяє працювати з реляційною структурою даних. Комунікація із базою відбувається за допомогою бібліотеки pg.

Усі конфіденційні параметри, такі як ключі доступу до бази даних або Amazon S3, зберігаються безпосередньо в файлі .env, в якому визначені змінні середовища (рис. 3.1).

```
JWT_SECRET=super_secret_key_123
DATABASE_URL=postgresql://bookclub_ugcq_user:h9rLI6kpRgZrc5nziFiLBU78PL6deMar@dpg-d6m73o1r0fns73cc1qh0-a-frankfurt-postgres.render.com/bookclub_ugcq

AWS_ACCESS_KEY_ID=AKIAWDVSVNCFQ6RM5D
AWS_SECRET_ACCESS_KEY=84NKRkFUcScUCUc8KmN62AWC//D013/g2j0J63Q7
AWS_REGION=eu-north-1
AWS_BUCKET_NAME=bookclub-covers
AWS_USER_BUCKET_NAME=bookclub-user
FRONTEND_URL=https://tubular-halva-551063.netlify.app
```

Рис. 3.1. файл env

(розроблено автором)

Файл db.js (рис. 3.2) відповідає за встановлення з'єднання з базою даних PostgreSQL. Для цього використовуються параметри, винесені у файл

конфігурації `.env`, що дозволяє централізовано керувати налаштуваннями середовища без необхідності змінювати код програми.

```
const { Pool } = require('pg');
require('dotenv').config();

require('dotenv').config();

const pool = new Pool({
  connectionString: process.env.DATABASE_URL,
  ssl: process.env.NODE_ENV === 'production' ? { rejectUnauthorized: false } : false
});

pool.connect().then(() => console.log('Без проблем підключено БД')).catch(err => console.error('Помилка підключення до БД:', err));
```

Рис. 3.2. Файл для підключення до бази даних

(розроблено автором)

У проєкті використовується хмарне сховище Amazon S3 для зберігання обкладинок книг, фрагментів текстів (уривків) та аватарів користувачів. Такий підхід дозволяє значно зменшити навантаження на сервер і базу даних, а також забезпечити швидкий доступ до зображень з будь-якого пристрою.

Для підключення до S3 використано (Додаток А.1) бібліотеку `@aws-sdk/client-s3`. На початку створюється клієнт S3 на основі даних із `.env`.

Для завантаження зображень або уривків книг реалізовано функцію `uploadUserFileToS3`, яка записує файл у відповідну папку (`covers/`, `pending/`, `approved/`, `users/`). Такий поділ спрощує модерацію та подальше керування файлами. Передбачено також функції для переміщення файлів у межах бакету (наприклад, після схвалення адміністратором) і видалення у разі відмови чи очищення каталогу.

Центральною частиною серверної логіки є файл `server.js`, він відповідає за ініціалізацію Express-застосунку, підключення необхідних `middleware`, реєстрацію маршрутів та запуск HTTP-сервера. Його структура дозволяє розділити відповідальність між модулями. На початку цього файлу відбувається підключення конфігурації та основних залежностей (рис 3.3).

```
require('dotenv').config();
const express : e | () => core.Express = require('express');
const cors : function(any): function(any, a... | {...} = require('cors');
const pool : {} = require('./db');
```

Рис. 3.3. Приклад конфігурацій

(розроблено автором)

Потім створюється сам застосунок і додаються базові middleware, наприклад, для того, щоб сервер міг приймати дані у форматі JSON і не блокував запити з клієнтської частини, яка працює на іншому порту (рис. 3.4).

```
const app = express();
app.use(cors({
  origin: FRONTEND_URL,
  methods: ['GET', 'POST', 'PUT', 'DELETE'],
  allowedHeaders: ['Content-Type', 'Authorization'],
  credentials: true
}));
```

Рис. 3.4. Налаштування CORS-захисту та обробки JSON-запитів

(розроблено автором)

Цей блок дозволяє серверу приймати JSON-запити з клієнтської частини та налаштовує CORS-захист для конкретного джерела. Це важливо для веб-застосунку, де клієнт і сервер працюють на різних портах. Після цього до застосунку підключаються всі основні маршрути, де кожен з яких реалізує логіку певного функціонального модуля (рис. 3.5).

Кожен маршрут є окремим модулем, який підключається до Express-застосунку. Наприклад, /api/auth відповідає за реєстрацію та вхід, /api/exchange – за логіку обміну книгами, а /api/books – за роботу з каталогом. Для кожного напрямку, чи то автентифікації, замовлень, обміну чи профілю, створено окремі

маршрути та контролери, що відповідає принципам модульності і спрощує розширення функціоналу.

```
app.use('/api/auth', authRoutes);
app.use('/api/orders', orderRoutes);
app.use('/api/books', booksRouter);
app.use('/static', express.static(path.join(__dirname, 'static')));
app.use('/api/profile', profileRoutes);
app.use('/api/exchange', exchangeRoutes);
app.use('/api/comments', require('./routes/comments'));
app.use('/api/notifications', require('./routes/notificationsRouter'));
app.use('/api/notifications', notificationRoutes);
app.use('/api/users', userRoutes);
```

Рис. 3.5. Маршрути файлу server

(розроблено автором)

Для захищених маршрутів, таких як profile, orders, exchange чи notifications, використовується middleware authenticateUser (рис. 3.6). Вона перевіряє наявність та дійсність токена авторизації, надаючи доступ лише авторизованим користувачам. Інакше запит блокується, що дозволяє захищати особисті дані користувачів та забезпечити контроль доступу.

```
const authenticateUser = (req, res, next) :any | undefined => { Show usages ⚡ Shego *
  const token = req.headers['authorization'];

  if (!token || !token.startsWith('Bearer ')) {
    return res.status(403).json({ message: 'Немає доступу або неправильний формат токена' });
  }

  const tokenWithoutBearer = token.split(' ')[1];
  try {
    const decoded :any = jwt.verify(tokenWithoutBearer, JWT_SECRET);
    req.user = decoded;
    next();
  } catch (err) {
    return res.status(401).json({ message: 'Невірний токен' });
  }
};
```

Рис. 3.6. Файл middleware

(розроблено автором)

У результаті серверна частина проекту забезпечує роботу з базою даних і API, й підтримує авторизацію, перевірку прав доступу, інтеграцію з Amazon S3. Також структуру, яка дозволяє легко розширювати функціональність.

3.2 Реалізація функціональності маркетплейса

Після реалізації базової серверної інфраструктури було впроваджено повноцінну бізнес-логіку маркетплейса, яка дозволяє користувачам не лише переглядати книги, а й активно взаємодіяти: додавати оголошення, надсилати запити на обмін, здійснювати покупки, отримувати сповіщення та переглядати історію дій.

Серед основних принципів, на яких побудовано систему:

- розмежування ролей між адміністраторами та звичайними користувачами;
- фіксація всіх взаємодій (обміни, покупки, запити);
- асинхронна комунікація через систему сповіщень;
- модерація вмісту для контролю якості;
- підтримка різних форматів участі: продаж, обмін або обидва.

Кожен зареєстрований користувач може додати книгу до обмінного фонду або виставити її на продаж. У формі вказуються назва, автор, стан примірника, опис, ціна (якщо передбачено продаж), тип пропозиції (обмін/продаж/обидва), а також додається обкладинка, яка одразу завантажується у хмарне сховище. Після додавання дані надсилаються на сервер і зберігаються в таблиці `exchange_books` (рис. 3.7). Нові записи автоматично отримують статус `approved = false`, тобто стають недоступними до проходження модерації.

Після збереження книга потрапляє до списку заявок на модерацію. На стороні адміністратора реалізовано окремий інтерфейс, де відображаються усі книги зі статусом "не схвалено". Адміністратор переглядає основні параметри книги і приймає рішення щодо її публікації. Якщо заявка відхиляється, книга видаляється з бази разом із супровідними файлами.

У разі схвалення статус змінюється на `approved = true`, і книга стає доступною у відкритому каталозі. Користувачі можуть або купити її, або обмінятися, залежно від того, які варіанти були вказані власником під час додавання (рис. 3.8).

```
const result = await pool.query(`
  INSERT INTO exchange_books
  (user_id, title, author, condition, description, image_url, is_for_sale, is_for_exchange, price)
  VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9)
  RETURNING *
`, [userId, title, author, condition, description, image_url, is_for_sale, is_for_exchange, price]);
```

Рис. 3.7. Додавання книги до бази

(розроблено автором)

При купівлі користувач додає книгу до кошика. Але якщо обирається обмін, то процес інший. Користувач повинен обрати одну зі своїх книг, яка виставлена на обмін, після чого система створює новий запис у таблиці `exchanges`. Такий запис містить ID обох користувачів, ID обох книг (якщо це обмін), або суму (якщо це покупка), тип дії (`exchange` або `purchase`) і початковий статус (`pending` чи `accepted`).

```
const result = await pool.query(`
  INSERT INTO exchanges
  (requester_id, owner_id, requested_book_id, offered_book_id, amount, type, status)
  VALUES ($1, $2, $3, $4, $5, $6, $7)
  RETURNING *
`, [
  requesterId,
  ownerId,
  requested_book_id,
  offered_book_id || null,
  amount || null,
  type || 'exchange',
  type === 'purchase' ? 'accepted' : 'pending'
]);
```

Рис 3.8. Створення нового запису обміну або покупки

(розроблено автором)

У разі купівлі система одразу деактивує книгу. У випадку обміну надсилаються сповіщення обом користувачам, і статус залишається "очікує підтвердження", доки власник не відреагує.

Для зручного обміну книгами, система інформує користувачів про всі етапи: від створення запиту до його підтвердження чи відмови. Коли потрібно сформувати список актуальних сповіщень, бекенд виконує запит, який враховує роль користувача в обміні, статус обміну та тип повідомлення (рис. 3.9). Також приєднується інформація про книги і імена учасників, щоб повідомлення були змістовними.

```

exports.getUserNotifications = async (req, res) => {
  const userId = req.user.id;

  try {
    const result = await pool.query(`
      SELECT
        n.id, n.type, n.message, n.is_read, n.payload,
        e.status AS exchange_status,
        CASE WHEN n.user_id = e.requester_id THEN ob.title ELSE rb.title END AS your_book_title,
        CASE WHEN n.user_id = e.requester_id THEN rb.title ELSE ob.title END AS other_book_title
      FROM notifications n
      LEFT JOIN exchanges e ON n.related_exchange_id = e.id
      LEFT JOIN exchange_books rb ON e.requested_book_id = rb.id
      LEFT JOIN exchange_books ob ON e.offered_book_id = ob.id
      WHERE n.user_id = $1 AND e.status IS NOT NULL
      ORDER BY n.created_at DESC
    `, [userId]);

    res.json(result.rows);
  } catch (err) {
    console.error('Помилка при отриманні сповіщень:', err);
    res.status(500).json({ message: 'Помилка сервера' });
  }
};

```

Рис. 3.9. Отримання сповіщень для користувача

(розроблено автором)

Щоб користувач міг легко згадати, які дії він вже виконав – реалізована сторінка історії (рис. 3.10). У ній відображаються завершені взаємодії (status = 'accepted'), і до кожного запису додаються дані про обидві книги та обох учасників. На бекенді це реалізовано запитом у контролері, який автоматично визначає, хто яку роль грав у кожному конкретному обміні.

У результаті користувач бачить лише ті записи, які справді були завершені, із змогою побачити: хто з ким обмінювався, які книги брали участь, і коли саме це відбулося.

Після завершення реалізації функціональності клієнтської та серверної частин постало завдання опублікувати веб-застосунок у мережі з дотриманням вимог безпеки. Проект розгорнений на двох окремих сервісах: Netlify для клієнтської частини та Render для серверної.

```

exports.getUserExchangeHistory = async (req, res) : Promise<void> => {
  const userId = req.user.id;

  try {
    const result = await pool.query(`
      SELECT
        e.id, e.type, e.status, e.exchange_date,
        rb.title AS requested_book_title,
        ob.title AS offered_book_title,
        ru.username AS requester_username,
        ou.username AS owner_username
      FROM exchanges e
      LEFT JOIN exchange_books rb ON e.requested_book_id = rb.id
      LEFT JOIN exchange_books ob ON e.offered_book_id = ob.id
      LEFT JOIN users ru ON e.requester_id = ru.id
      LEFT JOIN users ou ON e.owner_id = ou.id
      WHERE (e.requester_id = $1 OR e.owner_id = $1)
      AND e.status = 'accepted'
      ORDER BY e.exchange_date DESC NULLS LAST
    `, [userId]);

    res.json(result.rows);
  } catch (err) {
    console.error('Помилка при отриманні історії обмінів:', err);
    res.status(500).json({ message: 'Помилка сервера' });
  }
}

```

Рис. 3.10. Отримання історії

(розроблено автором)

Netlify забезпечує підключення до GitHub-репозиторію та автоматичне оновлення сайту після кожного коміту. Також, сервіс автоматично генерує SSL-сертифікат, що гарантує захищене HTTPS-з'єднання без необхідності ручного налаштування.

Оскільки клієнтська частина сайту реалізована у форматі SPA, виникає потреба у правильному маршрутизуванні навіть при прямому переході на вкладені маршрути (наприклад, /book/15 чи /profile). За замовчуванням сервер (у даному випадку Netlify) може не знати, як обробити такий запит і повернути помилку 404. Щоб уникнути цього, було додано спеціальний файл `_redirects` з правилом, яке змушує сервер перенаправляти усі запити до `index.html`, де React уже самостійно визначає, який маршрут слід показати. Таким чином, навіть якщо користувач оновлює сторінку або переходить за глибоким посиланням, застосунок працює правильно.

Render було обрано для серверної частини через підтримку Node.js, просте налаштування змінних середовища та інтеграцію з базою даних PostgreSQL. Зокрема, у файл `.env` на Render додано змінну `FRONTEND_URL`, яка містить адресу клієнтської частини на Netlify. Це дозволило налаштувати CORS-захист на сервері, обмежуючи запити лише з дозволеного джерела.

Крім цього, Render також автоматично шифрує передані дані, підтримує HTTPS і забезпечує контроль доступу до змінних середовища, які зберігаються окремо від вихідного коду. Завдяки цьому ключі доступу до бази даних, Amazon S3 та JWT-секрети не зберігаються у відкритому вигляді й недоступні стороннім особам.

3.3 Програмна реалізація інтерфейсу користувача

Інтерфейс користувача виконує роль посередника між системою та її відвідувачем. Він забезпечує доступ до функцій веб-застосунку, дозволяє переглядати, вводити та змінювати дані, а також формує загальне враження про роботу сервісу. У контексті електронного маркетплейса з можливістю обміну книгами інтерфейс повинен поєднувати зручність, логіку подання інформації та ефективну навігацію. В іншому випадку користувач не зможе повноцінно скористатися функціоналом, навіть якщо технічна частина працює без помилок.

Проектування інтерфейсу здійснювалося з урахуванням потреб різних груп користувачів: нових відвідувачів, зареєстрованих учасників, які виставляють або шукають книги, а також адміністраторів, які здійснюють перевірку контенту. Весь інтерфейс було реалізовано з використанням бібліотеки React, що надала змогу структурувати застосунок у вигляді окремих компонентів.

Однією з основних функцій веб-інтерфейсу є можливість користувача самостійно виставити книгу на продаж або обмін. Для цього реалізовано окрему сторінку з формою. Під час введення назви та автора працює механізм підказок, що працює на запиті до списку вже доданих книг. Це дозволяє зменшити кількість дублікатів і покращує уніфікацію даних. Передбачено можливість додавання опису, зазначення стану книги, завантаження зображення, а також вибору: продаж і/або обмін. Частина форми подана на рис. 3.11.

```
<div className="form-field">
  <label>Назва:</label>
  <div className="suggestion-wrapper" ref={titleRef}>
    <input
      type="text"
      name="title"
      value={formData.title}
      onChange={handleChange}
      onFocus={() : void => setShowTitleSuggestions( value: true)}
      autoComplete="off"
      placeholder="Почніть вводити назву..."
    />
    {showTitleSuggestions && filteredTitles.length > 0 && (
      <ul className="suggestions">
        {filteredTitles.map((t, i : number ) => (
          <li key={i} onClick={() : void => handleTitleSelect(t)}>{t}</li>
        ))}
      </ul>
    )}
  </div>
</div>
```

Рис. 3.11. Форма додавання книги

(розроблено автором)

Наступна функція для користувачів – це можливість обмінюватися книгами між собою. Після переходу на детальну сторінку книги користувач може обрати одну з доступних дій: купити або обміняти. Якщо вибрано обмін, відкривається модальне вікно, де відображається список його власних книг, доступних для обміну. Після вибору конкретної книги відкривається друге модальне вікно з підтвердженням наміру обміну. У вікні відображаються обидві книги, і користувачеві пропонується підтвердити або скасувати дію. Приклад компонента, що відповідає за цей етап, подано на рис. 3.12.

```

<h3>Підтвердження обміну</h3>

<div className="exchange-preview">
  <div className="book-preview">
    <div className="book-owner-label">Ваша книга</div>
    <img src={offeredBook.image_url} alt={offeredBook.title} />
    <div className="book-text">
      <p className="book-title">{offeredBook.title}</p>
      <p className="book-author">{offeredBook.author}</p>
    </div>
  </div>
</div>

<div className="exchange-arrow-wrapper">
  <span className="exchange-arrow">↔</span>
</div>

<div className="book-preview">
  <div className="book-owner-label">
    {requestedBook.first_name} {requestedBook.last_name}
  </div>
  <img src={requestedBook.image_url} alt={requestedBook.title} />
  <div className="book-text">
    <p className="book-title">{requestedBook.title}</p>
    <p className="book-author">{requestedBook.author}</p>
  </div>
</div>

<p className="exchange-question">
  Ви хочете запропонувати свою книгу в обмін на цю?
</p>

<div className="confirm-buttons">
  <button onClick={onConfirm} className="btn primary">Підтвердити</button>
  <button onClick={onCancel} className="btn secondary">Скасувати</button>
</div>

```

Рис. 3.12. Відкриття модального вікна

(розроблено автором)

Процес вибору книги для обміну реалізований у два етапи: спершу користувач бачить перелік своїх книг, доступних для обміну, після чого обирає одну з них. Після вибору відкривається друге модальне вікно (рис. 3.13), де відображається і його книга, і книга іншого користувача.

Для забезпечення безпечного контенту на платформі передбачено модерацію усіх нових оголошень. Після того як користувач додає книгу, система автоматично позначає її як "очікує підтвердження", а зображення тимчасово зберігається в папці pending/. Адміністратор платформи має доступ до спеціальної сторінки модерації, де йому відображаються всі книги, що потребують перевірки.

```

<div className="exchange-preview">
  <div className="book-preview">
    <div className="book-owner-label">Ваша книга</div>
    <img src={offeredBook.image_url} alt={offeredBook.title} />
    <div className="book-text">
      <p className="book-title">{offeredBook.title}</p>
      <p className="book-author">{offeredBook.author}</p>
    </div>
  </div>

  <div className="exchange-arrow-wrapper">
    <span className="exchange-arrow">↔</span>
  </div>

  <div className="book-preview">
    <div className="book-owner-label">
      {requestedBook.first_name} {requestedBook.last_name}
    </div>
    <img src={requestedBook.image_url} alt={requestedBook.title} />
    <div className="book-text">
      <p className="book-title">{requestedBook.title}</p>
      <p className="book-author">{requestedBook.author}</p>
    </div>
  </div>
</div>

```

Рис. 3.13. Порівняння книг у модальному вікні обміну

(розроблено автором)

Інтерфейс цієї сторінки побудований у вигляді сітки карток, кожна з яких містить мініатюру зображення, назву книги, автора та ID оголошення. Щоб переглянути детальну інформацію, достатньо натиснути на картку і відкриється модальне вікно (рис. 3.14) з описом, типом розміщення (продаж/обмін), станом книги, а також кнопками для підтвердження або відмови.

Останнім важливим елементом у взаємодії з користувачем є система сповіщень. Вона забезпечує зворотний зв'язок і дозволяє інформувати про дії інших учасників: надсилання або підтвердження запиту на обмін, отримання контактних даних, статус взаємодії, тощо.

```

{selectedBook && (
  <Modal onClose={() :void => setSelectedBook( value: null)}>
    <div style={{ display: 'flex', gap: '12px' }}>
      <img src={selectedBook.image_url}
        alt={selectedBook.title}
        style={{ width: 100, height: 140, objectFit: 'cover' }}/>
      <div>
        <p><strong>{selectedBook.author}</strong></p>
        <p>{selectedBook.title}</p>
        <p>Обмін: {selectedBook.is_for_exchange ? '✓' : '✗'}</p>
        <p>Продаж: {selectedBook.is_for_sale ? `(${selectedBook.price} грн) ✓` : '✗'}</p>
        <p>Стан: {selectedBook.condition}</p>
      </div>
    </div>
    <div className="modal-description">
      <p><strong>Опис:</strong></p>
      {selectedBook.description.split('\n').map((line, index) => (
        <p key={index}>{line}</p>
      ))}
    </div>
    <div className="modal-buttons">
      <button className="approve"
        onClick={() : Promise<void> =>
          handleApprove(selectedBook.id)}>Прийняти</button>
      <button className="reject"
        onClick={() : Promise<void> =>
          handleReject(selectedBook.id)}>Відмовити</button>
    </div>
  </Modal>
)}

```

Рис. 3.14. Вміст модального вікна для модерації книги

(розроблено автором)

Сторінка сповіщень реалізована у вигляді списку карток, де кожне повідомлення має тип (наприклад, «Запит на обмін» або «Дані для доставки»), короткий зміст і візуальне подання книг, що беруть участь у взаємодії. Завдяки цьому користувач відразу бачить, які саме книги стосуються конкретного повідомлення, хто є другим учасником, і які дії від нього очікуються. На рис. 3.15 зображено приклад коду, що відповідає за вивід інформації про сповіщення.

У разі надходження нового запиту на обмін користувач отримує відповідне сповіщення, в якому відображено обидві книги: його власну та ту, яку запропонувала інша сторона. Кожна книга супроводжується зображенням, назвою та автором, що дає змогу швидко оцінити обмін. У межах одного повідомлення також відображаються кнопки дій – «Прийняти» або «Відхилити» запит. Натискання однієї з них змінює статус у базі даних, а сама кнопка після цього деактивується, запобігаючи повторним діям. Крім того, система автоматично оновлює відображення статусу для обох сторін, унеможливаючи виникнення конфліктних дій.

```
<div className="book-preview">
  <img src={notif.your_book_image} alt={notif.your_book_title} />
  <p>Ваша книга</p>
  <p><strong>{notif.your_book_title}</strong></p>
  <p>{notif.your_book_author}</p>
</div>

<span className="exchange-arrow">↔</span>

<div className="book-preview">
  <img src={notif.other_book_image} alt={notif.other_book_title} />
  <p>{notif.other_first_name} {notif.other_last_name}</p>
  <p><strong>{notif.other_book_title}</strong></p>
  <p>{notif.other_book_author}</p>
</div>
```

Рис. 3.15. Відображення інформації про обмін у сповіщенні

(розроблено автором)

Також, після підтвердження обміну обом учасникам надходить сповіщення (рис. 3.16) з інформацією про ім'я, місто та місце доставки іншої сторони. Цей тип позначається як `delivery_info` і відображається у вигляді окремого блоку без зображень книг, але з акцентом на даних користувача.

```
<div className="delivery-info-block">
  <p><strong>Користувач:</strong> {notif.payload.name}</p>
  <p><strong>Місто:</strong> {notif.payload.city}</p>
  <p><strong>Місце доставки:</strong> {notif.payload.address}</p>
</div>
```

Рис. 3.16. Вивід контактної інформації для доставки

(розроблено автором)

Архітектура інтерфейсу формувалася на основі типової послідовності дій користувача, що охоплює створення оголошення, участь в обміні, перегляд сповіщень, тощо. Усі основні сценарії – створення оголошення, перегляд, підтвердження дії, модерація та повідомлення – реалізовані як окремі компоненти, які легко підтримувати й редагувати.

3.4 Тестування веб-сайту

Тестування було спрямоване на перевірку коректної роботи маркетплейса в умовах реальної взаємодії з користувачами. Мета тестування – переконатися, що всі дії, такі як додавання книг, оформлення обмінів або покупок, модерація контенту й система сповіщень, виконуються та без помилок.

Додавання книги користувачем. Цей функціонал є основою всієї логіки обміну, адже без можливості завантажити свою книгу жоден подальший процес не може бути ініційований. Важливо було перевірити не лише заповнення форми, а й правильне завантаження зображення, створення запису у базі та переведення книги до черги на модерацію.

У процесі тестування вводилися як повністю валідні дані, так і приклади з порожніми полями або некоректними форматами (рис. Б.1). Система правильно

обробляла обидва варіанти: у разі помилки виводилось відповідне повідомлення, а у разі успішного введення даних книга надсилалась на модерацію.

Модерація книг. Після додавання книга відображається у списку заявок на модерацію. Після додавання книга відображається у списку заявок на модерацію (рис. В.1) у вигляді окремих карток з ID, обкладинкою, назвою та автором відображаються всі книги, що очікують перевірки.

При натисканні на картку відкривається модальне вікно (рис. Г.1) з повною інформацією про книгу: назва, автор, стан, опис, варіант участі (обмін, продаж або обидва). У нижній частині містяться кнопки «Прийняти» та «Відмовити», що дозволяють адміністратору затвердити або відхилити заявку.

Було протестовано обидва варіанти рішення:

- при натисканні «Прийняти» книга змінює статус на `approved = true`, зображення переноситься з папки `pending/` до `approved/` у хмарному сховищі, і книга автоматично з'являється на публічній сторінці обміну (праворуч на рисунку);
- при натисканні «Відмовити» – запис видаляється з бази.

Купівля книги. Якщо книга, додана до обмінного фонду, також має встановлену ціну, користувач отримує можливість придбати без необхідності обміну. Як показано на рисунку 3.17, у кошику відображається обрана книга з інформацією про автора, тип (обмінна чи звичайна), ціна та вартість замовлення.

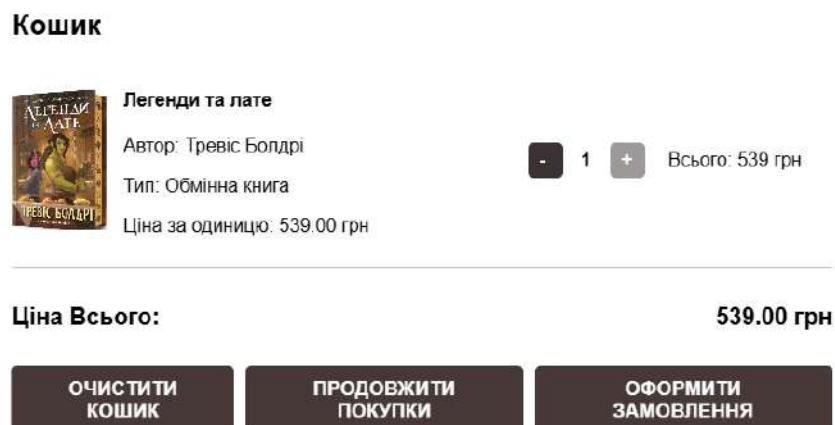


Рис. 3.17. Тестування купівлі книги з обмінного фонду

(розроблено автором)

Було протестовано сценарії додавання кількох книг до кошика, зміни кількості, а також повного очищення кошика. Система коректно обробляє ці дії, своєчасно оновлює вартість і не дозволяє оформлення замовлення за відсутності позицій. Таким чином, механізм купівлі обмінної книги працює аналогічно до звичайного товарного каталогу.

Обмін книгами. У випадку, якщо користувач не хоче купувати книгу, а натомість бажає здійснити обмін, він може скористатися функцією «Обмінити». Після натискання відкривається модальне вікно з вибором книги зі свого власного списку (рис. 3.18). Тут відображаються всі активні, схвалені книги, які користувач раніше виставив на обмін.

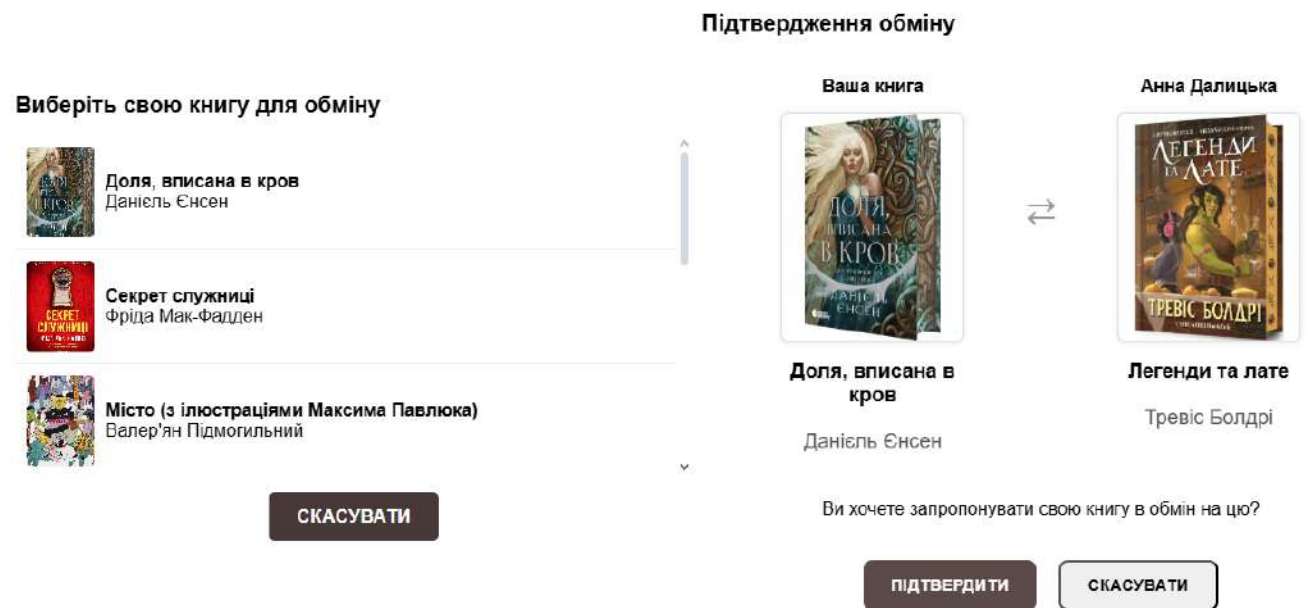


Рис. 3.18. Ініціювання обміну

(розроблено автором)

Після вибору книги система переходить до підтвердження обміну. У вікні відображаються обидві книги: запропонована та бажана, із зазначенням назви, автора й імені власника. Натискання кнопки «Підтвердити» створює запит на обмін і автоматично надсилає сповіщення обом учасникам.

Тестування підтвердило, що система правильно визначає роль користувача, створює запис у базі, фіксує статус і не дозволяє обмінювати власну книгу на свою ж. У разі помилки виводиться відповідне повідомлення, дублікати запитів не формуються.

Система сповіщень інформує обох учасників про ініційований обмін. У першому випадку ініціатор бачить статус «Очікує відповіді», що означає, що додаткових дій не потрібно. В іншому, власник книги отримує можливість підтвердити або відхилити запит (рис. 3.19).



Рис 3.19. Підтвердження запиту

(розроблено автором)

Обидва типи сповіщень мають уніфікований вигляд: обкладинки книг, назви, автори, імена користувачів. Це дає змогу зрозуміти, яка дія відбулася, хто її ініціював, і яку книгу пропонується натомість.

Після натискання кнопки «Підтвердити» у сповіщенні, система змінює статус відповідного запису обміну на «прийнято». Обидві книги, що брали участь в обміні, автоматично деактивуються, вони більше не відображаються в загальному каталозі, не можуть бути обрані для нового обміну або купівлі. Окрім

цього, кожен із учасників отримує фінальне сповіщення із зазначенням результату та детальною інформацією про відправлення.

Кожен користувач бачить блок (рис. 3.20) «Дані для доставки книги», в якому зазначено ім'я іншого учасника, місто, а також місце отримання (відділення пошти). Це дозволяє виконати фізичний обмін без додаткового листування.

Нижче відображено підтверджений обмін: зліва книга користувача, а справа – книга, яку він отримає.

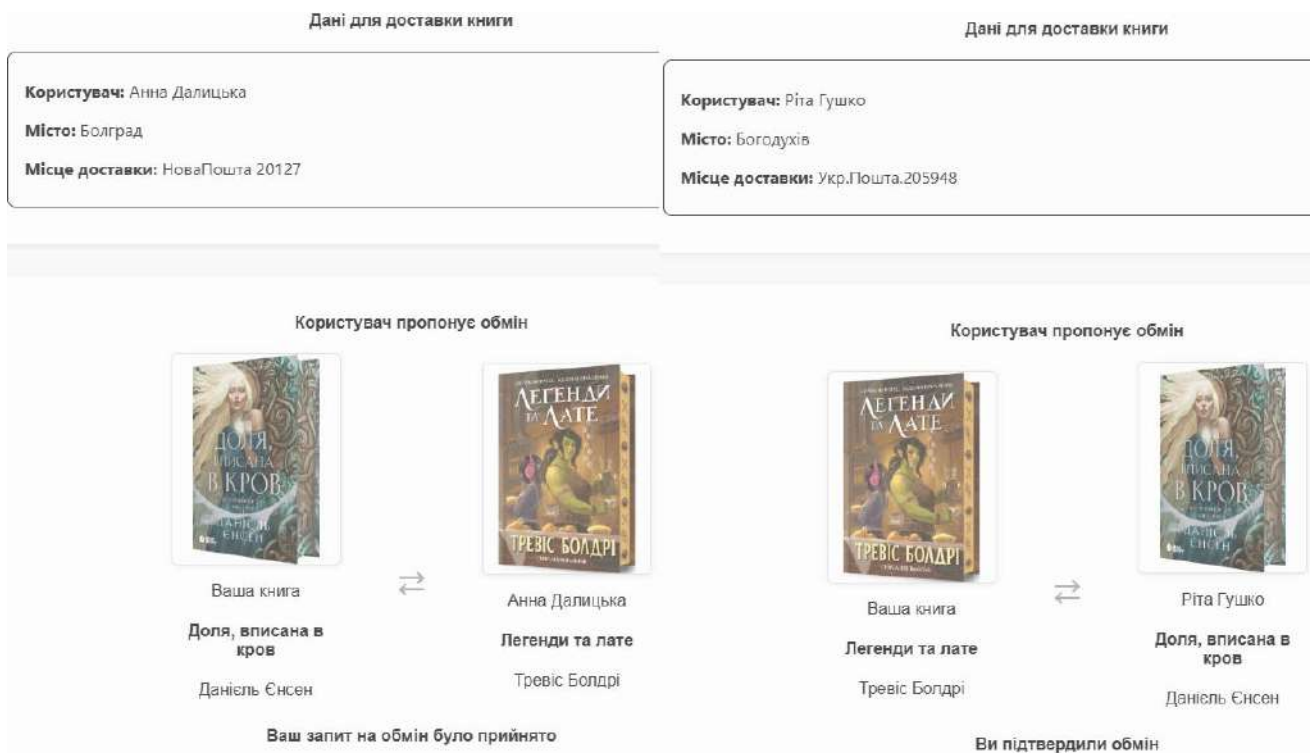


Рис 3.20. Підтвердження запиту

(розроблено автором)

У разі відхилення запиту на обмін система фіксує цю подію у вигляді сповіщення, не змінюючи стану книг у базі даних. Обидві книги залишаються активними й доступними для подальших запитів. Такий підхід дозволяє підтримувати процес обміну без додаткових дій з боку користувачів.

На рисунку 3.21 показано, як виглядає повідомлення про відхилення. Користувач, що скасував обмін, бачить відповідний напис «Ви відхилили обмін», а ініціатор запиту – «Ваш запит на обмін було відхилено».

У кожному випадку відображається повна інформація про книги, що брали участь у спробі обміну. Це дозволяє зберігати контекст взаємодії та уникати непорозумінь. Таким чином, жодні дані не втрачаються, а обидві сторони отримують повідомлення про результат



Рис. 3.21. Повідомлення після відхилення запиту на обмін
(розроблено автором)

На рисунку 3.22 зображено, як відображаються записи у цьому розділі. У випадку з обміном система чітко демонструє, яка книга була надана користувачем, а яку він отримав натомість.

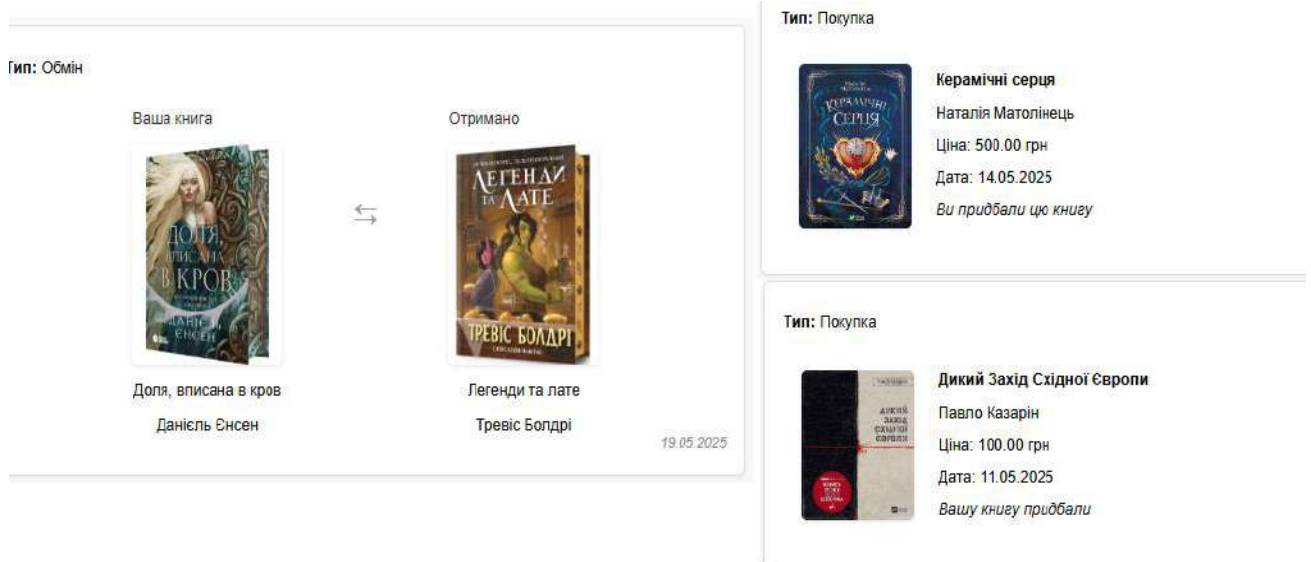


Рис. 3.22. Історія обміну

(розроблено автором)

Що ж до покупок, то інтерфейс окремо показує книги, які користувач придбав або продав, із зазначенням вартості, дати та відповідним текстом, який пояснює, яку саме дію він виконав.

На цьому етапі було підтверджено, що система правильно фіксує завершені взаємодії, точно визначає роль кожного учасника та забезпечує достовірність усіх даних.

Після усунення виявлених неточностей та уточнення логіки взаємодії модулів веб-сайт успішно пройшов функціональне тестування. Усі компоненти: додавання книг, модерація, сповіщення, обміни, покупки та історія дій – працюють узгоджено. Користувацькі дії обробляються правильно на клієнтській і серверній стороні. Функціональність відповідає вимогам проекту, а система готова до повноцінного використання.

Висновки до розділу 3

У результаті реалізації веб-застосунку вдалося створити повноцінну платформу для продажу та обміну книг, що поєднує інтерфейс користувача, надійну серверну логіку та гнучку базу даних. Під час роботи над розділом було

враховано аспекти функціонування сучасного книжкового маркетплейса, а саме: від створення оголошення до фінального підтвердження обміну з відображенням історії.

Розроблений інтерфейс передбачає інтуїтивну навігацію для різних ролей користувачів: покупців, продавців, учасників обміну та адміністраторів. Реалізовано динамічне оновлення даних без перезавантаження сторінки, модальні вікна для підтвердження дій, а також окремі сторінки профілю та історії. Всі компоненти працюють як єдине ціле і користувач може легко додати книгу, переглянути доступні варіанти, оформити покупку або ініціювати обмін.

На рівні серверної частини розроблено архітектуру, що базується на Node.js, Express та PostgreSQL. Впроваджено захист доступу через JWT, підключено хмарне сховище AWS S3 для зображень, а також забезпечено розділення логіки між контролерами та маршрутами. Всі маршрути перевіряються на автентифікацію, що гарантує безпеку доступу до особистих даних.

Було приділено увагу логіці обміну книгами: реалізовано механізм створення запиту, надсилання сповіщень, підтвердження або відхилення обміну, деактивації книг та фіксації результату. Завдяки системі сповіщень кожен користувач бачить повну історію своїх дій, а також може отримати необхідну інформацію для доставки книги іншому учаснику. Адміністративна частина дозволяє модерувати контент і слідкувати за якістю оголошень.

Після завершення реалізації було проведено. Усі компоненти, як клієнтські, так і серверні працюють узгоджено. Реалізована система не лише відповідає поставленим вимогам, але й дозволяє масштабування в майбутньому.

Крім того, визначено перспективи оптимізації веб-застосунку, зокрема впровадження кешування, використання CDN для зображень і поліпшення мобільного досвіду.

ВИСНОВКИ

В ході виконанні дипломної роботи було здійснено повний цикл розробки веб-застосунку для книжкового маркетплейса, який поєднує класичну модель електронної комерції з функцією обміну книгами між користувачами. Робота охоплює теоретичний аналіз предметної області, проектування архітектури платформи, реалізацію клієнтської та серверної частин, а також тестування основних сценаріїв користувацької взаємодії.

На першому етапі було досліджено сучасні технології веб-розробки, використання фреймворків React і Node.js, інтеграцію з хмарними сховищами, підходи до побудови маркетплейсів і механізми обміну книгами.

У процесі проектування було розроблено архітектуру системи, створено базу даних з урахуванням ролей користувачів, реалізовано інтерфейс, адаптований до різних сценаріїв взаємодії: як для гостей відвідувачів, так і для авторизованих користувачів і адміністратора. Також розглянуто модуль обміну, що реалізує повний цикл: від додавання книги до підтвердженого обміну із системою сповіщень і контролем доступу до контактних даних.

Серверна частина забезпечує взаємодію між клієнтським інтерфейсом і базою даних, підтримує захищену авторизацію, а також виконує зберігання зображень і уривків книг у хмарному середовищі. Було реалізовано модерацію контенту, фільтрацію, історію дій користувача та механізм повідомлень. Тестування підтвердило стабільну роботу системи, правильність маршрутизації, обробки запитів і відповідей сервера.

На основі аналізу отриманих результатів можна позначити напрямки подальшого вдосконалення платформи. Наприклад, із часом зростає кількість повторюваних запитів, таких як запити до списку жанрів або популярних книг. Щоб не перевантажувати базу, варто запровадити кешування, наприклад, за допомогою Redis.

Ще одна річ, яку можна вдосконалити – це робота із зображеннями. Наразі вони зберігаються в S3, і все працює стабільно, але для економії простору

логічно було б обмежити розмір файлів ще на етапі завантаження або автоматично стискати їх. Також є сенс підключити CDN, для забезпечення швидкої доставки статичних ресурсів.

На рівні клієнтської частини ефективним буде використання lazy-loading компонентів і зображень для прискорення первинного завантаження, особливо на мобільних пристроях.

Реалізований продукт може слугувати базою для впровадження повноцінного комерційного сервісу або бути використаний як шаблон для аналогічних рішень у сфері електронної торгівлі.

Таким чином, поставлені цілі дипломної роботи було досягнуто, а створений веб-додаток є прикладом сучасного підходу до розробки інформаційних систем з орієнтацією на потреби реального користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шипунова О. В., Єльнікова Ю. В. Автоматизація управління підприємством: основні принципи, функції та підходи. 2010.
2. Фриз М.Р. Історія Інтернету. URL: <https://sites.google.com/view/bezresnyjinternet/історія-інтернету> (дата звернення: 14.02.2025).
3. Історія інтернету від ARPANET до сьогодні. 21.11.2019. URL: <https://ucloud.ua/istoriya-internetu-vid-arpanet-do-sogodni/> (дата звернення: 14.02.2025).
4. Інтернет-додатки. URL: https://stud.com.ua/121246/informatika/internet_dodatki (дата звернення: 14.02.2025).
5. Solomon. Що таке маркетплейс та чим відрізняється від інтернет-магазину. URL: <https://solomono.net/uk/scho-take-marketplejs-ta-chim-vidriznjaiet-sja-vid-internet-magazina-a-523.html> (дата звернення: 18.02.2025).
6. Ткаченко А. Що таке маркетплейс, види, переваги та недоліки. URL: <https://wezom.com.ua/ua/blog/yak-prosuvati-biznes-na-marketpleysah?> (дата звернення: 18.02.2025).
7. Ладуба М. Що таке маркетплейси: принципи роботи, види та приклади успішних майданчиків. URL: <https://mc.today/uk/shho-take-marketplejsi/> (дата звернення: 26.02.2025).
8. Розробка маркетплейса в Україні під ключ. URL: <https://pandateam.net.ua/razrobotka-marketpleysa/> (дата звернення 18.03.2025).
9. Розповідаємо про буккросинг. URL: <https://ecoclubua.com/state/rozpovidayemo-pro-bukkrosyng/> (дата звернення: 28.02.2025).
10. Glodibi, URL: <https://marketplace.glodibi.com/uk> (дата звернення: 20.03.2025).
11. Веб-технології. Їх різновиди та функції. URL: <https://sites.znu.edu.ua/webprog/lect/1170.ukr.html> (дата звернення: 04.03.2025)

12. Що таке веб-додаток: чим сайт відрізняється від веб-додатку? URL:<https://outsourcing.team/uk/blog/development/shho-take-veb-dodatok-chim-sajt-vidriznyayetsya-vid-veb-dodatku/> (дата звернення: 04.03.2025)
13. Себастьян Ч. Веб-додаток і веб-сайт – у чому різниця? 21.02.23. URL:https://firmao.com.ua/blog_net/ua/management/web-application-vs-website-what-are-the-differences#:~:text=Відмінності%20між%20веб-сайтами%20та%20в%20веб-додатками&text=Вебсайти%20як%20правило%20є%20інформаційн%20іми%20та%20надають%20відвідувачам,виконувати%20дії%20та%20отримувати%20відгуки (дата звернення: 04.03.2025).
14. Денис. Як створити веб-додаток: типи, переваги, принцип роботи. 17.04.2023. URL:<https://wezom.com.ua/ua/blog/kak-sozdat-veb-prilozhenie> (дата звернення: 17.03.2025).
15. Лукас Б. Різниця між веб-сайтом і веб-додатком (веб-додаток). 13.08.2024. URL:<https://www.guru99.com/uk/difference-web-applicationwebsite.html> (дата звернення: 18.03.2025).
16. Веб-додаток проти веб-сайту: розкриття переваг кожного рішення. URL: <https://pnn.com.ua/ua/blog/detail/web-application-vs-website-uncovering-benefits-of-each-solution> (дата звернення: 18.03.2025).
17. Що таке вебдодатки та як вони функціонують? URL: <https://asabix.com.ua/what-are-web-applications-and-how-do-they-function/> (дата звернення: 24.03.2025).
18. Роман. Михайлович. Технології для розробки сайту: що це, які найпопулярніші та як вибрати. 22.11.2024. URL: <https://iampm.club/ua/blog/tehnologiyi-dlya-rozrobki-sajtu-shho-cze-yaki-najpopulyarnishi-ta-yak-vibrati/> (дата звернення: 24.03.2025).
19. Структура HTML. 16.10.2016. URL: <http://weplib.com.ua/blog/struktura-html> (дата звернення: 24.03.2025).
20. Що таке CSS? URL: <https://go-mother.com/cssua/> (дата звернення: 24.03.2025).

21. Що таке JavaScript. 22.06.2023. URL: <https://cases.media/article/show-take-javascript> (дата звернення: 26.03.2025).
22. Посібник: знайомство з React. URL: <https://uk.legacy.reactjs.org/tutorial/tutorial.html> (дата звернення: 24.03.2025).
23. Ганна С. Український книжковий ринок продовжує зростати: видавництва, ритейл, тренди. 07.08.2024 URL: <https://hub.kyivstar.ua/articles/ukrayinskij-knizhkovij-rinok-prodovzhuye-zrostati-vidavnicztva-ritejl-trendi> (дата звернення: 26. 03.2025).
24. Аніта П. Україні цьогоріч випуск книжкової продукції зріс на 73%, також побільшало видавців – Мінкульт. 22.12.2023. URL: <https://forbes.ua/news/v-ukraini-tsogorich-vipusk-knizhkovoi-produktsii-zbilshivsya-na-73-minkult-22122023-18104> (дата звернення: 27. 03.2025).
25. Українське книговидання: тенденції та перспективи розвитку галузі. 08.03.2024. URL:<https://mcsc.gov.ua/news/ukrayinske-knygovydannya-tendencziyi-ta-perspektyvu-rozvytku-galuzi/> (дата звернення: 27. 03.2025).
26. Костянтин С. Глобальна електронна комерція: ключові цифри та тренди e-commerce 2024. URL:<https://rau.ua/novyni/trendi-e-com-2024/> (дата звернення: 29. 03.2025).
27. Електронна Комерція: Топ Тенденції та Інновації у 2024 році. URL:<https://youknow.com.ua/2024/04/24/elektronna-komertsii-top-tendentsii-ta-innovatsii-u-2024-rotsi/> (дата звернення: 29. 03.2025).

ЗГОДА здобувачки вищої освіти

Державного університету економіки і технологій про
перевірку кваліфікаційної роботи на прояви
академічного плагіату
та розміщення в Репозитарії Університету

Я, Зацеркляна Тетяна Олегівна (ППП),

підтримую політику Державного університету економіки і технологій з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська робота

Розробка інтерактивного книжкового маркетплейса з обмінним фондом

(назва роботи повністю) виконана самостійно та не містить академічного плагіату. Я не надавала і не одержувала недозволену допомогу під час підготовки цієї роботи. Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Державного університету економіки і технологій ознайомена. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформована, що відповідно до «Положення про Репозитарій (електронну базу даних) Державного університету економіки і технологій» зазначена робота буде розміщена в Електронному архіві Університету (Репозитарії ДУЕТ). З умовами такого розміщення ознайомена.

01.06.2025 р.

Дата



підпис

Т.О.Зацеркляна

ініціали, прізвище