

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту, назва факультету (відділення))

Кафедра обчислювальної техніки
(повна назва кафедри (предметної, циклової комісії))

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна система для обліку товарів у книжковому магазині»

08-54.БКР.002.00.000 ПЗ

Виконала: студентка 2 курсу, групи КІ-23мсз
спеціальності 123 — Комп'ютерна інженерія
(шифр і назва напрямку підготовки, спеціальності)



Єрмакова А. М.

(підпис)

Керівник: ст. викл. каф. ОТ

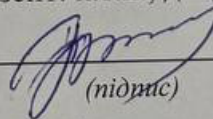


Кисюк Д. В.

(підпис)

«_____» _____ 2025 р.

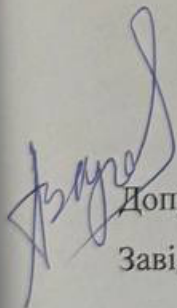
Рецензент: к.т.н., доцент, каф.ПЗ



Хошаба О.М.

(підпис)

«_____» _____ 2025 р.



Допущено до захисту

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

«14» 06 2025 р.

Вінниця ВНТУ — 2025 рік

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії

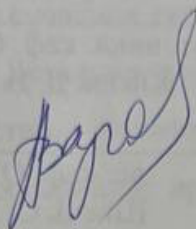
Кафедра обчислювальної техніки

Освітньо-кваліфікаційний рівень бакалавр

Галузь знань — 12 «Інформаційні технології»

Спеціальність — 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерна інженерія»



ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

« 21 » березня 2025 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Єрмаковій Ангеліні Михайлівні

1 Тема роботи: Інформаційна система для ведення обліку товарів у книжковому магазині, керівник роботи Кисюк Дмитро Васильович, затверджено наказом вищого навчального закладу від 20 березня 2025 року №97.

2 Строк подання студентом роботи 13.05.2025

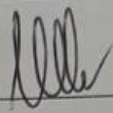
3 Вихідні дані до роботи: сфера розробки — інформаційна система обліку книг та замовлень, цільова платформа — вебсередовище, мови програмування: Java (для backend), JavaScript (для frontend), інструменти розробки: Spring Boot, React, PostgreSQL, протокол обміну даними — REST API, формат — JSON, середовище розробки: IntelliJ IDEA, Visual Studio Code.

4 Зміст розрахунково-пояснювальної записки: вступ; аналіз існуючих прототипів та визначення вимог до створюваної системи; обґрунтування структури та принципів роботи системи; розробка й тестування апаратно-програмної системи; висновки.

5 Перелік графічного матеріалу: структурна схема вебдодатку, ер-діаграма бази даних, ui-макети інтерфейсу користувача, архітектура клієнт-серверної взаємодії, скріншоти роботи вебзастосунку.

6 Консультанти розділів роботи приведені в таблиці 1.

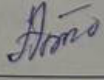
Таблиця 1 — Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	ст. викл. каф. ОТ Кисюк Д. В.	21.03.25 	13.05.25 
Нормоконтроль	ас. каф. ОТ Швець С. І. 	13.05.25	30.05.25

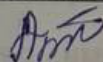
7 Дата видачі завдання 21.03.2025 р.

8 Календарний план приведений в таблиці 2.

Таблиця 2 — Календарний план

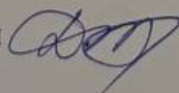
№ з/п	Назва етапу	Термін виконання	Підпис
1	Постановка задачі роботи	21.03.25	
2	Пошук матеріалів та інформаційних джерел	21.03.25— 30.03.25	
3	Аналіз існуючих прототипів та визначення вимог до створюваної системи	21.03.25— 30.03.25	
4	Обґрунтування структури та принципів роботи системи	31.03.25— 13.04.25	
5	Розробка й тестування апаратно-програмної системи	14.04.25— 27.04.25	
6	Оформлення пояснювальної записки та ілюстративного матеріалу	28.04.25— 12.05.25	
7	Перевірка якості виконання бакалаврської кваліфікаційної роботи та усунення недоліків	13.05.25	

Студентка



Ангеліна ЄРМАКОВА

Керівник роботи



ст. викл. каф. ОТ Дмитро КИСЮК

АНОТАЦІЯ

УДК 004.42

Єрмакова А. М. Розроблення інформаційної системи для книжкового магазину. Бакалаврська кваліфікаційна робота зі спеціальності 123 «Інженерія програмного забезпечення», освітня програма «Інженерія програмного забезпечення». Вінниця: ВНТУ, 2025. 65 с.

На укр. мові. Бібліогр.: 28 назв, рис.: 18, табл. 2

У роботі проведено аналіз сучасних технологій розроблення веб-додатків та систем електронної комерції. Детально розглянуто архітектурні рішення для побудови інформаційних систем, включаючи клієнт-серверну архітектуру, принципи REST API та мікросервісну архітектуру. Висвітлено особливості використання реляційних баз даних PostgreSQL для забезпечення надійного збереження та обробки даних. Проаналізовано сучасні фреймворки для розроблення фронтенд-частини (React.js) та бекенд-частини (Spring Boot) веб-додатків. Особлива увага приділена питанням безпеки веб-додатків, включаючи аутентифікацію користувачів, захист від SQL-ін'єкцій та шифрування даних. На основі проведеного аналізу розроблено повнофункціональну інформаційну систему для книжкового магазину з можливостями управління каталогом товарів, обробки замовлень, управління користувачами та адміністративною панеллю. Реалізовано контейнеризацію додатку за допомогою Docker для забезпечення зручного розгортання та масштабування системи. Розроблений додаток може бути адаптований для інших типів онлайн-магазинів та використовуватися як основа для створення e-commerce платформ.

Ключові слова: інформаційна система, електронна комерція, веб-додаток, React.js, Spring Boot, PostgreSQL, Docker, безпека даних, REST API.

ABSTRACT

UDC 004.42

Yermakova A. M. Development of Information System for Bookstore. Bachelor's thesis in specialty 121 "Software Engineering", educational program "Software Engineering". Vinnytsia: VNTU, 2025. 65 p.

In Ukrainian. Bibliography: 28 titles, figures: 18, tables: 2.

The paper analyzes modern technologies for web application development and e-commerce systems. Architectural solutions for building information systems are examined in detail, including client-server architecture, REST API principles, and microservice architecture. The features of using PostgreSQL relational databases to ensure reliable data storage and processing are highlighted. Modern frameworks for frontend development (React.js) and backend development (Spring Boot) of web applications are analyzed. Special attention is paid to web application security issues, including user authentication, protection against SQL injections, and data encryption. Based on the conducted analysis, a full-featured information system for a bookstore has been developed with capabilities for product catalog management, order processing, user management, and administrative panel. Application containerization using Docker has been implemented to ensure convenient deployment and system scaling. The developed application can be adapted for other types of online stores and used as a foundation for creating e-commerce platforms.

Keywords: information system, e-commerce, web application, React.js, Spring Boot, PostgreSQL, Docker, data security, REST API.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ОСНОВНИХ ПОНЯТЬ.....	6
1.1 Поняття інформаційної системи	6
1.2 Поняття веб-сторінок та їх типи	7
1.3 Основні види сайтів	8
1.4 Огляд сучасних інформаційних систем для інтернет-магазинів.....	10
1.5 Аналіз вимог до інформаційної системи для інтернет-магазину книг	14
1.6 Дослідження впливу дизайну та функціональності інформаційної системи на задоволення та ефективність користувачів	17
1.7 Розгляд методів та технологій для оптимізації інтернет-магазину для пошукових систем та просування продуктів.....	21
1.8 Огляд методів та технологій для забезпечення прозорості та етичності в інтернет-магазинах	23
1.9 Аналіз впливу мобільних технологій на електронну комерцію.....	25
1.10 Створення динамічних вебсторінок JavaScript	26
2 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ	28
2.1 Огляд технологій для розробки системи.....	28
2.2 Логічна та фізична структура вебдодатку.....	33
3 ПРОЄКТНА ЧАСТИНА	35
3.1 Обґрунтування та вибір колірної гами вебдодатку	35
3.2 Інтерактивна взаємодія з каталогом	37
3.3 Динамічні компоненти клієнтської частини вебдодатку	40
3.4 Реалізація адмін-панелі та CRUD-функцій	41
4 ТЕСТУВАННЯ ПРАЦЕЗДАТНОСТІ РОЗРОБЛЕНОЇ ПЛАТФОРМИ....	48
4.1 Оптимізація відображення слайдера	48
4.2 Додавання товарів в кошик	59
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56

ДОДАТОК А Технічне завдання	ЗМІСТ.....	58
ДОДАТОК Б Протокол перевірки кваліфікаційної роботи		62
ДОДАТОК В Ілюстративна частина		63
ДОДАТОК Г Лістинги інтерактивності сторінок		66

ВСТУП

З поширенням інтернету та розвитком електронної комерції онлайн-магазини стали невід’ємною частиною сучасного бізнесу. Покупка книг через інтернет — це зручно, швидко та ефективно. Проте для повноцінного функціонування такого магазину необхідна надійна інформаційна система, яка автоматизує процеси управління товарами, замовленнями, обслуговуванням клієнтів та обробкою даних.

Актуальність створення інформаційної системи для книжкового магазину зумовлена зростанням попиту на онлайн-продажі, необхідністю зручного пошуку книг, швидкого оформлення замовлень та інтеграції з платіжними системами. Така система дозволяє значно підвищити ефективність бізнес-процесів, покращити взаємодію з клієнтами та забезпечити стабільність роботи магазину.

Метою бакалаврської кваліфікаційної роботи є розробка інформаційної системи для книжкового онлайн-магазину, яка дозволить автоматизувати ключові бізнес-процеси та забезпечить користувачам зручну навігацію, фільтрацію товарів, перегляд каталогу та оформлення замовлень.

Завдання дослідження:

- провести огляд сучасних інформаційних систем для онлайн-магазинів книг;
- дослідити вимоги цільової аудиторії та сформулювати функціональні вимоги до системи;
- розробити логічну та фізичну модель бази даних;
- реалізувати інтерфейс з фільтрами, слайдерами, кошиком та адмін-панеллю;
- протестувати роботу системи на прикладі реального каталогу книжок.

Об’єктом дослідження є процес створення вебзастосунків для онлайн-торгівлі.

Предметом дослідження є інформаційна система для управління книжковим магазином.

Методи дослідження:

- аналіз існуючих рішень у сфері онлайн-продажу книжкової продукції;
- проектування та моделювання системи на основі сучасних архітектурних підходів (клієнт-сервер);
- використання технологій React, Spring Boot, PostgreSQL та Docker для реалізації;
- перевірка працездатності та тестування функціоналу.

Наукова новизна роботи полягає в інтеграції сучасних технологій веброзробки з гнучкою архітектурою, що забезпечує ефективну роботу системи, зручність для користувачів та можливість масштабування. Практична цінність роботи полягає в можливості адаптації цієї системи для різних сфер електронної комерції.

1 АНАЛІЗ ОСНОВНИХ ПОНЯТЬ

1.1 Поняття інформаційної системи

Інформаційна система (ІС) — це сукупність взаємозалежних елементів, що забезпечують збирання, обробку, зберігання, пошук, передачу та подання інформації, яка необхідна для підтримки процесів прийняття рішень, управління та контролю в організації або установі [1].

Згідно з міжнародним стандартом ISO/IEC 2382, інформаційна система — це система для збирання, обробки, зберігання і розповсюдження інформації, що складається з апаратного і програмного забезпечення, процедур і персоналу [40].

Також в українській науковій літературі часто подається визначення, згідно з яким ІС — це взаємозв'язана сукупність технічних, програмних, інформаційних і організаційних засобів, призначених для автоматизації інформаційних процесів в конкретній предметній області [2].

Інформаційна система зазвичай включає три основні групи компонентів:

— технічні засоби (hardware) — комп'ютери, сервери, мережеве обладнання, засоби зберігання даних, в контексті вебдодатку це, наприклад, сервер, де розгорнуто backend-програму, хостинг бази даних, маршрутизатори тощо;

— програмне забезпечення (software) — операційні системи, системи управління базами даних (СУБД), вебсервери, середовища розробки та самі прикладні системи (вебсайти, CRM, ERP), в межах цієї роботи використано такі компоненти, як React (інтерфейс), Spring Boot (бізнес-логіка), PostgreSQL (зберігання даних);

— персонал (peopleware) — користувачі та спеціалісти, які проєктують, реалізують, підтримують або користуються системою: адміністратори, розробники, менеджери, кінцеві користувачі.

Ці компоненти взаємодіють між собою, формуючи єдину інформаційну екосистему [3].

Класифікація інформаційних систем за сферою застосування:

— адміністративні (для державних структур);

- корпоративні (для підприємств — ERP, CRM тощо);
- науково-дослідні (використовуються в науці, медицині, інженерії);
- комерційні (включно з інтернет-магазинами).

Класифікація інформаційних систем за функціональністю:

- оперативні(здійснюють поточні транзакції, наприклад: покупка товару);
- аналітичні(обробляють великі масиви даних для ухвалення рішеньВІ-системи);
- інформаційно-довідкові (призначені для пошуку, зберігання та надання інформації).

Класифікація інформаційних систем за способом обробки даних:

- централізовані — обробка на одному сервері або в одному середовищі;
- розподілені — дані обробляються на кількох вузлах (наприклад, мікросервісна архітектура).

1.2 Поняття веб-сторінок та їх типи

Вебсторінка – це документ, доступний з будь-якого браузера з доступом до Інтернету, і він може включати аудіо, відео, текст та їх різні комбінації.

Перші вебсайти з'явилися в 1992 році. Вони були статичними сторінками, і їх можна було змінити лише для зміни HTML-коду. На цих вебсайтах переважали кольори та зображення. Вони мали привертати увагу, але їм було байдуже щодо зручності та доступності.

Багато з цих домашніх сторінок дратували музику та анімацію. У 1997 році Flash з'явився як новаторська програма у створенні вебсайтів, і вони почали розроблятися за допомогою цієї технології, яка характеризується підвищеною анімацією, повільним завантаженням та кількістю відвідувань.

З 2002 року з'являються перші блоги, а також деякі успішні соціальні мережі. Інші типи інструментів також використовуються для створення вебсторінок, які більше не прагнуть бути такими ефектними, а покращити зручність користування користувачами.

Однією з найбільш широко використовуваних систем управління вмістом (CMS) є WordPress. Користувачі шукають поліпшення копірайтингу, зручності використання та уникнення нудних анімацій, що заважають мережі сповільнюватися.

Типи вебсторінок:

- статичні вебсторінки — такі сторінки, на яких зміст змінити неможливо, бо з'являється постійна інформація, і користувач не може взаємодіяти;
- динамічні вебсторінки, на яких користувачі можуть створювати та співпрацювати над вмістом, вони припускають протилежність попереднім і серед них можна виділити: блоги, форуми, інтернет-магазини;
- блоги — це портали, на яких автор зазвичай публікує інформативні статті, пов'язані з певною темою, вони сьогодні дуже популярні, оскільки служать для додавання вартості аудиторії та демонстрації досвіду професіоналів, які їх пишуть;
- інтернет-магазини — це сторінки, орієнтовані на купівлю та продаж предметів, багато фізичних платформ виявили потенціал інтернету та ведуть власні інтернет-магазини, щоб товари можна було придбати з будь-якої точки світу;
- пошукові системи — це сторінки, завдяки яким користувачі можуть шукати вміст, що цікавить певну тему.

1.3 Основні види сайтів

Вебсайт – сукупність вебсторінок та залежного вмісту, доступних у мережі Інтернет, які об'єднані як за змістом, так і за навігацією під єдиним доменним ім'ям, фізично сайт може розміщуватися як на одному, так і на кількох серверах [6].

Сайт-візитівка:

- найчастіше односторінковий сайт, або такий, що має головну сторінку та кілька вкладок, де можна знайти профільну інформацію, наприклад: головна сторінка про фірму та вкладки про історію створення, контакти, продукти тощо;

- має на меті ознайомити користувача із діяльністю компанії;
- не передбачає конверсії;
- найпростіший у створенні, найдешевший та передбачений для тих оунерів, що не ведуть бізнес в Інтернеті.

Лендінг:

- це сайт-презентація, де Ви можете представити свої послуги чи продукт;
- найголовніша ознака лендінгу – зробити якусь дію, наприклад, окрім розсилки, користувачі можуть купити квиток, зареєструватись на подію або одразу скачати документ чи презентацію;
- зазвичай користувачі просто гортають сайт, та шукають так всю інформацію;
- недорогий, ефективний, аніж сайт-візитка, передбачений для власників бізнесу, що ведуть справи в інтернеті.

Каталог:

- це сайт із великою кількістю сторінок, розділів та підрозділів;
- основним завданням сайту-каталогу є презентація переліку товарів чи послуг;
- клієнти будуть мати постійний доступ до прайсів, описів продукції, нових надходжень товарів;
- це не інтернет-магазин, тут клієнт може зробити замовлення зателефонувавши за контактним номером телефону, або написавши на пошту та уточнити замовлення;
- не передбачає розробки складного функціоналу.

Корпоративний сайт:

- формує імідж компанії, залучає нових клієнтів, розширює вплив та створює позитивну репутацію;
- містить повну інформацію про фірму, функціонал зв'язку, чат-боти, сервіси замовлення послуг або продуктів;
- має складний функціонал, відносно дорогий у розробці та підтримці, потребує великої кількості часу для створення та тестування.

Інтернет-магазин:

- на відміну від сайту-каталогу, тут розроблений функціонал, що дозволяє зробити замовлення, вибрати варіант розрахунку, спосіб отримання замовлення та одержати рахунок на оплату;
- спрямований на залучення лідів та клієнтів, створюється для продуктових компаній, що пропонують широкий спектр товарів чи послуг;
- сайт має постійно оновлюватись, і при цьому не заважати потоку клієнтів;
- сайт має бути гарно протестований на стрес та лод навантаження;
- зазвичай синхронізується з різними CRM системами для зручної комунікації менеджерів та клієнтів.

Портал:

- це сайт з посиланням на інші сайти з подібної тематики;
- тут можливий функціонал чату, переписки, коментарів щодо обраної тематики;
- портали зазвичай створюються для вирішення конкретних завдань, тобто, як правило, є частиною бізнес-плану з відомою схемою монетизації;
- створення порталу, як правило, має на увазі досить складну розробку, що зачіпає безліч аспектів від забезпечення швидкодії ресурсу на конкретному обладнанні до агресивних тестів з інформаційної безпеки.

1.4 Огляд сучасних інформаційних систем для інтернет-магазинів

У сучасному світі електронна комерція набула стрімкого розвитку. Онлайн-магазини книг стають зручним інструментом для придбання літератури, оскільки дають змогу легко обрати потрібне видання, прочитати опис, відгуки, порівняти ціни та оформити доставку. Для ефективної роботи таких сервісів необхідні інформаційні системи, що автоматизують основні бізнес-процеси: управління асортиментом, обробку замовлень, реєстрацію користувачів, а також зв'язок з платіжними та логістичними платформами [4].

На сьогоднішній день, для ефективної роботи, будь-яке успішне підприємство, організація чи фірма потребують інтернет ресурс для просування свого бренду: розміщення інформації про свою продукцію чи послуг, створення зручних умов для клієнтів та свого персоналу. З задачею написання сайту можна звернутись до веброзробника, але спочатку необхідно визначитись, як буде реалізовано даний сайт, який функціонал він повинен містити та які функції виконувати.

Специфіка вебсайту залежить від кількох факторів, зокрема:

- призначення та мети, адже кожен вебсайт має свою унікальну мету та призначення, наприклад, це може бути особистий блог, інтернет-магазин, сайт компанії, форум, портфоліо або будь-який інший тип сайту, коли призначення та використання визначають функціональність, дизайн і зміст вебсайту;

- цільової аудиторії, оскільки вебсайт повинен відповідати потребам і очікуванням цільової аудиторії, наприклад, вебсайт бренду молодіжної моди виглядатиме та працюватиме інакше, ніж вебсайт фінансової компанії, вік аудиторії, інтереси, географічне розташування та інші характеристики впливають на специфіку сайту;

- функціональних вимог, бо для різних типів сайтів потрібні різні функції, наприклад, для сайту блогу можуть знадобитися системи керування вмістом, огляди та інтеграція соціальних медіа, тоді як для магазину електронної комерції можуть знадобитися каталоги продуктів, кошики для покупок і можливості онлайн-платежів, функціональні вимоги можуть впливати на вибір платформи або інструменту, який використовується для створення сайту;

- графічний дизайн та візуальна концепція сайту також впливають на його специфіку, дизайн повинен відповідати бренду або стилю сайту, створювати позитивне враження на користувачів та підкреслювати його цілі, бо це охоплює вибір кольорової палітри, шрифтів, макету, зображень та загального вигляду сайту.

Існує багато способів створення інформаційних систем, зокрема веборієнтованих, які активно використовуються для автоматизації внутрішніх

бізнес-процесів. У контексті розробки облікової системи для книжкового магазину доцільно розглянути як класичні підходи, так і сучасні фреймворки та технології, які дозволяють створювати масштабовані та зручні рішення.

Інформаційні системи інтернет-магазинів зазвичай створюють на основі сучасних CMS, CRM або власних розробок. Найпопулярнішими серед CMS для книжкових інтернет-магазинів є WordPress та WooCommerce — безкоштовні системи з відкритим кодом, яка завдяки додатку WooCommerce дозволяє створити повноцінний магазин книг. WooCommerce підтримує розділення книг за жанрами, авторами, роками видання, а також дозволяє керувати відгуками та рейтингами [2].

WordPress — це одна з найпопулярніших у світі систем управління контентом (CMS), що дозволяє створювати сайти різного рівня складності. WooCommerce — це плагін для WordPress, який перетворює сайт на повноцінний інтернет-магазин. Ця платформа активно використовується в книжковій сфері завдяки гнучким можливостям налаштування структури каталогу, сторінок товарів, фільтрації та відгуків (рисунок 1.1).

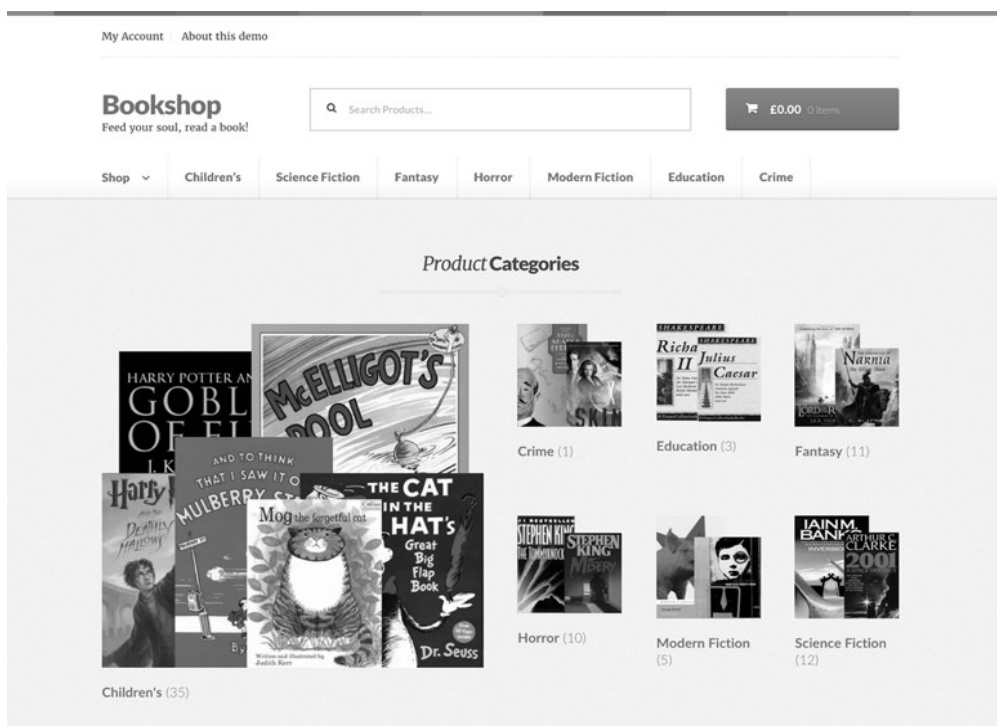


Рисунок 1.1 — Інтерфейс магазину книг на базі WooCommerce

WooCommerce дозволяє адміністратору:

- створювати категорії за жанрами, авторами, мовами, роками видання;
- прикріплювати обкладинки, анотації, ISBN-коди;
- керувати акціями, рейтингами, відгуками читачів;
- інтегрувати платіжні шлюзи та методи доставки;
- налаштувати SEO-поля для кожної книги.

Це рішення з відкритим кодом, що дозволяє легко розширювати функціонал за допомогою плагінів або кастомного коду.

OpenCart — платформа, що надає зручну адміністративну панель, підтримує багатомовність, різні способи доставки та оплати. Часто використовується малими та середніми книжковими крамницями [3]. Це ще одна CMS, спеціалізована для створення онлайн-магазинів. Вона легка у використанні, має простий інтерфейс та є безкоштовною. OpenCart особливо популярна серед невеликих книгарень через мінімальні вимоги до серверу та наявність багатьох шаблонів для книг (рисунк 1.2).

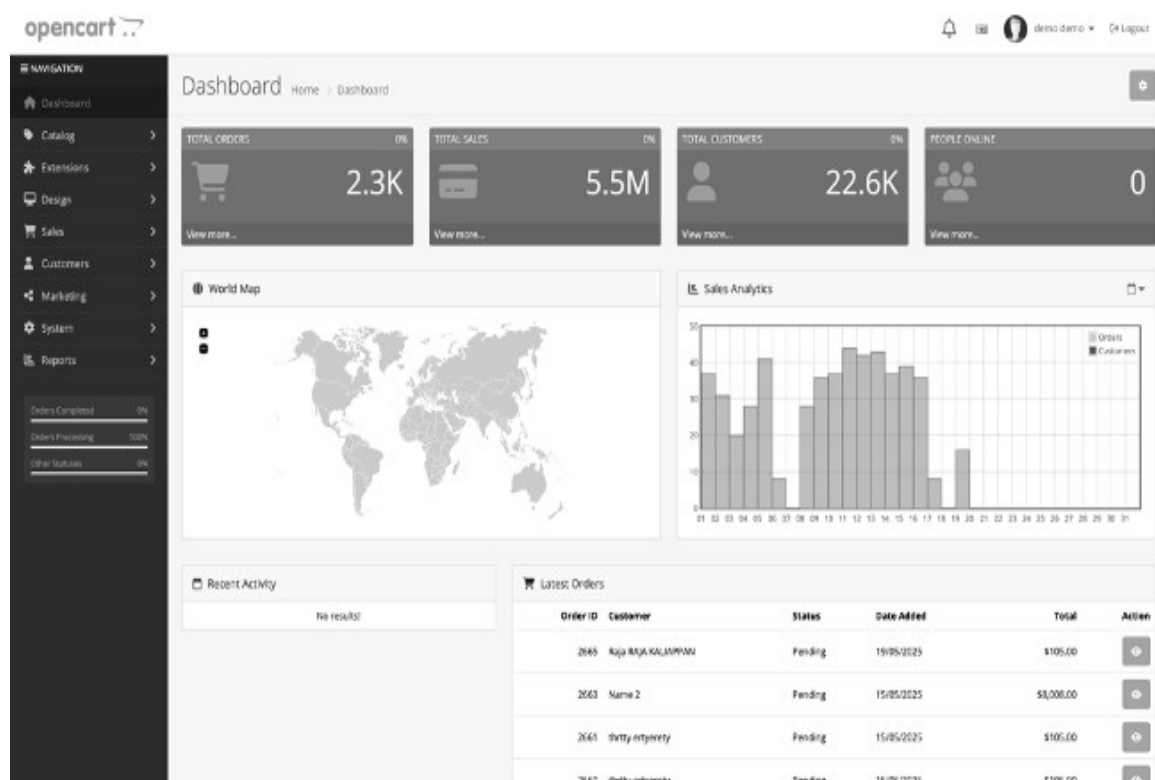


Рисунок 1.2 — Інтерфейс налаштування OpenCart

Можливості OpenCart включають:

- додавання книжок з докладними характеристиками;
- фільтрацію за тегами, категоріями, ціною;
- підтримку кількох валют і мов (для міжнародних магазинів);
- систему купонів, знижок, оптових цін;
- підключення доставки (Nova Poshta, Ukrposhta тощо).

Користувач отримує просту і зручну адміністративну панель для керування товарами, замовленнями, клієнтами.

Joomla + VirtueMart — гнучка CMS із широким вибором шаблонів та розширень. Використовується для створення книжкових магазинів з розвинутою системою категоризації книг [4]. Це розширювана CMS, що дозволяє керувати як простими, так і складними сайтами. У поєднанні з компонентом VirtueMart вона може слугувати основою для створення онлайн-магазинів книжок. VirtueMart — це eCommerce-розширення з підтримкою великої кількості типів товарів, включно з електронними книгами (PDF, EPUB).

Переваги Joomla + VirtueMart:

- чітке розмежування товарів за категоріями: художня література, навчальна, дитяча;
- підтримка багатоуровневої структури розділів;
- можливість створення кількох типів користувачів (менеджери, автори, адміністратори);
- вбудований механізм SEO-оптимізації;
- гнучка система шаблонів для оформлення книжкової вітрини.

1.5 Аналіз вимог до інформаційної системи для інтернет-магазину книг

Інформаційна система для інтернет-магазину книг повинна відповідати сучасним вимогам до електронної комерції, враховуючи як технічні аспекти функціонування, так і потреби кінцевого користувача. У цьому підрозділі розглянемо основні вимоги до функціональності, безпеки, зручності,

масштабованості та підтримки системи на прикладі книжкової онлайн-платформи.

Функціональні вимоги визначають основні можливості, які повинна забезпечити система. Для книжкового інтернет-магазину вони включають такі компоненти:

Каталог книг:

- категоризація за жанром, автором, видавництвом, мовою, роком видання [5];
- відображення обкладинки, опису, анотації, ISBN;
- підтримка фільтрів та сортування за популярністю, новизною, ціною тощо.

Система пошуку:

- повнотекстовий пошук за назвою книги, автором або ключовими словами;
- автопідказки при введенні пошукового запиту [6].

Кошик та оформлення замовлення:

- додавання та видалення товарів з кошика;
- оформлення замовлення з вибором способу доставки та оплати;
- підтвердження замовлення через email або особистий кабінет.

Користувацький обліковий запис:

- реєстрація та авторизація (в тому числі через Google/Facebook);
- перегляд історії замовлень, збереження книг до списку бажань;
- редагування особистої інформації.

Адміністративна панель:

- додавання, редагування та видалення книг;
- управління категоріями та тегами;
- перегляд статистики продажів, замовлень, активності користувачів [7].

Юзабіліті (зручність у використанні):

- інтуїтивний інтерфейс;
- адаптивна верстка для мобільних пристроїв;

- швидке завантаження сторінок (менше 2 сек.) [8].

Безпека:

- зберігання паролів у хешованому вигляді;
- захист від SQL-ін'єкцій, XSS, CSRF;
- шифрування трафіку (HTTPS);
- обмеження доступу до адмін-панелі.

Масштабованість:

- можливість підключення нових модулів (наприклад, програма лояльності);
- підтримка великої кількості товарів та користувачів.

Підтримка мультимовності:

- локалізація інтерфейсу;
- можливість додавання описів книг кількома мовами [9].

Інтеграція з такими сервісами є важливою умовою для повноцінної роботи:

- платіжні системи (LiqPay, Stripe, PayPal);
- служби доставки (Нова Пошта, Укрпошта, Meest);
- сторонні API для перевірки ISBN або відгуків (Goodreads API);
- поштові сервіси (SMTP або SendGrid) для надсилання повідомлень.

База даних повинна відповідати таким умовам:

- зберігати інформацію про книги, користувачів, замовлення, платежі, доставки;
- мати чітко продуману реляційну структуру (ER-діаграма);
- забезпечувати резервне копіювання;
- підтримувати зв'язки типу "багато до багатьох" (наприклад, автори <-> книги) [10].

Юридичні та етичні вимоги:

- відповідність Закону України "Про захист персональних даних";
- наявність політики конфіденційності та умов використання;
- прозоре інформування про умови доставки, оплати, повернення товарів;
- підтримка авторського права на електронні книги (у разі надання).

Інформаційна система для книжкового інтернет-магазину повинна бути багатофункціональною, безпечною, адаптивною до нових вимог та відкритою до розширення. Її розробка вимагає врахування як технічних, так і правових аспектів. Якісно сформульовані вимоги на цьому етапі забезпечують успішну реалізацію наступних етапів розробки системи.

1.6 Дослідження впливу дизайну та функціональності інформаційної системи на задоволення та ефективність користувачів

У сучасному цифровому середовищі, де конкуренція серед онлайн-платформ зростає щодня, роль дизайну та функціональності інформаційних систем стає ключовою для досягнення високої користувацької задоволеності й ефективності. Для інтернет-магазинів, особливо книжкових, де важливо створити довіру, зручність навігації та емоційний зв'язок із продуктом, дизайн та функціональність є не лише технічними характеристиками, а й важливими складовими стратегічного успіху [11].

Дизайн — це не просто зовнішнє оформлення, а спосіб комунікації з користувачем. У книжкових онлайн-магазинах естетично приємний, комфортний інтерфейс здатен викликати відчуття надійності, знайомості та заохочення до покупки. Теорія емоційного дизайну Дональда Нормана [12] підкреслює, що гармонійне візуальне середовище знижує рівень тривожності користувача та сприяє довготривалій взаємодії з системою.

Важливо:

- застосування кольорів, які не перевантажують сприйняття (наприклад, пастельні тони або м'який контраст);
- використання знайомих шаблонів взаємодії, як-от типова навігація «каталог – кошик – оформлення»;
- розташування ключових елементів (кнопок «Купити», описів книг) у зонах візуальної уваги.

UX стосується загального досвіду користувача під час використання вебресурсу: наскільки зручно і швидко він може знайти потрібну інформацію,

оформити замовлення, виконати інші дії. UX враховує логіку переходів, ефективність сценаріїв взаємодії, адаптивність до пристроїв, швидкість роботи сайту [6].

UI — це безпосередньо візуальний інтерфейс, з яким взаємодіє користувач. Сюди входять кнопки, меню, шрифти, кольори, розміщення елементів, анімації тощо. UI-дизайн повинен бути не лише естетично привабливим, але й інтуїтивно зрозумілим [7].

Разом UX і UI формують сприйняття користувачем вебдодатку, впливають на задоволення від взаємодії та готовність повертатися до ресурсу знову.

User Interface (UI) має забезпечувати інтуїтивність у використанні. Якщо користувач не витрачає зусиль на те, щоб зрозуміти, як працює сайт — це ознака хорошого дизайну. Успішна навігація базується на таких засадах:

- логічне розміщення меню, фільтрів, кнопок;
- передбачуваність дій системи при натисканні;
- відсутність надмірної кількості дій до досягнення мети [8].

Дослідження показують, що оптимальна кількість кліків до здійснення покупки повинна бути п'яти. Кожна додаткова дія — потенційний ризик втрати користувача [10].

Оформлення вебсторінки має психологічний та емоційний вплив на користувача, навіть якщо це не усвідомлюється.

- кожен колір викликає певні емоції, наприклад, синій асоціюється зі стабільністю, довірою, спокоєм — тому часто використовується в інтернет-магазинах оранжевий — активізує увагу, асоціюється з енергією та закликами до швидкої дії (наприклад, кнопка «Купити»), у проєкті магазину книг було обрано поєднання спокійного синього кольору для фону з контрастними елементами для акцентів;

- від правильного вибору шрифту залежить зчитуваність тексту, надто декоративні чи дрібні шрифти ускладнюють сприйняття, а от класичні — наприклад, Roboto, Open Sans — сприяють зручному читанню, у проєкті

використано сучасний читабельний шрифт, який добре адаптується до різних екранів;

— вдале розташування блоків дозволяє користувачеві швидко зорієнтуватися, адаптивна верстка (responsive design) забезпечує коректне відображення на мобільних, планшетах і десктопах, грамотна структура верстки з правильними відступами, розмірами кнопок і візуальними маркерами сприяє комфорту [11].

Інтерфейс — перше, що бачить користувач. Недостатньо лише мати функціональний сайт, він має викликати довіру. Користувач підсвідомо оцінює:

- чи професійно виглядає сайт;
- наскільки легко знайти товар;
- чи безпечно виглядає форма оплати;
- як швидко сторінка реагує на дії.

Якщо сайт має застарілий або перевантажений інтерфейс, користувач може покинути його ще до оформлення покупки. Водночас інтуїтивно зрозумілий інтерфейс із правильним кольоровим акцентом на кнопках заклику до дії (СТА), чіткими категоріями та спрощеною навігацією підвищує конверсію та рівень задоволення користувачів [24].

Таким чином, дизайн — це не лише естетика, а стратегічний інструмент, який напряду впливає на ефективність інформаційної системи. У випадку інтернет-магазину, грамотний UX/UI підвищує тривалість сесії, знижує кількість відмов та збільшує продажі.

Функціональність визначає, наскільки повноцінно система задовольняє потреби користувача. У контексті книжкового магазину це:

- можливість швидкого пошуку за назвою, автором, ISBN;
- фільтрація результатів (жанр, мова, рік видання);
- збереження історії переглядів, списків бажаного;
- рекомендаційні алгоритми (на основі жанрових уподобань, популярності тощо);
- прозоре, зручне оформлення замовлення.

Наявність таких функцій не лише полегшує виконання цілей, а й сприяє формуванню позитивного досвіду користувача, що підтверджено результатами багатьох UX-досліджень [15].

Більше 60% покупок у сфері B2C здійснюється з мобільних пристроїв [16]. Відтак мобільна адаптація системи стає не побажанням, а необхідністю. Інтерфейс має масштабуватись без втрати логіки, елементів навігації та функціоналу.

Використання гнучких сіток, медіа-запитів, мобільного меню, а також елементів типу «запам'ятати останню дію» значно покращує мобільний UX і зменшує кількість відмов.

Систематичне використання фірмового стилю, логотипу, типографіки та кольорової схеми створює ефект знайомства, що підвищує довіру. Згідно з дослідженнями Nielsen Norman Group [17], візуальна цілісність та узгодженість усіх елементів знижує рівень когнітивного напруження та сприяє утриманню уваги.

Особливо важливим є візуальне представлення книг: обкладинка, форматування опису, розділення блоків (наприклад, «Про автора», «Схожі книги»).

Чим більше користувач відчуває, що система підлаштована під нього, тим вищий рівень задоволення. Інтерактивні функції включають:

- «розумні» рекомендації;
- адаптацію контенту на основі поведінки користувача;
- інтеграцію зі сторонніми обліковками (Google, Facebook) для спрощення входу;
- сповіщення про новинки в обраних жанрах.

Такі можливості підвищують ефективність виконання цілей користувача та його лояльність до платформи.

Щоб об'єктивно оцінити якість дизайну й функціоналу, використовують такі метрики:

- Time on Task – час на виконання типових завдань (наприклад, додавання книги в кошик);

- Success Rate – відсоток успішного завершення завдань;
- Error Rate – кількість помилкових дій;
- Net Promoter Score — готовність користувача рекомендувати сервіс.

Ці метрики дозволяють ітеративно покращувати інтерфейс, ґрунтуючись на реальних даних.

Таким чином, дизайн і функціональність інформаційної системи є взаємопов'язаними аспектами, які безпосередньо впливають на ефективність використання системи, рівень задоволеності, а отже — і комерційний успіх онлайн-магазину. Забезпечення інтуїтивного, доступного й емоційно позитивного досвіду користувача має бути ключовим пріоритетом під час розробки будь-якої сучасної інформаційної системи.

1.7 Розгляд методів та технологій для оптимізації інтернет-магазину для пошукових систем та просування продуктів

У сучасному цифровому просторі конкурентоспроможність інтернет-магазину багато в чому залежить не лише від його функціональності та зручності, але й від видимості у пошукових системах. Для інтернет-магазину книг це особливо актуально, оскільки потенційні покупці часто шукають літературу за запитами в Google, Bing чи інших пошуковиках. Тому оптимізація під пошукові системи (SEO) та стратегічне просування контенту мають ключове значення [18].

Search Engine Optimization — це комплекс дій, спрямованих на покращення позицій сайту в органічній видачі пошукових систем. Для книжкових онлайн-магазинів оптимізація включає:

SEO-дружні URL: наприклад, `bookshop.com/books/classics/1984-orwell` замість `bookshop.com?id=4832`;

- унікальні мета-теги (title, description) для кожної книги та категорії;
- оптимізовані заголовки H1–H3, що чітко вказують на зміст сторінки;
- використання ключових слів, пов'язаних із темою книги, жанром, автором [19].

Крім контенту, пошукові системи враховують технічні параметри:

- швидкість завантаження сторінки (не більше 2-3 сек);
- адаптивний дизайн, який коректно відображається на мобільних пристроях;

- структуровані дані (schema.org для книг, авторів, рейтингів);
- коректна XML-карта сайту та robots.txt [20].

Впровадження цих елементів підвищує шанси на потрапляння у розширені результати пошуку (rich snippets).

Окрім технічного SEO, важливою складовою є контент:

- відгуки користувачів;
- блоги про літературу, огляди новинок, добірки «топ-10 книг місяця»;
- сторінки авторів з біографією, фото, списком творів;
- рецензії на популярні книги з ключовими словами [21].

Такий контент:

- затримує користувача на сайті (що позитивно впливає на SEO);
- генерує природні зворотні посилання (backlinks);
- залучає трафік із соціальних мереж та пошуку одночасно.

Якщо інтернет-магазин також має фізичну локацію або пункт самовивозу — варто використовувати локальну оптимізацію:

- реєстрація в Google Мій Бізнес;
- додавання актуального розкладу роботи, адреси, контактів;
- збір відгуків у Google Maps;
- згадування регіону в мета-описах і контенті.

Це дозволяє потрапити у блок локального пошуку, особливо для запитів типу «купити книги Дніпро» [22].

SEO дає довготривалий ефект, але вимагає часу. Для швидких результатів використовують контекстну рекламу (PPC):

- Google Ads (по ключових запитах, наприклад, «книга 1984 купити»);
- соціальні мережі: реклама добірок книг в Instagram, Facebook;
- партнерські програми (CРА-маркетинг з блогерами чи книжковими платформами).

Такі методи дозволяють охопити аудиторію, яка ще не знайома з брендом, але активно цікавиться читанням.

Для оцінки ефективності SEO і рекламних кампаній важливо постійно аналізувати: показники переходів із пошуку (organic traffic); CTR заголовків; час перебування на сайті та глибину перегляду; конверсії (перегляд → покупка).

Використовуються інструменти: Google Analytics, Search Console, Ahrefs, Semrush, Hotjar тощо [21].

Оптимізація інтернет-магазину книг для пошукових систем є комплексним завданням, яке включає як технічні елементи, так і маркетингові стратегії. Лише поєднання SEO, якісного контенту, локальної оптимізації та рекламного просування забезпечує стабільне зростання відвідуваності та продажів.

1.8 Огляд методів та технологій для забезпечення прозорості та етичності в інтернет-магазинах

У сучасному електронному бізнесі прозорість та етичність функціонування онлайн-магазинів стали одним із найважливіших факторів формування довіри користувачів. Для інтернет-магазину книг це особливо актуально, адже покупці очікують не лише якісний товар, а й відкритість у політиках обслуговування, захисті персональних даних, поверненнях і дотриманні авторських прав [12].

Прозорість — це здатність компанії відкрито й доступно доносити до клієнтів інформацію про свої послуги, умови продажу та обслуговування.

Також важливо надавати чесні і точні описи товарів, зображення обкладинок, характеристики (кількість сторінок, мова, ISBN), а також дату останнього оновлення даних.

Інтернет-магазини зобов'язані відповідати чинному законодавству у сфері захисту персональних даних (в Україні – Закон "Про захист персональних даних"). Основні вимоги включають:

- запит згоди користувача на обробку персональних даних;
- можливість редагування або видалення облікового запису;
- чітке роз'яснення, які дані збираються (email, ПІБ, адреса доставки);

- шифрування переданих даних через HTTPS;
- обмеження доступу до даних третім особам [14].

Прозора політика конфіденційності повинна бути доступною на сайті, містити просту мову й вичерпну інформацію.

Книжкові інтернет-магазини зобов'язані дотримуватись авторських прав:

- продаж електронних книг дозволений лише за наявності ліцензій або договорів з правовласниками;
- зображення обкладинок повинні використовуватись із дозволом або відповідно до умов публікацій;
- необхідно вказувати авторів та видавництва кожної книги;
- рецензії користувачів не повинні містити плагіат або дискримінаційні висловлювання [15].

Порушення авторського права загрожує не лише блокуванням ресурсу, але й юридичними санкціями, тому дотримання етичних норм — це не лише добра практика, а й необхідність.

Справжні відгуки — цінне джерело інформації як для покупців, так і для бізнесу. Етичне ставлення до цього аспекту включає:

- заборону фальсифікації відгуків або видалення негативних без причини;
- можливість відповіді адміністрації магазину на коментарі;
- перевірку відгуків на мову ворожнечі чи порушення правил поведінки [13].

Довіра до магазину підвищується, якщо користувач бачить як позитивні, так і критичні оцінки з адекватною реакцією на них.

Сучасні покупці все частіше звертають увагу на соціальну відповідальність брендів. Для книжкового інтернет-магазину це може включати:

- підтримку благодійних проєктів, пов'язаних із освітою та книговидавництвом;
- можливість передати частину книг до бібліотек або на волонтерські ініціативи;
- прозорі звіти про участь у соціальних ініціативах;

— продаж книг на екологічному папері або доставка в мінімальній упаковці [8].

Прозорість та етичність — це не просто тренди, а базові принципи, які формують довіру, підвищують репутацію та забезпечують довготривалий успіх інтернет-магазину книг. Використання чітких політик, дотримання законодавства, відповідальне ставлення до контенту та підтримка соціальних цінностей створюють міцний фундамент для сталого розвитку онлайн-бізнесу.

1.9 Аналіз впливу мобільних технологій на електронну комерцію

Мобільні технології кардинально змінили поведінку споживачів у сфері електронної комерції. Зі зростанням популярності смартфонів і планшетів користувачі все частіше здійснюють покупки, переглядають каталоги товарів, читають відгуки та спілкуються з продавцями саме з мобільних пристроїв. Для інтернет-магазину книг це відкриває як нові можливості, так і виклики, пов'язані з адаптацією інтерфейсу, оптимізацією продуктивності та збереженням якості користувацького досвіду [22].

Згідно з дослідженнями, частка мобільної електронної комерції (m-commerce) у загальному обсязі онлайн-продажів щороку зростає. У 2023 році вона перевищила 60% у світовому масштабі [23]. Це означає, що більше половини потенційних клієнтів інтернет-магазину книг шукають, переглядають і купують товари саме зі смартфонів. Користувачі очікують швидке завантаження сторінок (менше 3 секунд), зручну навігацію, адаптовану верстку, просте оформлення замовлення в кілька кроків і коректне відображення графічних елементів.

Важливо враховувати специфіку поведінки мобільних користувачів: їх сесії зазвичай коротші, а прийняття рішень швидше. Тому інтерфейс має бути інтуїтивним, з великими інтерактивними зонами, простими формами та спрощеною навігацією. Для забезпечення цього використовують такі технології, як респонсивна верстка (responsive design), прогресивні вебдодатки (PWA), які поєднують функціональність вебсайту і мобільного застосунку, а також створення

нативних мобільних застосунків, що дозволяють зберігати історію покупок, надсилати push-сповіщення й працювати офлайн [12].

Окрім інтерфейсної частини, важливою є підтримка зручних платіжних методів. Сучасні мобільні користувачі очікують підтримку Google Pay, Apple Pay, збереження платіжних карток, а також наявність локальних сервісів, таких як LiqPay чи Portmone [16]. Прості та швидкі платіжні рішення значно знижують ймовірність покидання кошика.

З технічної точки зору, мобільна версія сайту має бути оптимізована під mobile-first indexing, який використовує Google. Це означає необхідність мінімізації скриптів і стилів, використання адаптивних зображень у форматах WebP, впровадження Lazy Loading та усунення елементів, що блокують рендеринг [17].

Вимірювання ефективності мобільної версії відбувається за допомогою аналітичних інструментів, зокрема Google Analytics 4. Вони дозволяють аналізувати трафік з мобільних пристроїв, середню тривалість сесій, показники відмов та поведінкові особливості користувачів на різних етапах воронки продажу. Виявлені точки відмови стають підґрунтям для подальшої оптимізації UX [18].

Таким чином, мобільні технології стали невід’ємною частиною сучасної електронної комерції. Вони визначають вимоги до дизайну, функціоналу, швидкості, платіжних рішень і навіть маркетингової стратегії. Для інтернет-магазину книг наявність повноцінної мобільної версії або застосунку є критично важливою умовою забезпечення комфорту користувачів та підвищення конкурентоспроможності ресурсу.

1.10 Створення динамічних вебсторінок JavaScript

JavaScript — динамічна мова програмування, яка використовується для створення інтерфейсу користувача. У даному проєкті вона застосовується разом із бібліотекою React, яка дозволяє створювати інтерактивні компоненти, керувати станом додатку та реалізовувати маршрутизацію між сторінками .

React дозволяє:

- створювати адаптивні та зручні інтерфейси;
- організовувати логіку взаємодії з API через HTTP-запити (наприклад, за допомогою бібліотеки Axios або Fetch API);
- ефективно керувати відображенням даних у компонентах.

Це дає змогу створити зручну адміністративну панель для персоналу книжкового магазину та інтерфейс для клієнтів (якщо потрібно).

JavaScript дозволяє вам створювати інтерактивні елементи на вебсторінках, такі як кнопки, що реагують на натискання, форми, що перевіряють введені дані, анімації та інше. Також використовується для створення серверних додатків, мобільних додатків, ігор та інших типів програмного забезпечення.

Синтаксис JavaScript подібний до інших мов програмування, таких як Java та C, що робить його відносно простим у вивченні для розробників, знайомих з цими мовами. Також вільно типізована мова, що означає, що змінні можуть мати значення будь-якого типу без явного оголошення.

JavaScript підтримує асинхронне програмування за допомогою зворотних викликів, обіцянок та синтаксису `async/await` і дозволяє виконувати неблокуючі операції, що має вирішальне значення для виконання таких завдань, як отримання даних з серверів без заморожування користувацького інтерфейсу.

Об'єктна модель документа (DOM) представляє структуру HTML-документів, а JavaScript дозволяє легко маніпулювати DOM. Дозволяє розробникам динамічно змінювати вміст, структуру і стилі вебсторінок у відповідь на дії користувача або інші події.

JavaScript має багату екосистему фреймворків і бібліотек, які спрощують і прискорюють веброзробку. Деякі популярні з них включають React.js для створення користувацьких інтерфейсів, Angular.js для створення односторінкових додатків та Vue.js для створення інтерактивних вебінтерфейсів.

JavaScript підтримується всіма основними веббраузерами, що робить її крос-платформною мовою. Це означає, що додатки, написані на JavaScript, можуть працювати на різних пристроях і операційних системах без змін.

2 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ

2.1 Огляд технологій для розробки системи

У сучасних умовах стрімкого розвитку цифрових технологій створення якісної інформаційної системи для інтернет-магазину книг вимагає поєднання ефективних інструментів, технологічної гнучкості, стабільності та зручності в обслуговуванні. Одним із ключових принципів побудови ефективної вебсистеми є розподіл на серверну частину (backend) та клієнтську частину (frontend), що відповідає класичній клієнт-серверній архітектурі. Така структура дозволяє чітко розділити відповідальність між частинами: клієнтська сторона відповідає за візуальну взаємодію з користувачем, а серверна — за обробку логіки, запитів, зберігання та захист даних.

У межах проєкту було обґрунтовано вибір архітектури: використання Spring Boot для розробки серверної частини з підтримкою REST API, React (з JSX) для створення інтерактивного інтерфейсу користувача, а також PostgreSQL як надійної системи керування базами даних, що дозволяє зберігати, обробляти та масштабувати товарну інформацію та операції з замовленнями (рисунок 2.1).

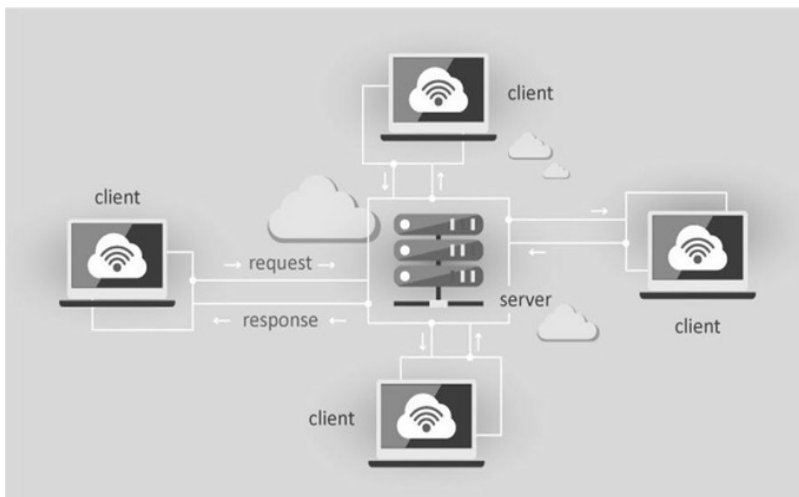


Рисунок 2.1 — Загальна схема взаємодії клієнтської та серверної частин на API

Серверна частина (backend) розроблена з використанням Spring Boot — одного з найпопулярніших Java-фреймворків. Spring Boot дозволяє швидко розгорнути повноцінні вебсервіси, включає підтримку REST API, механізми

безпеки, валідації та логування. Його головна перевага — це зручність конфігурації, автоматизація багатьох процесів і швидкий запуск проєктів із мінімальними налаштуваннями. На відміну від "класичного" Spring, Spring Boot дозволяє розробнику зосередитися саме на логіці бізнес-процесів, а не на довготривалій конфігурації, що особливо важливо для проєктів з обмеженими термінами виконання. Його використання дає змогу створити масштабовану і надійну логіку обробки запитів до бази даних, авторизації користувачів та оформлення замовлень.

Spring Boot — це сучасний Java-фреймворк, що дозволяє швидко створювати масштабовані, продуктивні вебзастосунки з мінімальними налаштуваннями. Він побудований на основі Spring Framework та надає низку інструментів для автоматичної конфігурації, побудови REST API, підключення до баз даних, обробки безпеки та взаємодії з frontend-застосунками.

Spring Boot активно застосовується в розробці інформаційних систем, які потребують надійності, чіткої логіки обробки даних і зручного розгортання.

Основні переваги Spring Boot:

- автоматична конфігурація, адже система «розуміє», що саме потрібно налаштувати на основі підключених залежностей (наприклад, MySQL, Security);
- підтримка REST API, бо розробник може швидко створювати контролери, які взаємодіють із фронтендом або зовнішніми системами;
- інтеграція з базами даних через використання ORM-бібліотеки Hibernate дозволяє ефективно працювати з MySQL через об'єктно-реляційне відображення (Entity-класи);
- вбудований сервер (Tomcat), бо не потрібно окремо запускати сервер — застосунок можна запускати як окремий .jar-файл;
- безпека через використання бібліотеки Spring Security дозволяє реалізувати авторизацію, обмеження доступу, ролі користувачів.

У межах кваліфікаційної роботи Spring Boot реалізує:

- авторизацію користувачів та доступ до функцій системи;
- обробку CRUD-операцій з товарами, категоріями, замовленнями;

- взаємодію з базою даних MySQL;
- передачу даних на фронтенд через REST API.[37]

Фреймворк підтримує архітектуру REST (Representational State Transfer), що дозволяє реалізувати гнучкий API для обміну даними між клієнтом (React) та сервером у форматі JSON. Це забезпечує хорошу масштабованість, особливо якщо в майбутньому передбачено інтеграцію з мобільними застосунками або сторонніми сервісами.

Крім того, Spring Boot має розвинену екосистему, що включає:

- Spring Security для організації реєстрації, входу, розмежування доступів;
- Spring Data JPA для зручної роботи з базою даних (PostgreSQL);
- Spring Validation для перевірки даних з форм користувача;
- Actuator для моніторингу роботи системи;
- Spring DevTools для автоматичного перезапуску під час розробки;
- Logging Frameworks (Logback, SLF4J), які вбудовані й допомагають вести повний журнал подій.

За статистикою Stack Overflow Developer Survey 2023, Spring Boot входить до трійки найбільш використовуваних фреймворків для backend-розробки на Java [21]. За даними ресурсу JetBrains Developer Ecosystem 2023, понад 66% розробників Java використовують саме Spring Boot у своїх проєктах. Більше того, понад 30% корпоративних вебдодатків у світі (особливо в фінансовому та e-commerce секторах) використовують Spring Boot завдяки його надійності, безпеці та широкій спільноті підтримки.

У межах цього дипломного проєкту Spring Boot виконує критичні функції, зокрема:

- обробку запитів від клієнтської частини;
- управління базою даних через ORM;
- захист ендпоінтів від несанкціонованого доступу;
- створення REST API для взаємодії з React-додатком.

Клієнтська частина (frontend) реалізована з використанням React — сучасної JavaScript-бібліотеки, яка дозволяє створювати гнучкі та динамічні інтерфейси з

використанням компонентного підходу. Завдяки цьому користувачі мають змогу зручно переглядати каталог книг, додавати товари до кошика, фільтрувати та сортувати їх, оформлювати замовлення, що значно підвищує загальну юзабіліті системи.

React — це популярна JavaScript-бібліотека для створення інтерфейсів користувача, розроблена компанією Meta (Facebook). React дозволяє будувати динамічні, швидкі та зручні вебінтерфейси на основі компонентної архітектури, де кожен елемент сторінки (форма, таблиця, кнопка) — окремий багаторазовий компонент.

У роботі React використовується для реалізації SPA (Single Page Application) — односторінкового застосунку, який динамічно оновлює контент без перезавантаження сторінки. Це значно покращує зручність роботи для працівників магазину, оскільки усі дії — від додавання книг до створення замовлення — відбуваються плавно й інтерактивно.

React використовує розширення JavaScript під назвою JSX (JavaScript XML) — синтаксис, що дозволяє писати HTML-подібний код безпосередньо у JavaScript. Це забезпечує тісну інтеграцію між логікою та версткою компонентів.

У межах даної роботи React реалізує такі функції:

- відображення списку книг, замовлень і категорій;
- форми створення та редагування товарів;
- валідація полів на стороні клієнта;
- взаємодія з API, отриманим із Spring Boot;
- реактивна зміна стану сторінки без оновлення.

Таким чином, React забезпечує інтуїтивно зрозумілий інтерфейс системи та покращує досвід користувача [3].

Як система керування базами даних (СУБД) обрано PostgreSQL, яка відзначається PostgreSQL — це потужна об'єктно-реляційна система керування базами даних з відкритим вихідним кодом, яка забезпечує високу продуктивність, розширюваність та підтримку складних запитів. Вона є надійною платформою для зберігання структурованої інформації у сучасних інформаційних системах.

У межах даного дипломного проєкту PostgreSQL використовується для збереження:

- даних про книги (назва, автор, ціна, кількість в наявності);
- інформації про категорії товарів;
- замовлень клієнтів;
- даних користувачів (логін, пароль, роль).

Таким чином, побудована інформаційна система базується на чітко структурованій архітектурі з чітким розподілом ролей між клієнтом та сервером, що сприяє гнучкості, масштабованості та легкості супроводу. Використання перевірених технологій дозволило забезпечити швидкодію, надійність, безпеку та зручність використання як для адміністраторів магазину, так і для кінцевих користувачів.

База даних побудована з урахуванням принципів нормалізації, що дозволяє забезпечити цілісність, унеможливити дублювання інформації та оптимізувати зв'язки між таблицями. Зовнішні ключі (FOREIGN KEY) забезпечують логічні залежності між сутностями, наприклад, між книгами й категоріями або між замовленням і користувачем.

Взаємодія з PostgreSQL здійснюється за допомогою фреймворку Spring Boot через ORM-технологію Hibernate. Це дозволяє маніпулювати базою даних на рівні Java-об'єктів без необхідності написання складних SQL-запитів.

Застосування PostgreSQL забезпечує масштабованість, стабільність та гнучкість інформаційної системи, що є важливими критеріями при створенні вебзастосунків для автоматизації бізнес-процесів.

Для кращого розуміння логіки взаємодії між компонентами серверної частини інформаційної системи було побудовано структурну схему архітектури вебдодатку(див. рис. 2.2).

Вона відображає основні модулі backend-частини, взаємозв'язки між ними, а також їх роль у загальній системі. Схема дозволяє наочно продемонструвати, як запити користувача проходять через різні шари — від зовнішнього інтерфейсу до бази даних та у зворотному напрямку.

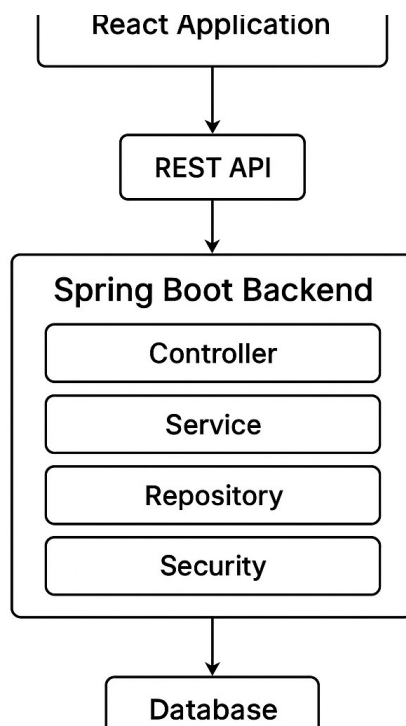


Рисунок 2.2 — Структурна схема backend-архітектури вебдодатку на Spring Boot

2.2 Логічна та фізична структура вебдодатку

Розробка інформаційної системи для інтернет-магазину книг базується на чіткій структурі, яка включає логічне та фізичне представлення архітектури вебзастосунку. Ці аспекти відображають принципи побудови системи з позицій організації коду, структури проєкту, способу взаємодії між компонентами та поділу.

Логічна структура демонструє, як саме компоненти системи пов’язані між собою на рівні функціональності, та визначає ролі окремих частин вебдодатку. Вона дозволяє представити загальну концепцію взаємодії між клієнтською частиною, серверною логікою і базою даних.

У рамках даного проєкту логічна структура базується на класичній клієнт–серверній моделі, де клієнтська частина, реалізована на React, відповідає за візуальне представлення і взаємодію з користувачем, а серверна частина, побудована за допомогою Spring Boot, обробляє запити, реалізує бізнес-логіку і забезпечує доступ до бази даних PostgreSQL.

Клієнтська частина надсилає HTTP-запити до серверної частини через REST

API. У відповідь вона отримує дані у форматі JSON, які відображаються в інтерфейсі. Завдяки цьому можлива гнучка і масштабована побудова як стандартного вебінтерфейсу, так і мобільних застосунків на тій самій базі.

Головна точка входу — файл `index.js`, що підключає кореневий компонент `App.js`, де налаштовуються маршрути (через `react-router`), загальні обгортки (навігація, футер) та глобальні провайдери. У файлі `package.json` міститься перелік залежностей, включно з `React`, `Axios`, `React Router`, `Material UI` тощо.

На стороні серверної частини проєкт побудовано за допомогою `Spring Boot` у структурі `Maven`. Налаштування підключення до бази, портів та конфігурацій API відбувається у файлі `application.properties`.

База даних `PostgreSQL` містить такі основні таблиці: `books`, `users`, `orders`, `order_items`, `authors`. Таблиці зв'язані через зовнішні ключі. Наприклад, таблиця `order_items` має поля `order_id` і `book_id`, які є зовнішніми ключами до таблиць.

Контейнеризація всієї системи здійснюється за допомогою `Docker`. `Dockerfile` для `backend` описує створення Java-застосунку у `JAR`-файлі. Для `frontend` — базується на `Node.js`-образі з використанням `yarn` або `npm`. Компонент `docker-compose.yml` об'єднує всі частини (`frontend`, `backend`, `postgres`) в одну логічну мережу з попередньо заданими змінними середовища (рисунок 2.3).

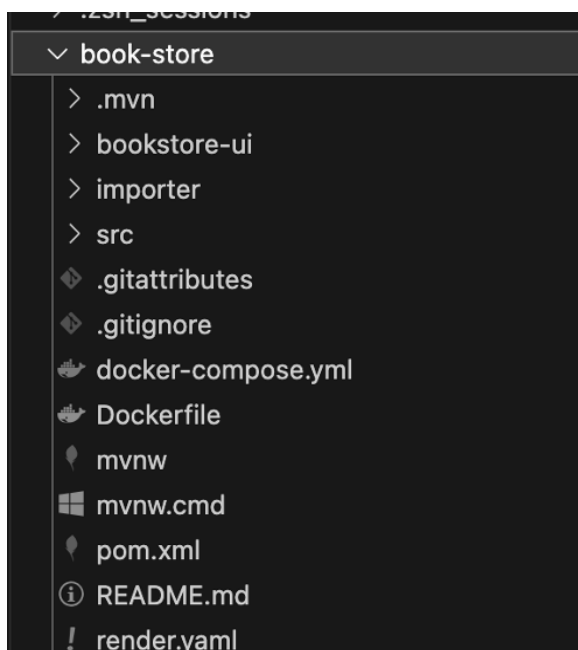


Рисунок 2.3 — Демонстрація фізичної структури вебсайту

3 ПРОЄКТНА ЧАСТИНА

3.1 Обґрунтування та вибір колірної гами вебдодатку

Під час створення вебдодатку для інтернет-магазину книг «Book Shop» ключову увагу було приділено не лише функціональності, але й візуальній складовій — структурі інтерфейсу, вибору кольорів, організації навігації та поданню контенту. У контексті електронної торгівлі книжками дизайн має сприяти орієнтуванню в каталозі, стимулювати до взаємодії з платформою та формувати відчуття довіри до ресурсу.

Важливим чинником при розробці інтерфейсу стала психологія кольору. Для проєкту обрано відтінки синього, які візуально асоціюються зі спокоєм, надійністю та інтелектуальною сферою — саме ці асоціації важливі при продажу літератури. Синій колір також покращує сприйняття інформації на екрані, не перевантажуючи користувача. Білий слугує нейтральним тлом, дозволяючи виділити важливі елементи інтерфейсу — навігацію, кнопки, вміст кошика.

Навігаційна панель (див. рис. 3.1) реалізована у верхній частині інтерфейсу і виконує функцію глобального орієнтиру по сайту. Вона містить логотип магазину, посилання на основні розділи сайту (каталог, авторизація), а також іконку кошика. Розташування та колірне виділення цих елементів відповідає принципам UI/UX-дизайну: максимальна доступність і мінімальне відволікання від вмісту сторінки.

Контент сайту подано у вигляді інтерактивних карток книг, які містять обкладинку, назву, автора та ціну. Усі елементи мають однакову стилізацію, що забезпечує візуальну узгодженість. Реалізація карток виконана за допомогою компонентів React, що дозволяє динамічно підвантажувати книги з бази даних через API та відображати їх відповідно до вибраного фільтра або пошукового запиту.

Окрему увагу приділено читабельності шрифтів — обрано класичні без засічок (наприклад, Roboto), які добре відображаються на різних пристроях. Розміри та кольори шрифтів підібрано з урахуванням контрастності, що покращує доступність для користувачів із вадами зору (рисунки 3.1).

Рисунок 3.1 — Навігаційна панель

Наступний розділ головної сторінки інтернет-магазину Book Shop реалізовано у вигляді слайдера, який демонструє ключові пропозиції магазину — новинки, знижки або тематичні добірки літератури. Для цього використано бібліотеку Swiper, що забезпечує сучасну взаємодію з користувачем завдяки плавному перемиканню між слайдами.

Кожен слайд містить:

- назву акції або категорії книг (заголовок),
- короткий опис (наприклад, «Добірка новинок української прози»),
- візуальне зображення (обкладинка або банер).

Слайдер налаштовано так, щоб автоматично змінювати слайди кожні 5 секунд, забезпечуючи безперервну прокрутку за допомогою параметра `loop: true`. Це привертає увагу користувача та дозволяє відразу ознайомитися з ключовими розділами без потреби прокручувати сторінку вручну (рисунок 3.2).



Рисунок 3.2 — Слайд головної сторінки

Основною функціональною частиною головної сторінки інтернет-магазину Book Shop є каталог книг, який представляє собою сітку карток товарів, що динамічно формуються залежно від результатів фільтрації, сортування чи пошукового запиту.

Кожна картка містить:

- зображення обкладинки книги;

— назву книги (використовується обрізання за довжиною для збереження однорідного вигляду);

— ім'я автора (у поточному прикладі — «Невідомий Автор» у випадку відсутності даних);

— ціну в гривнях.

Каталог реалізовано з використанням компонентів Material UI Grid, що забезпечують адаптивну верстку — картки автоматично перерозподіляються залежно від ширини екрана, що особливо важливо для мобільних пристроїв.

Дані про книги отримуються з backend-частини (Spring Boot) за допомогою REST API. Клієнтська частина на React динамічно оновлює список карток без перезавантаження сторінки, що підвищує швидкодію й зручність взаємодії (рисунок 3.3).

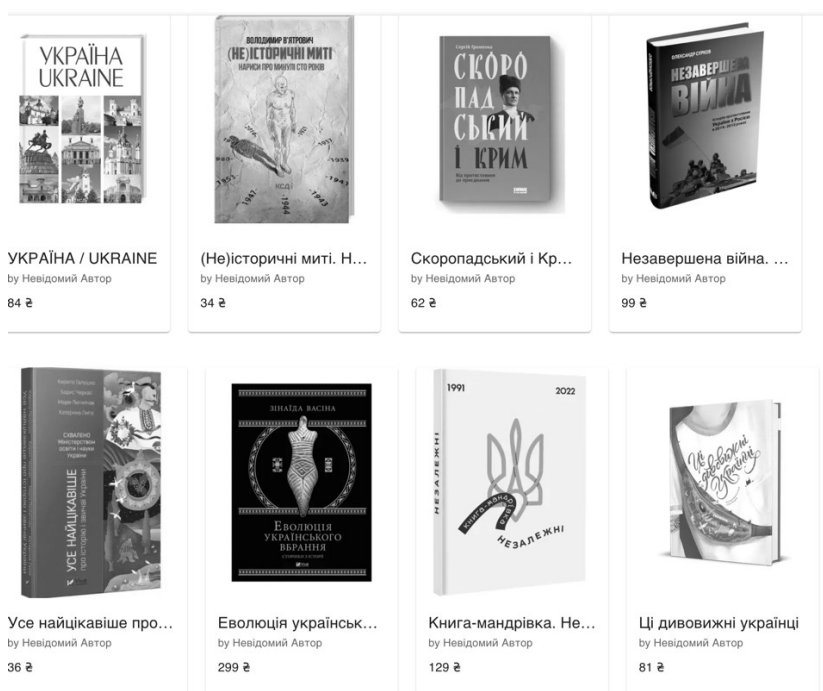


Рисунок 3.3 — Виведення каталогу книг у вигляді карток товарів

3.2 Інтерактивна взаємодія з каталогом

Одним із ключових інструментів, що покращує зручність пошуку в інтернет-магазині Book Shop, є панель фільтрів, яка дозволяє користувачам швидко знаходити необхідні книги за обраними критеріями. Цей інтерфейсний

компонент розташовано в лівій частині сторінки з каталогом і реалізовано за допомогою React-компонентів та стилізовано за допомогою Material UI.

Панель фільтрів складається з двох основних блоків:

— слайдер для ціни — дозволяє користувачу вказати бажаний діапазон вартості книг, рухомі маркери динамічно змінюють межі фільтра, а відповідні запити надсилаються до бекенду, який повертає лише релевантні позиції;

— фільтрація за автором — реалізована у вигляді чекбоксів, користувач може обрати одного або кількох авторів, і каталог автоматично оновиться відповідно до обраного фільтра;

— кнопка «Очистити фільтри» скидає всі вибрані параметри й відображає повний список книг у каталозі, завдяки React-архітектурі цей компонент оновлює стан без перезавантаження сторінки, що підвищує зручність взаємодії (рис. 3.4).

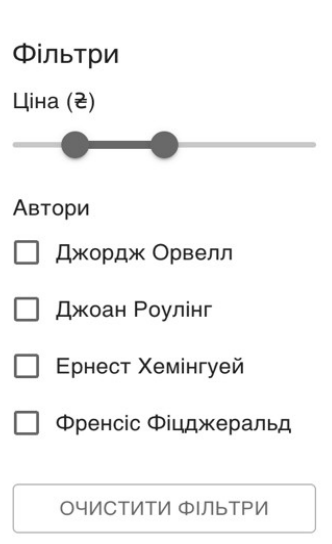


Рисунок 3.4 — Компонент фільтрації книг за ціною та автором у вебдодатку Book Shop

Для підвищення зручності користувача при перегляді каталогу книг у вебдодатку Book Shop реалізовано компонент сортування, що дозволяє впорядкувати товари за визначеними критеріями. Його розміщено над списком книг, поруч із фільтрами.

Інтерфейс компонента побудовано у вигляді випадаючого списку (Select) з двома варіантами:

- Title — сортування за алфавітом за назвою книги;
- Price — сортування за ціною (від дешевших до дорожчих або навпаки).

При виборі користувачем певного параметра, такого як категорія, автор, ціна або рейтинг, відповідний запит надсилається на сервер або обробляється безпосередньо на клієнтській стороні залежно від логіки реалізації застосунку. У разі надсилання запиту на сервер відбувається звернення до API, яке повертає оновлений список книг відповідно до вказаних параметрів. Якщо ж фільтрація реалізована на клієнті, то дані обробляються локально за допомогою JavaScript, що дозволяє зменшити навантаження на сервер і забезпечити швидшу реакцію інтерфейсу.

Оновлений список миттєво відображається на сторінці, і користувач бачить лише ті позиції, які відповідають обраним критеріям. Наприклад, якщо обрано фільтр "за зростанням ціни", книги сортуються відповідним чином, надаючи змогу зручно порівняти варіанти за бюджетом. Подібна інтерактивність підвищує якість взаємодії користувача із сайтом, оскільки дає змогу легко знаходити потрібну книгу серед великої кількості доступних товарів. Такий підхід до реалізації динамічного оновлення даних є невід'ємною частиною сучасних інтерфейсів електронної комерції, орієнтованих на зручність та ефективність користувацького досвіду.

Завдяки реалізації на React, сортування відбувається без перезавантаження сторінки, а зміна критерію миттєво впливає на виведені результати.

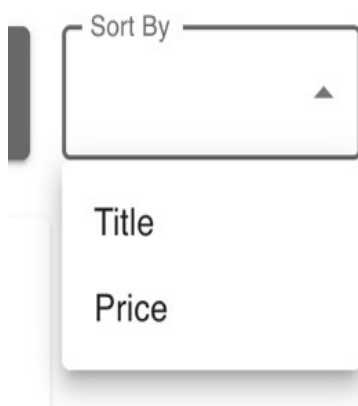


Рисунок 3.4 — Компонент сортування книг у каталозі

3.3 Динамічні компоненти клієнтської частини вебдодатку

У вебдодатку інтернет-магазину книг реалізовано декілька динамічних компонентів, які сприяють покращенню зручності користувачів, підвищенню інтерактивності та ефективності взаємодії з інтерфейсом. Основна логіка роботи цих компонентів реалізована на клієнтській частині за допомогою сучасної бібліотеки React, яка забезпечує швидке оновлення інтерфейсу без перезавантаження сторінки. Для створення адаптивного, привабливого та функціонального дизайну використано бібліотеку компонентів Material UI, а для реалізації слайдерів і каруселей – бібліотеку Swiper, що дозволяє зручно відображати колекції книжок, новинки чи акційні пропозиції. Такий підхід значно покращує загальне враження від використання вебдодатку та підвищує рівень задоволеності користувачів. (див. лістинг 3.1).

Лістинг 3.1 — Ініціалізація автоматичного слайдера з книгами (Swiper)

```
const quantitySelectButtons = document.querySelectorAll
const swiper = new Swiper('.book-slider', {
  loop: true,
  autoplay: {
    delay: 5000,
    disableOnInteraction: false, },
  pagination: {
    el: '.swiper-pagination',
    clickable: true,
  },
});
```

Слайдер використовується для показу новинок або рекомендованих книг на головній сторінці. Він автоматично прокручує слайди кожні 5 секунд і дозволяє зручно перемикатися між акційними або тематичними добірками, покращуючи візуальну привабливість ресурсу (лістинг 3.2).

Лістинг 3.2 — Реалізація кнопки повернення до початку сторінки

```
const toTopButton = document.querySelector('.to-top');
toTopButton.addEventListener('click', function () {
  window.scrollTo({
    top: 0,
    behavior: 'smooth'
  });
});
```

Функціонал плавного повернення до початку сторінки покращує UX, особливо при довгих списках книг. Реалізовано через JavaScript із використанням методу `scrollTo()` з параметром `behavior: 'smooth'`.

Функція очищає обрані чекбокси авторів та скидає ціновий фільтр до початкових значень. Вона прив'язана до кнопки «Очистити фільтри» в React-компоненті. Дозволяє миттєво оновити список товарів відповідно до нових параметрів без перезавантаження сторінки. (див. лістинг 3.3).

Лістинг 3.3 — Очищення фільтрів

```
document.addEventListener("DOMContentLoaded", function() {  
  const resetFilters = () => {  
    setSelectedAuthors([]);  
    setPriceRange([minPrice, maxPrice]);  
  };  
});
```

Компонент сортування на сторінці каталогу дозволяє впорядковувати книги за назвою або ціною. Вибране значення зберігається у стані React-компонента та передається до запиту, що повертає відсортований список товарів (лістинг 3.4).

Лістинг 3.4 — Зміна сортування книг

```
document.addEventListener("DOMContentLoaded", function() {  
  const resetFilters = () => {  
    setSelectedAuthors([]);  
    setPriceRange([minPrice, maxPrice]);  
  };  
});
```

3.4 Реалізація адмін-панелі та CRUD-функцій

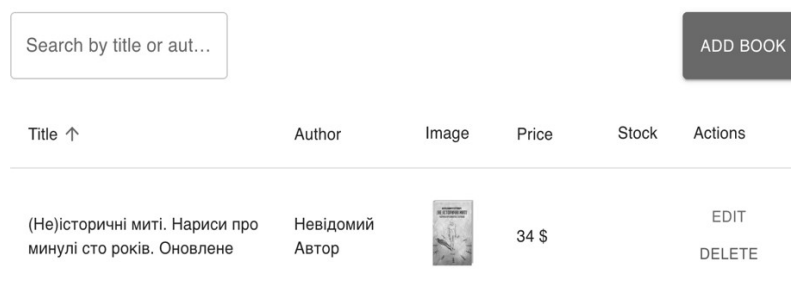
У даній роботі для інформаційної системи інтернет-магазину книг було реалізовано повнофункціональну адміністративну панель, яка забезпечує зручне управління товарами (книгами), категоріями, замовленнями та користувачами. Адмінпанель реалізована з використанням сучасного інтерфейсу, що забезпечує інтуїтивно зрозумілу навігацію та оперативну взаємодію з даними. Вхід до панелі адміністратора доступний лише авторизованим користувачам з відповідною роллю ADMIN, що дозволяє забезпечити високий рівень безпеки та контроль доступу до критично важливих функцій системи. Це, зокрема, включає

можливість додавання, редагування й видалення товарів, перегляду історії замовлень, зміни статусів доставок та керування категоріями, що робить адміністрування сайту максимально ефективним і зручним.

Після входу до адмін-панелі адміністратор отримує доступ до таких функцій:

- перегляд списку всіх книг у базі;
- створення нових книг через форму із валідацією;
- редагування назв, авторів, цін, категорій та кількості товарів;
- видалення книг, які більше не продаються;
- керування категоріями (додавання, редагування, видалення);
- перегляд замовлень користувачів (на перспективу — з можливістю обробки).

Інтерфейс реалізовано з використанням React та Material UI. Кожна дія викликає відповідний запит до backend API, який розроблено на базі Spring Boot (рисунок 3.5).



The screenshot shows an admin interface with a search bar at the top left labeled 'Search by title or aut...', an 'ADD BOOK' button at the top right, and a table below. The table has columns: Title ↑, Author, Image, Price, Stock, and Actions. One book is listed with the title '(Не)історичні миті. Нариси про минулі сто років. Оновлене', author 'Невідомий Автор', a book cover image, price '34 \$', and actions 'EDIT' and 'DELETE'.

Title ↑	Author	Image	Price	Stock	Actions
(Не)історичні миті. Нариси про минулі сто років. Оновлене	Невідомий Автор		34 \$		EDIT DELETE

Рисунок 3.5 — Admin-panel з CRUD-операціями для книг

Доступ до адміністративної панелі захищено через механізм JWT-автентифікації. Після успішного входу користувача, токен зберігається у localStorage або sessionStorage на клієнтській стороні та передається в кожному запиті до API у заголовок «Authorization: Bearer <JWT>»

На стороні сервера налаштовано Spring Security, який обмежує доступ до маршрутів /api/books/** лише для користувачів із роллю ADMIN. Фрагмент конфігурації безпеки наведено на лістингу 3.5.

Лістинг 3.5 — Налаштування доступу в Spring Security

```

@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .csrf().disable()
        .authorizeRequests()
        .antMatchers("/api/books/**").hasRole("ADMIN")
        .anyRequest().permitAll()
        .and()
        .sessionManagement()
        .sessionCreationPolicy(SessionCreationPolicy.STATELESS);
}

```

Для реалізації операцій над книгами використано клас `BookController`, що відповідає за обробку запитів від клієнта. Він містить усі чотири основні операції — Create, Read, Update, Delete (CRUD), які є критично важливими для забезпечення повноцінного функціонування інформаційної системи. CRUD-операції лежать в основі будь-якої системи управління даними, оскільки дозволяють не лише створювати нові записи в базі даних (додавання книг), а й забезпечують можливість їх перегляду, редагування або видалення при зміні асортименту, оновленні цін або видаленні неактуальних товарів. Особливо важливо це для адміністратора, який завдяки цим функціям може підтримувати актуальність контенту магазину, оперативно реагувати на зміни та ефективно управляти каталогом. Наявність CRUD-функціоналу також спрощує масштабування системи та подальше доповнення новими сутностями (наприклад, категоріями, авторами, рецензіями), що робить систему гнучкою та придатною для розширення (лістинг 3.6).

Лістинг 3.6 — Налаштування доступу в Spring Security

```

@RestController
@RequestMapping("/api/books")
@RequiredArgsConstructor
public class BookController {
    private final BookService bookService;

    @GetMapping
    public List<BookDto> getAllBooks() {
        return bookService.getAllBooks();
    }

    @PostMapping
    public BookDto createBook(@RequestBody BookDto bookDto) {

```

```

        return bookService.createBook(bookDto);
    }
    @PostMapping("/{id}")
    public BookDto updateBook(@PathVariable Long id, @RequestBody BookDto
bookDto) {
        return bookService.updateBook(id, bookDto);
    }
    @DeleteMapping("/{id}")
    public void deleteBook(@PathVariable Long id) {
        bookService.deleteBook(id);
    }
}

```

Точки доступу до API:

- GET /api/books — отримання списку всіх книг;
- POST /api/books — додавання нової книги;
- PUT /api/books/{id} — редагування книги;
- DELETE /api/books/{id} — видалення книги.

Таке рішення забезпечує централізовану взаємодію клієнтської частини з базою даних і дозволяє ефективно керувати інформацією в системі.

Реалізація CRUD-функцій (frontend)

На клієнтській стороні вебзастосунку реалізовано кілька ключових компонентів, що відповідають за управління даними книг у адміністративній частині інтерфейсу. Зокрема, AdminBooksPage, BookForm та CategoryForm є окремими складовими, кожна з яких виконує свою роль у взаємодії адміністратора з даними. Компонент AdminBooksPage відповідає за відображення загального списку книг у вигляді таблиці або карток. У ньому реалізовано можливість пошуку, сортування, а також доступ до функцій редагування та видалення кожного окремого запису. Користувач може переглянути наявні книги, ознайомитися з їхніми характеристиками, і, при потребі, швидко перейти до редагування.

Компонент BookForm виконує функцію форми для створення або редагування книги. У ньому реалізовано всі необхідні поля — назва, опис, автор, ціна, категорія, зображення тощо. Залежно від контексту, компонент працює в режимі створення нового запису або редагування вже наявного. Після заповнення

форми адміністратор може надіслати дані, які автоматично обробляються та передаються на сервер через API.

Аналогічним чином компонент `CategoryForm` відповідає за створення та редагування категорій книг. Це важлива частина функціоналу, оскільки категорії дозволяють структурувати каталог і забезпечити зручну навігацію для користувачів. Форма передбачає введення назви категорії, а також можливе оновлення її параметрів.

Для взаємодії з серверною частиною у всіх цих компонентах використовується популярна JavaScript-бібліотека `Axios`, яка дозволяє зручно й ефективно надсилати HTTP-запити. `Axios` забезпечує обробку таких операцій, як отримання списків, створення нових записів, оновлення існуючих і їх видалення. Запити надсилаються до відповідних маршрутів API, а отримані відповіді автоматично оновлюють стан компонентів, що забезпечує інтерактивну й динамічну взаємодію адміністратора з даними у реальному часі. Завдяки цьому структура адміністративного інтерфейсу є гнучкою, розширюваною та зручною для подальшого розвитку.

Стан компонента в додатку зберігається за допомогою хука `useState`, що дозволяє керувати динамічними даними інтерфейсу, наприклад, зберігати значення полів форми, обрані фільтри або поточний список книг. Початкове значення стану задається при ініціалізації компонента, і при кожній зміні цього стану `React` автоматично оновлює інтерфейс, забезпечуючи реактивність додатку. Завантаження зовнішніх даних, таких як список книг або категорій, виконується за допомогою хука `useEffect`, який дозволяє запускати побічні ефекти після першого рендеру компонента або при зміні певних залежностей. Це забезпечує завантаження актуальної інформації з сервера без потреби в ручному оновленні сторінки.

Передача даних до `backend` відбувається у форматі `JSON`, що є стандартом для обміну даними між клієнтом і сервером у сучасних вебзастосунках. При відправленні форми або збереженні нової книги об'єкт з даними серіалізується у формат `JSON` та передається через HTTP-запит, найчастіше методом `POST` або

PUT, залежно від дії. Сервер, у свою чергу, приймає ці дані, розпізнає їх і зберігає у відповідній базі даних. Така взаємодія забезпечує ефективну і зрозумілу комунікацію між фронтендом і бекендом, підтримуючи структуру SPA (single-page application), де оновлення даних відбувається без повного перезавантаження сторінки.

Переваги реалізованої адмін-панелі:

- спрощення процесу керування асортиментом;
- розмежування ролей користувачів і контроль доступу;
- можливість розширення (наприклад, імпорт книг через CSV, керування знижками);
- наочність і простота використання для менеджерів або власників магазину;
- зручність редагування навіть великих каталогів.

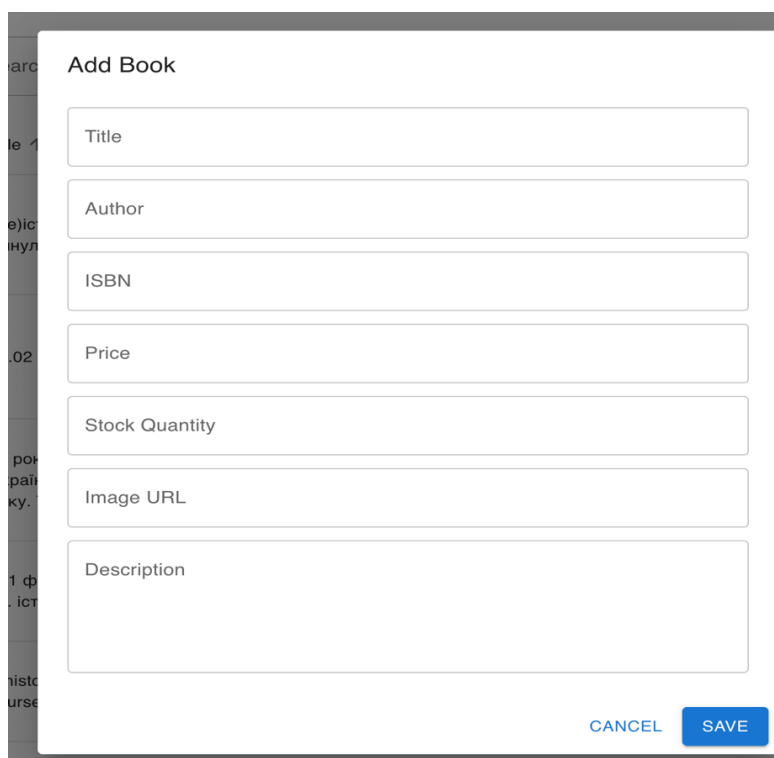
The image shows a web form titled "Add Book" within an administrative interface. The form is a vertical stack of input fields: "Title", "Author", "ISBN", "Price", "Stock Quantity", "Image URL", and a larger "Description" field. At the bottom right of the form, there are two buttons: a light blue "CANCEL" button and a dark blue "SAVE" button. The form is overlaid on a blurred background of the admin panel.

Рисунок 3.6 — Форма додавання книги у адміністративній панелі

Таким чином, реалізація адміністративної панелі та CRUD-функцій дозволила створити зручний інструмент для внутрішньої роботи з даними в системі. Адміністратори отримали можливість швидко додавати нові книги,

редагувати наявну інформацію, змінювати категорії, видаляти застарілі записи та керувати асортиментом без потреби втручання в код. Усе це реалізовано через інтуїтивно зрозумілий інтерфейс, що працює на основі компонентного підходу React та API-запитів до серверної частини. Такий підхід забезпечує динамічність, масштабованість і легкість у супроводі додатку. У результаті, ефективність управління книжковим інтернет-магазином значно зростає, а також зменшується час, необхідний на внесення змін. Крім того, гнучкість та модульність реалізованої системи сприяє її подальшому розвитку — легко додавати нові функції, інтегрувати сторонні сервіси або змінювати логіку роботи згідно з потребами бізнесу.

4 ТЕСТУВАННЯ ПРАЦЕЗДАТНОСТІ РОЗРОБЛЕНОЇ ПЛАТФОРМИ

4.1 Оптимізація відображення слайдера

Кожен слайд у головному блоці інтернет-магазину книг реалізовано з використанням бібліотеки Swiper, яка забезпечує зручну і динамічну зміну слайдів. Усі слайди розміщені на головній сторінці та мають однакову структуру: фонове зображення, заголовок рекламного характеру, підзаголовок з коротким поясненням і навігаційну кнопку «До каталогу». Зміна слайдів відбувається автоматично завдяки параметру autoplay, що задає інтервал у 5 секунд між переходами. Крім того, реалізовано елементи ручного керування — кнопки зі стрілками ліворуч та праворуч, які дозволяють самостійно переключати слайди (рис. 4.1—4.3).

Використання Swiper дає змогу реалізувати циклічну прокрутку слайдів (через параметр loop: true), що створює ефект безперервного перегляду, підвищуючи динамічність інтерфейсу. Такий підхід дозволяє поєднати інформативність із візуальною привабливістю та сприяє зручному ознайомленню користувача з актуальними акціями або категоріями продукції.



Рисунок 4.1 — Початкова сторінка слайдера на головній сторінці



Рисунок 4.2 — Наступна сторінка слайдера на головній сторінці

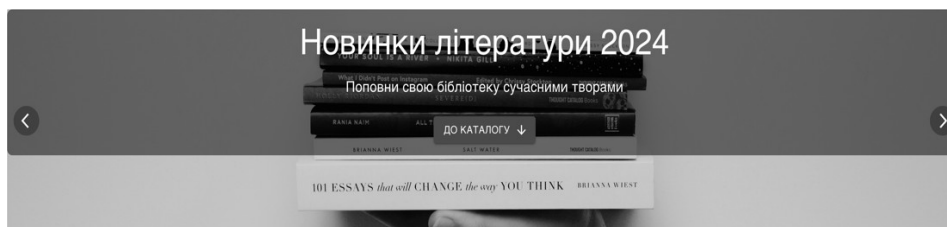


Рисунок 4.3 — Наступна сторінка слайдера на сторінці каталогу

Коли користувач натискає на кнопку, виконується дія, яка модифікує стиль для показу відповідного списку продуктів і активує вибрану кнопку. Код слідкує за натисканнями на кнопки і, відповідно до атрибутів даних, показує відповідні списки продуктів (рисунок 4.4).

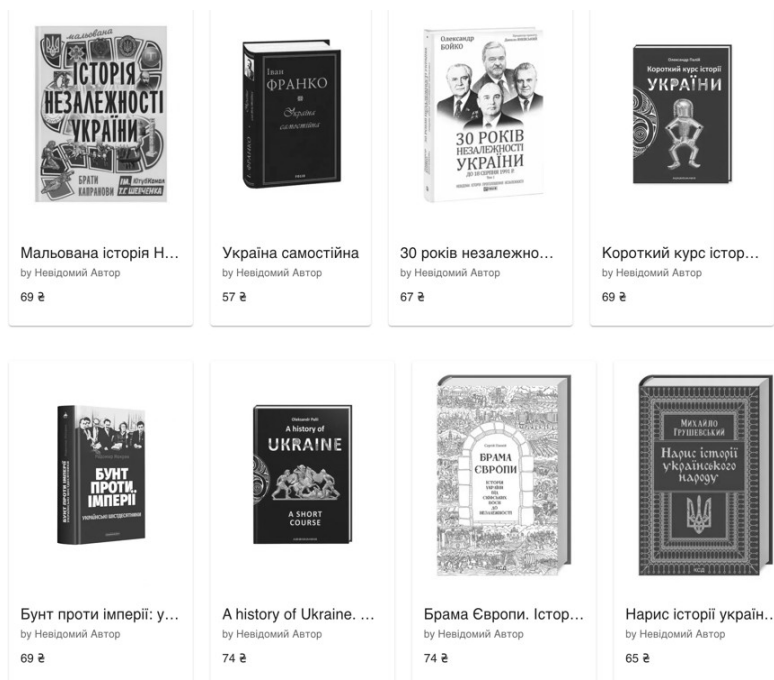


Рисунок 4.4 — Сторінка каталогу товарів

4.2 Додавання товарів в кошик

Функціональність додавання книги до кошика є однією з ключових частин системи електронної комерції, яка забезпечує зручність взаємодії користувача з інтерфейсом інтернет-магазину та спрощує процес покупки. Ця функція дозволяє покупцям формувати замовлення без потреби одразу його оформлювати, забезпечуючи таким чином гнучкість і зручність під час перегляду каталогу.

У вебзастосунку інтернет-магазину книг реалізовано функціональність додавання товару до кошика, яка є однією з основних складових будь-якої системи електронної комерції. Ця функціональність дозволяє користувачу обрати певну книгу та додати її до кошика, щоб згодом оформити замовлення.

На сторінці каталогу під кожною карткою книги розміщено кнопку «ДОДАТИ В КОШИК» (рис. 4.5). При її натисканні користувач бачить повідомлення що товар додано в кошик, а також змінюється вигляд кнопки кошика. При цьому відбувається виклик функції, яка передає ідентифікатор книги у локальне сховище або контекстний стан програми, залежно від реалізації. Стан кошика миттєво оновлюється, і відповідний товар з'являється на сторінці кошика.

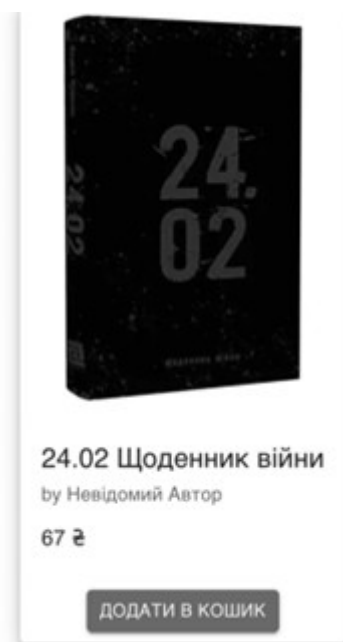


Рисунок 4.5 — Кнопка додавання книги в кошик

Після додавання товару до кошика користувач отримує можливість перейти на сторінку /cart, яка є окремим розділом інтернет-магазину й виконує роль віртуального кошика. На цій сторінці відображається повна інформація про всі вибрані книги, які були додані під час перегляду каталогу (рисунки 4.6).

Для кожної книги показується її назва, зображення, ціна за одиницю та кількість, яку можна змінити без потреби видаляти товар. Інтерфейс реалізовано таким чином, щоб користувач міг зручно змінювати кількість обраних книг. Також поруч із кожною позицією є кнопка видалення, що дозволяє швидко прибрати товар із кошика, якщо користувач передумав його купувати.

У нижній частині сторінки відображається автоматично розрахована загальна сума замовлення, яка оновлюється щоразу при зміні кількості товарів або після їх видалення. Це дає змогу користувачу в реальному часі бачити, скільки коштуватиме його покупка. Додатково на сторінці розміщено дві основні кнопки керування: «CHECKOUT» і «CLEAR CART». Кнопка «CHECKOUT» перенаправляє користувача до наступного етапу — оформлення замовлення, де потрібно буде ввести особисті дані, адресу доставки та обрати спосіб оплати. Кнопка «CLEAR CART» призначена для повного очищення кошика — після її натискання всі обрані книги видаляються зі списку, і кошик повертається до початкового порожнього стану.

Ця сторінка відіграє важливу роль у взаємодії користувача з магазином, оскільки забезпечує останній етап перед переходом до купівлі товару. Вона повинна бути максимально зручною, інформативною й адаптивною до змін, для того щоб користувач міг швидко внести корективи до свого вибору, не втрачаючи вже доданих товарів або не повторюючи попередні дії, не витрачаючи на це багато часу. Зокрема, можливість оперативно змінити кількість екземплярів книги або повністю видалити її з кошика дозволяє гнучко керувати покупкою без необхідності починати процес заново. Усі зміни миттєво відображаються у підсумковій сумі замовлення, що допомагає контролювати витрати. Інтерфейс кошика також повинен адаптуватися до різних розмірів екранів, що забезпечує комфортне використання як на комп'ютерах, так і на мобільних пристроях. Добре продумана логіка кошика значно підвищує рівень задоволення користувача, створюючи відчуття зручності та контролю, що, у свою чергу, сприяє завершенню покупки та підвищує конверсію інтернет-магазину. Таке рішення забезпечує швидкий відгук системи, зручність для користувача та легкість подальшого

оформлення покупки, що є ключовими критеріями ефективного інтерфейсу інтернет-магазину.

Крім того, важливим елементом сторінки кошика є її інтеграція з іншими частинами системи, зокрема — з модулем авторизації користувачів та оформлення замовлення. Якщо користувач ще не авторизований, система може запропонувати увійти або зареєструватися перед завершенням покупки, що дозволяє зберігати історію замовлень та пришвидшує процес заповнення даних під час оформлення. Також інтеграція з системами оплати й доставки забезпечує можливість попереднього вибору бажаного способу отримання товару та розрахунку. Завдяки цьому сторінка кошика перетворюється з простої функціональної частини на повноцінний центр керування замовленням, який поєднує в собі зручність, функціональність та логічну послідовність дій користувача.

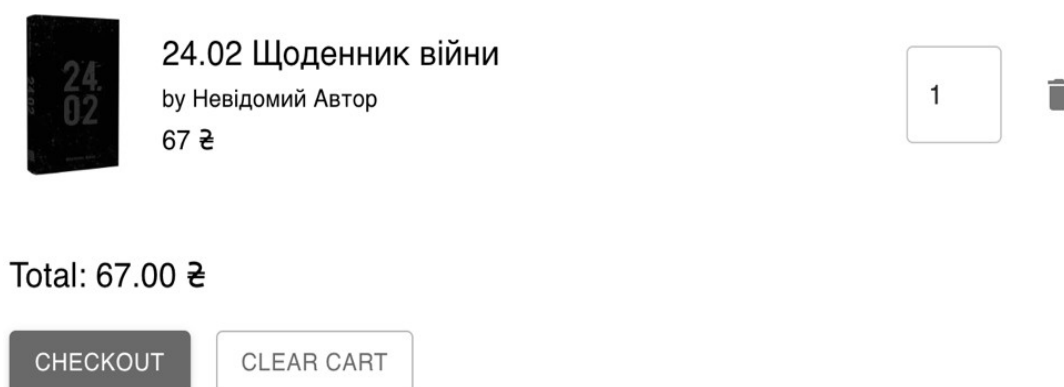


Рисунок 4.6 — Кошик із вибраним товаром

Уся взаємодія реалізована за допомогою JavaScript (React), з використанням компонентів, які відповідають за відображення, обчислення вартості та інтеграцію з backend-частиною на Spring Boot. Компоненти організовані таким чином, щоб забезпечити логічну побудову інтерфейсу, а також мінімізувати кількість дублювання коду за рахунок повторного використання спільних елементів. Операції над кошиком можуть зберігатись тимчасово у localStorage, що дозволяє зберегти дані навіть при оновленні сторінки або випадковому її закритті. У випадку авторизованого користувача дані синхронізуються із сервером, де

формуються попередні замовлення або зберігається історія покупок. Це дозволяє реалізувати більш гнучку систему обліку, персоналізації та аналітики.

Таке рішення забезпечує швидкий відгук системи, зручність для користувача та легкість подальшого оформлення покупки, що є ключовими критеріями ефективного інтерфейсу інтернет-магазину. Висока швидкість взаємодії між клієнтською та серверною частинами дозволяє мінімізувати час очікування, що особливо важливо для сучасних користувачів, які цінують швидкість і комфорт. Інтуїтивний інтерфейс допомагає покупцеві легко орієнтуватися у процесі вибору та придбання товарів, навіть без попереднього досвіду користування подібними платформами. Крім того, безперебійна інтеграція з backend-системою гарантує надійне збереження інформації про замовлення, що важливо як для користувача, так і для адміністратора магазину. Загалом, така архітектура не лише підвищує задоволеність клієнтів, а й сприяє зростанню рівня довіри до онлайн-сервісу, що в перспективі позитивно впливає на лояльність аудиторії та фінансові показники бізнесу.

ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи було створено інформаційну систему для інтернет-магазину книг, яка відповідає сучасним вимогам до електронної комерції та забезпечує високий рівень зручності для користувачів. Систему розроблено з використанням сучасного стеку технологій — Spring Boot для серверної частини, React для клієнтської частини, PostgreSQL для зберігання даних, а також засобів контейнеризації Docker для гнучкого розгортання.

На першому етапі було проведено аналіз предметної області, розглянуто сучасні приклади інформаційних систем для онлайн-магазинів, зокрема книжкових. Це дозволило виділити ключові функціональні компоненти, серед яких: каталог продукції, пошук, фільтрація, сортування, обробка замовлень, а також елементи адміністрування.

На основі вивчених вимог було сформовано логічну та фізичну структуру системи, спроектовано зручний інтерфейс із дотриманням принципів UX/UI дизайну. Вибір кольорової гами, структура навігації, реалізація слайдерів, фільтрів і сортування сприяли створенню зручної, зрозумілої та адаптивної системи для користувачів.

Особливу увагу приділено динамічним елементам: реалізовано автоматичний слайдер, приховані фільтри, сортування без перезавантаження, плавне повернення до верху сторінки. Усе це значно покращує взаємодію з інтерфейсом.

Проведено дослідження впливу мобільних технологій на електронну комерцію та впроваджено адаптивну верстку для коректної роботи на різних пристроях. Також проаналізовано аспекти безпеки транзакцій і збереження користувацьких даних.

Проведено тестування основних функціональних модулів, зокрема роботи з каталогом, обробки фільтрів і сортування. Усі тести показали стабільність, коректність та відповідність функціональності поставленим вимогам.

Результати роботи доводять, що розроблена система є ефективною,

зручною, масштабованою та готовою до впровадження. Вона може слугувати основою для подальшого розвитку інтернет-магазинів книг або подібних проєктів.

Таким чином, поставлені в бакалаврській кваліфікаційній роботі завдання були виконані повністю, а мета досягнута. Розроблена інформаційна система відповідає актуальним технічним та функціональним стандартам, забезпечує зручність для кінцевого користувача та містить потенціал для інтеграції з іншими службами й сервісами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дьяків Ю. Інформаційні системи в електронній комерції. — 2020.
2. Офіційна документація WooCommerce: Режим доступу: <https://woocommerce.com> (дата звернення: 18.04.2022). — Назва з екрана.
3. Magento eCommerce Platform [Електронний ресурс] // MDN Web Docs. — Режим доступу <https://magento.com> (дата звернення: 19.04.2025). — Назва з екрана.
4. WooCommerce Documentation — <https://woocommerce.com/document/> (дата звернення: 19.04.2025). — Назва з екрана. Pavlyuk M.O. Web Performance Optimization. — 2020.
5. Magento Internationalization Guide — <https://developer.adobe.com/commerce> (дата звернення: 19.04.2025). — Назва з екрана.
6. Coronel C., Morris S. Database Systems: Design, Implementation, & Management. — 2022.
7. Garrett J.J. The Elements of User Experience. — 2010.
8. Колірна гамма сайту: рекомендації по підбору [Електронний ресурс] // PBB Design. — Режим доступу: <https://pbb.lviv.ua/statti-i-novyny/statti-shchodostvorennia-saitu/kolirna-hamma-saytu/> (дата звернення: 20.04.2025). — Назва з екрана.
9. Nielsen Norman Group. UX Design Guidelines. — <https://www.nngroup.com/>
10. Fishkin R. The Art of SEO. — 2020.
11. Moz Beginner's Guide to SEO — <https://moz.com/beginners-guide-to-seo>
12. Google Search Central. SEO Starter Guide — <https://developers.google.com/search/docs/fundamentals/seo-starter-guide>.
13. Semrush Academy. Analytics for SEO and PPC. — <https://www.semrush.com/academy/>
14. ISO 10002:2018: Quality management — Guidelines for complaints handling in organizations.

15. Закон України «Про захист персональних даних» –
<https://zakon.rada.gov.ua/laws/show/2297-17>
16. WIPO. Intellectual Property and Copyright in Publishing –
<https://www.wipo.int/copyright/en/>
17. Google Business Reviews Policy –
<https://support.google.com/business/answer/4596773?hl=uk>
18. Shopify. Guide to Ethical eCommerce
<https://www.shopify.com/blog/ethical-e-commerce>
19. Nielsen Norman Group. Mobile User Experience Guidelines –
<https://www.nngroup.com/>
20. Google Developers. Progressive Web Apps Guide –
<https://web.dev/progressive-web-apps/>
21. Google Search Central. Mobile-First Indexing. – Режим доступу:
<https://developers.google.com/search/mobile-sites/>
22. GA4 Documentation. Mobile Analytics Reporting –
<https://support.google.com/analytics>
23. Spring Boot Documentation [Електронний ресурс]. – Режим доступу:
<https://spring.io/projects/spring-boot> (дата звернення: 30.04.2025).
24. React Documentation [Електронний ресурс]. – Режим доступу:
<https://react.dev/> (дата звернення: 30.04.2025).
25. Калюжний А.М. Основи інформаційних технологій. — Львів: Новий Світ, 2021.
26. Laudon K.C., Laudon J.P. Management Information Systems: Managing the Digital Firm. — Pearson, 2022.
27. Molich R., Nielsen J. Improving a Human-Computer Dialogue. — Communications of the ACM, 1990.
28. Інформаційна система для обліку товарів у книжковому магазині [Електронний ресурс] / Кисюк Д. В., Єрмакова А. М. // Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)», Вінниця, 16 червня 2025р.

ДОДАТОК А**Технічне завдання**

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ
проф., д.т.н.. Азаров О.Д.
21 березня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи
«Інформаційна система для обліку товарів у книжковому
магазині»

08-54.БКР.027.00.000 ТЗ

Науковий керівник: ст. викл. каф. ОТ
_____Кисюк Д. В.

Студентка групи КІ-23мсз
_____Єрмакова А. М.

1 Підстава для використання бакалаврської кваліфікаційної роботи (БКР)

1.1 Підстава розробки інформаційних систем для інтернет-магазинів є для сучасного світі, де все більше та більше покупок здійснюється онлайн, ефективність та надійність інтернет-магазинів стає критично важливою. Інформаційні системи допомагають інтернет-магазинам управляти своїми продуктами, замовленнями, взаємодією з клієнтами та іншими аспектами бізнесу. Вони допомагають підвищити продуктивність, поліпшити обслуговування клієнтів та збільшити продажі.

1.2 Наказ про затвердження теми бакалаврської кваліфікаційної роботи від 21 березня 2025 р. №97.

2 Мета БКР і призначення розробки

2.1 Мета роботи – полягає в розробці та реалізації інформаційної системи для книжкового інтернет-магазину. Дана система буде спрямована на оптимізацію процесів управління та підвищення ефективності бізнесу, забезпечуючи зручну покупку для клієнтів та швидку обробку їх замовлень.

2.2 Призначенням даної розробки є створення ефективної та зручної інформаційної системи для інтернет-магазину, спеціалізованого на продажу книг. Реалізація цієї інформаційної системи дозволить значно підвищити ефективність роботи і книжкового інтернет-магазину, забезпечить зручність для клієнтів та сприятиме зростанню продажів і лояльності покупців.

3 Вихідні дані для виконання БКР:

3.1 Вивчення існуючих інформаційних систем, їхніх переваг та недоліків.

3.2 Створення моделі для представлення різних аспектів інформаційної системи.

3.3 Розробка прототипу інформаційної системи.

3.4 Тестування і оцінка прототипу системи.

4 Вимоги до виконання БКР Головна вимога – використати, як основний, метод створення моделі для представлення різних аспектів інформаційної системи, розробка прототипу інформаційної системи.

5 Етапи БКР та очікувані результати приведено в Таблиці А.

Таблиця А.1 — Етапи БКР

№ з/п	Назва етапу	Термін виконання	Очікувані результати
1	Постановка задачі роботи	21.03.25	Розділ 1
2	Пошук матеріалів та інформаційних джерел	21.03.25— 30.03.25	Розділ 1
3	Аналіз існуючих прототипів та визначення вимог до створюваної системи	21.03.25— 30.03.25	Розділ 2
4	Обґрунтування структури та принципів роботи системи	31.03.25— 13.04.25	Розділ 2
5	Розробка й тестування апаратно-програмної системи	14.04.25— 27.04.25	Розділ 3
6	Оформлення пояснювальної записки та демонстраційної презентації	28.04.25— 12.05.25	ПЗ, презентація
7	Підготовка супровідної документації та перевірка на плагіат	13.05.25	ПЗ, презентація

6 Матеріали, що подаються до захисту БКР. До захисту подаються: пояснювальна записка БКР, ілюстративні матеріали, протокол попереднього захисту БКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до БКП українською та іноземною мовами, довідка про відповідність оформлення БКП діючим вимогам.

7 Порядок контролю виконання та захисту БКР. Виконання етапів графічної та розрахункової документації БКР контролюється науковим керівником згідно зі встановленими термінами. Захист БКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання БКР

8.1 При оформлювання БКР використовуються:

- ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;
- ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;
- методичні вказівки до виконання бакалаврських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія. Кафедра обчислювальної техніки ВНТУ 2023.

8.2 Порядок виконання БКР викладено в «Положення про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

_____Єрмакова А.М._____
(прізвище, ініціали)

ДОДАТОК В

ІЛЮСТРАТИВНА ЧАСТИНА

**ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОБЛІКУ ТОВАРІВ У КНИЖКОВОМУ
МАГАЗИНІ**

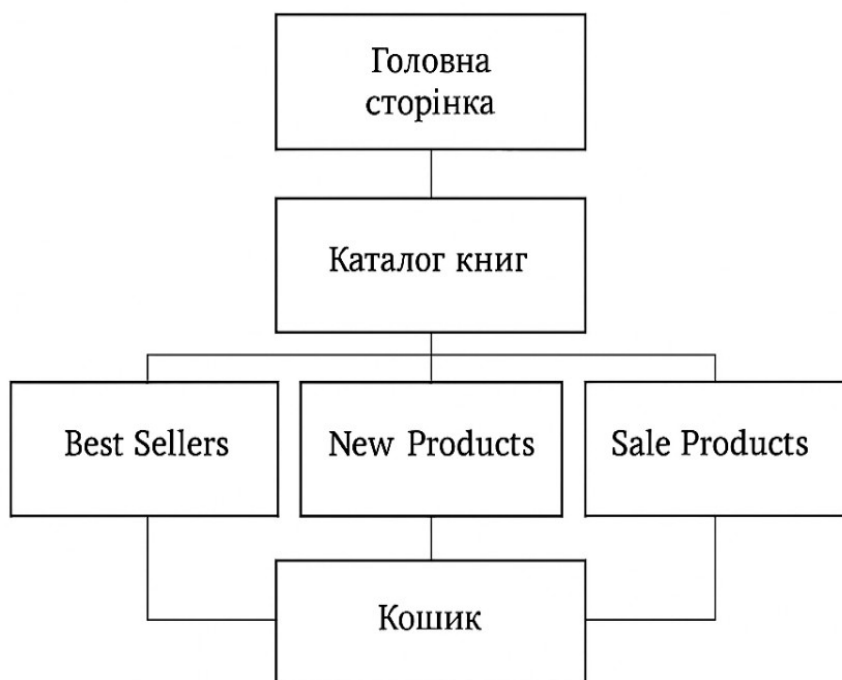


Рисунок В.1 — Відображення структурної схеми

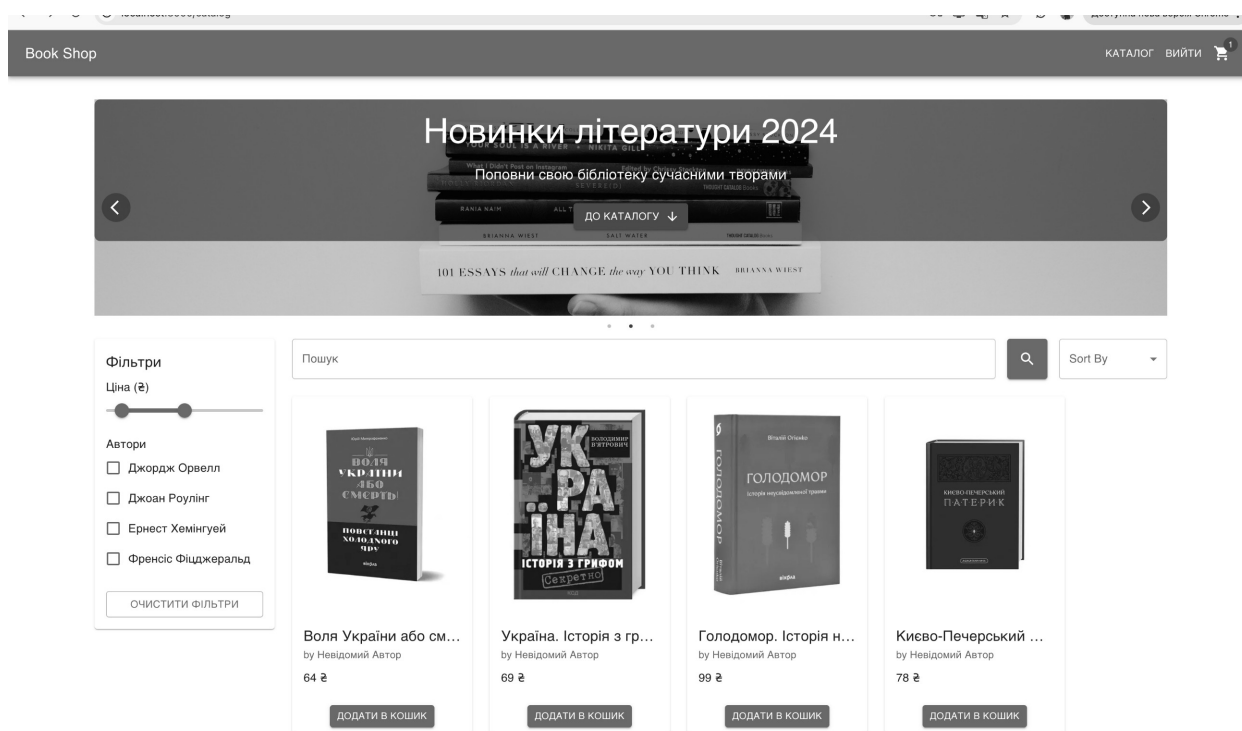


Рисунок В.2 — Вигляд головної сторінки

Search by title or aut...

ADD BOOK

Title ↑	Author	Image	Price	Stock	Actions
(Не)історичні міти. Нариси про минулі сто років. Оновлене	Невідомий Автор		34 \$		EDIT DELETE
24.02 Щоденник війни	Невідомий Автор		67 \$		EDIT DELETE
30 років незалежності України. До 18 серпня 1991 року. Том 1	Невідомий Автор		67 \$		EDIT DELETE
501 факт, який треба знати з... історії України	Невідомий Автор		79 \$		EDIT DELETE
A history of Ukraine. A short course	Невідомий Автор		74 \$		EDIT DELETE
Ukrainer. Ukrainian Insider	Невідомий Автор		143 \$		EDIT DELETE
Ukrainer. Країна зсередини	Невідомий Автор		143 \$		EDIT DELETE
Історичні студії та монографії. Том IV. Україна на переломі 1657–1659	Невідомий Автор		99 \$		EDIT DELETE

Рисунок В.3 — Панель адміністратора

ДОДАТОК Г

Лістинги інтерактивності сторінок

Лістинг Г — React-компонент HeroCarousel (головна сторінка)

```
import React from 'react';
import Slider from 'react-slick';
import { Box, Typography, Button } from '@mui/material';
import ArrowDownwardIcon from '@mui/icons-material/ArrowDownward';

const slides = [
  { title: 'Весняний розпродаж книг!', subtitle: 'Знижки до 50%', image: 'url1' },
  { title: 'Новинки літератури 2024', subtitle: 'Сучасні твори', image: 'url2' },
  { title: 'Класика в моді', subtitle: 'Книги, що пережили століття', image: 'url3' }
];

const HeroCarousel = () => {
  const settings = {
    dots: true, autoplay: true, speed: 500, slidesToShow: 1
  };

  const handleScroll = () => {
    document.getElementById('catalog')?.scrollIntoView({ behavior: 'smooth' });
  };

  return (
    <Box sx={{ mb: 4 }}>
      <Slider {...settings}>
        {slides.map((slide, index) => (
          <Box key={index} sx={{
            height: 300, backgroundImage: `url(${slide.image})`,
            backgroundSize: 'cover', display: 'flex', alignItems: 'center', justifyContent: 'center'
          }}>
            <Box sx={{ bgcolor: 'rgba(0,0,0,0.5)', p: 2 }}>
              <Typography variant="h3">{slide.title}</Typography>
              <Typography variant="h6">{slide.subtile}</Typography>
              <Button variant="contained" onClick={handleScroll}>
                До каталогу
              </Button>
            </Box>
          </Box>
        ))}
      </Slider>
    </Box>
  );
};

export default HeroCarousel;
```