ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Проєктування інформаційної системи книжкового інтернет магазину»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2021
Горб Денис Леонідович
Керівник: Поночовний Юрій
Леонідович
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2025 р.

ВСТУП

Актуальність. У сучасному світі електронна комерція стала невід'ємною частиною економічної діяльності, забезпечуючи зручність купівлі-продажу товарів через Інтернет. За даними аналітичних звітів [1,2], обсяг світового ринку e-commerce у 2024 р. перевищив 6 трлн доларів США, а темпи зростання онлайн-продажів в Україні становлять близько 20–25% щорічно. Особливо популярними стають книжкові інтернет-магазини [3], які дозволяють користувачам швидко знаходити книги, порівнювати ціни та оформлювати замовлення з будь-якої точки світу. В Україні, за оцінками Асоціації видавців, у 2024 р. онлайн-продажі книг склали понад 30% від загального обсягу книжкового ринку, що свідчить про зростання попиту на цифрові платформи для покупки літератури [4].

Однак, незважаючи на зростання популярності онлайн-магазинів, багато існуючих платформ мають обмеження: недостатня локалізація для українського ринку, відсутність адаптивного дизайну, низька швидкість обробки запитів чи слабкий захист даних [5]. Це створює потребу в розробці інформаційних систем, які б відповідали сучасним вимогам користувачів, забезпечували зручність, безпеку та економічну ефективність. Розробка книжкового інтернет-магазину, орієнтованого на український ринок, із підтримкою україномовного контенту, адаптивним дизайном і швидкою обробкою запитів, є актуальним завданням, що сприяє розвитку електронної комерції та популяризації української літератури.

Метою кваліфікаційної роботи є розробка інформаційної системи книжкового інтернет-магазину з використанням сучасних веб-технологій, що забезпечує автоматизацію процесів продажу книг, зручність для користувачів і економічну ефективність.

Відповідно до мети роботи необхідно вирішити наступні завдання:

1) провести аналіз предметної галузі, сучасних технологій, вимог до інформаційних систем книжкових інтернет-магазинів та аналогів;

2) спроєктувати архітектуру бази даних для зберігання даних про книги, користувачів, замовлення та відгуки;

3) реалізувати інформаційну систему, включаючи базу даних, клієнтську та серверну частини, а також провести техніко-економічний аналіз її ефективності.

Об'єкт дослідження – процеси проєктування інформаційних систем для електронної комерції з клієнт-серверною архітектурою.

Предмет дослідження – проєктування та розробка інформаційної системи книжкового інтернет-магазину з використанням веб-технологій і реляційних баз даних.

Методи досліджень – дослідження базуються на методах програмної інженерії, аналізу вимог, проєктування баз даних (нотація Чена, нормалізація), розробки веб-додатків із використанням клієнт-серверної архітектури, а також техніко-економічного аналізу.

Інформаційна база – Інтернет-ресурси, наукові статті, довідкові матеріали про веб-технології (HTML, CSS, JavaScript, PHP, MySQL), стандарти розробки програмного забезпечення, а також статистичні дані про ринок електронної комерції та книжкової індустрії.

Практична значущість – розроблена інформаційна система книжкового інтернет-магазину забезпечує зручність для користувачів і швидкий доступ до контенту. Запропонована архітектура бази даних, модульна структура проєкту та адаптивний інтерфейс спрощують масштабування та підтримку системи, а економічний аналіз підтверджує її ефективність для малого бізнесу.

Результати роботи апробовані в рамках наукової конференції здобувачів вищої освіти за результатами науково-дослідної роботи у 2024-2025 рр.

Структура кваліфікаційної роботи логічно пов'язана з задачами досліджень і містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел. Загальний обсяг текстової частини кваліфікаційної роботи складає 55 сторінок формату А4. Вона містить 11 рисунків і 10 таблиць.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ КНИЖКОВИХ ІНТЕРНЕТ-МАГАЗИНІВ

1.1 Аналіз поняття інформаційної системи та її ролі в електронній комерції

Інформаційна система (ІС) – це сукупність апаратних, програмних, інформаційних і людських ресурсів, які взаємодіють для збору, обробки, зберігання та надання даних з метою підтримки управлінських, операційних і стратегічних функцій організації [6]. У контексті електронної комерції ІС відіграє ключову роль, забезпечуючи автоматизацію бізнес-процесів, управління даними про товари, клієнтів і транзакції, а також взаємодію між користувачами та бізнесом через цифрові канали. Основними компонентами ІС є бази даних, серверні та клієнтські програми, а також інтерфейси користувача, які забезпечують зручність взаємодії.

Електронна комерція (e-commerce) – це сфера економічної діяльності, що передбачає здійснення комерційних операцій через Інтернет [7]. За даними досліджень, обсяг світового ринку e-commerce у 2024 р. перевищив 6 трлн. доларів США, що підкреслює її зростаючу значущість [8]. ІС у цій сфері виконує функції каталогізації товарів, обробки замовлень, управління складом, забезпечення безпеки транзакцій і персоналізації пропозицій для клієнтів. Наприклад, у книжкових інтернет-магазинах ІС дозволяє користувачам шукати книги за жанрами, авторами чи ключовими словами, додавати їх до кошика та оформлювати замовлення.

Основні функції ІС в електронній комерції можна представити у вигляді таблиці 1.1. Роль ІС у книжкових інтернет-магазинах полягає у створенні зручного та ефективного середовища для купівлі-продажу. Наприклад, архітектура ІС зазвичай базується на шаблоні MVC (Model-View-Controller), що забезпечує розділення даних, інтерфейсу та логіки управління [9].

Таблиця 1.1 – Функції інформаційної системи в електронній комерції

Функція	Опис
Управління каталогом	Формування структури товарів, їх описів, цін та зображень.
Обробка транзакцій	Автоматизація платежів, підтвердження замовлень, генерація чеків.
Комунікація з клієнтами	Надсилання сповіщень, підтримка через чати або форми зворотного зв'язку.
Аналітика та персоналізація	Збір даних про поведінку користувачів для формування рекомендацій.
Безпека даних	Захист персональних даних і транзакцій за допомогою шифрування.

Модель MVC відповідає за взаємодію з базою даних, вигляд – за відображення інформації користувачу, а контролер – за обробку запитів. Схема роботи такої системи наведена на рис.1.1 [10].

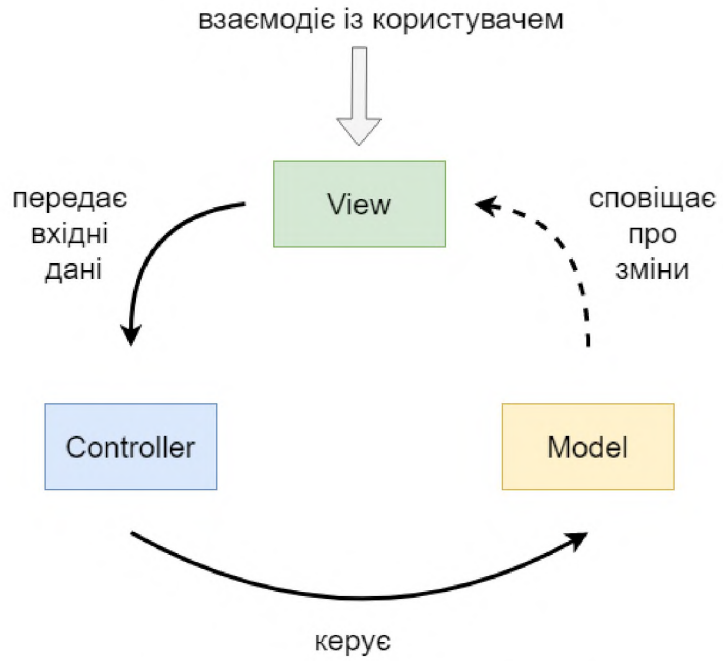

Рисунок 1.1 – Схема роботи інформаційної системи за шаблоном MVC [10]

Ефективність ІС залежить від правильного вибору технологій. Наприклад, для серверної частини часто використовують PHP або Python [11], для бази даних – MySQL [12], а для клієнтської частини – HTML, CSS і JavaScript [13]. Ці технології забезпечують швидкість обробки запитів, масштабованість і зручність для користувачів. У книжкових інтернет-магазинах ІС також сприяє зниженню

витрат на фізичну інфраструктуру порівняно з традиційними магазинами, що робить бізнес більш конкурентоспроможним.

Отже, інформаційна система є основою електронної комерції, забезпечуючи автоматизацію процесів, зручність для користувачів і економічну ефективність. У контексті книжкових інтернет-магазинів ІС не лише полегшує доступ до асортименту, але й створює можливості для масштабування бізнесу та підвищення якості обслуговування клієнтів.

1.2 Огляд сучасних технологій створення веб-додатків для інтернет-магазинів

Розробка веб-додатків для інтернет-магазинів є складним процесом, що вимагає використання сучасних технологій для забезпечення швидкості, безпеки, масштабованості та зручності користувацького інтерфейсу. Сучасні веб-додатки для електронної комерції базуються на комбінації клієнтських і серверних технологій, систем управління базами даних (СУБД) та інструментів для створення інтерактивних інтерфейсів. Цей підрозділ присвячено огляду ключових технологій, які використовуються для створення веб-додатків інтернет-магазинів, зокрема книжкових, з урахуванням їх переваг і недоліків.

Клієнтська частина веб-додатку відповідає за відображення контенту в браузері користувача та забезпечення інтерактивності. Основними технологіями тут є HTML, CSS і JavaScript [13].

HTML (HyperText Markup Language) – стандартизована мова розмітки, що використовується для структурування вмісту веб-сторінок [14]. HTML визначає елементи, такі як заголовки, списки, таблиці та форми, які формують основу сторінки. Завдяки підтримці семантичних тегів (наприклад, <header>, <article>) HTML5 полегшує створення доступних і SEO-оптимізованих сторінок.

CSS (Cascading Style Sheets) – мова для опису зовнішнього вигляду веб-сторінок [15]. CSS дозволяє задавати кольори, шрифти, розташування елементів

і адаптивність дизайну для різних пристроїв. Наприклад, використання фреймворків, таких як Tailwind CSS або Bootstrap, значно прискорює розробку стилів для книжкових інтернет-магазинів, забезпечуючи адаптивність для мобільних пристроїв.

JavaScript – мова програмування, що додає інтерактивність, наприклад, динамічне оновлення вмісту, обробку подій (натискання кнопок, заповнення форм) і асинхронні запити до сервера [16]. Популярні бібліотеки, такі як React або Vue.js, дозволяють створювати компонентно-орієнтовані інтерфейси, що підвищують швидкість роботи користувача з каталогом книг чи кошиком.

Серверна частина веб-додатку обробляє запити користувачів, взаємодіє з базою даних і формує відповіді. PHP – серверна мова програмування, проста у вивченні та широко поширена [17]. Вона підтримує роботу з базами даних через розширення, такі як PDO або MySQLi, і використовується для створення динамічних сторінок. Незважаючи на критику щодо застарілої об'єктно-орієнтованої моделі, PHP залишається популярним завдяки великій кількості готових бібліотек і фреймворків, таких як Laravel, який спрощує розробку складних веб-додатків.

Node.js – сучасна альтернатива PHP, що дозволяє використовувати JavaScript на сервері [18]. Node.js забезпечує високу продуктивність завдяки асинхронній обробці запитів, що особливо важливо для інтернет-магазинів із великою кількістю одночасних користувачів.

Python з фреймворками Django або Flask – ще одна популярна технологія, що вирізняється простотою синтаксису та потужними бібліотеками для роботи з даними [19]. Django, наприклад, пропонує вбудовані інструменти для управління користувачами, що корисно для реалізації системи авторизації в книжковому магазині.

Системи управління базами даних (СУБД) є ключовим компонентом для зберігання даних про товари, користувачів і замовлення. Так, СУБД MySQL є популярним вибором через свою швидкість і простоту.

MySQL – реляційна СУБД, що підтримує швидкий доступ до даних і транзакції [12]. Вона ідеально підходить для малих і середніх інтернет-магазинів завдяки підтримці типів таблиць, таких як InnoDB (для транзакцій) і MyISAM (для швидкого пошуку). Наприклад, таблиця books у книжковому магазині може містити поля id, name, author, price, category_id.

PostgreSQL – альтернатива MySQL, що пропонує розширені можливості, такі як підтримка JSON для зберігання складних даних чи географічних об'єктів. Вона підходить для магазинів із великими обсягами даних [21].

MongoDB – нереляційна СУБД, що використовується для роботи з документо-орієнтованими даними. Вона може бути корисною для зберігання відгуків користувачів або динамічних описів книг [22].

Таблиця 1.2 – Порівняння СУБД для інтернет-магазинів

СУБД	Переваги	Недоліки
MySQL	Висока швидкість, простота, широка підтримка	Обмежені можливості для складних запитів
PostgreSQL	Підтримка JSON, розширені функції	Вища складність налаштування
MongoDB	Гнучкість для неструктурованих даних	Менша ефективність для реляційних даних

Для створення веб-додатків часто використовується шаблон MVC (Model-View-Controller), що розділяє логіку даних, інтерфейс і обробку запитів. У книжковому магазині модель відповідає за доступ до бази даних (наприклад, вибірка книг за жанром), вигляд – за відображення каталогу, а контролер – за обробку дій користувача (додавання книги до кошика).

Для створення веб-додатків використовуються локальні сервери, такі як Apache або сучасніші альтернативи, як Nginx. Локальні середовища, такі як XAMPP або Docker, дозволяють тестувати додатки без підключення до мережі. Наприклад, Apache підтримує обробку запитів і роботу з PHP-скриптами, що забезпечує швидке розгортання веб-додатку [23].

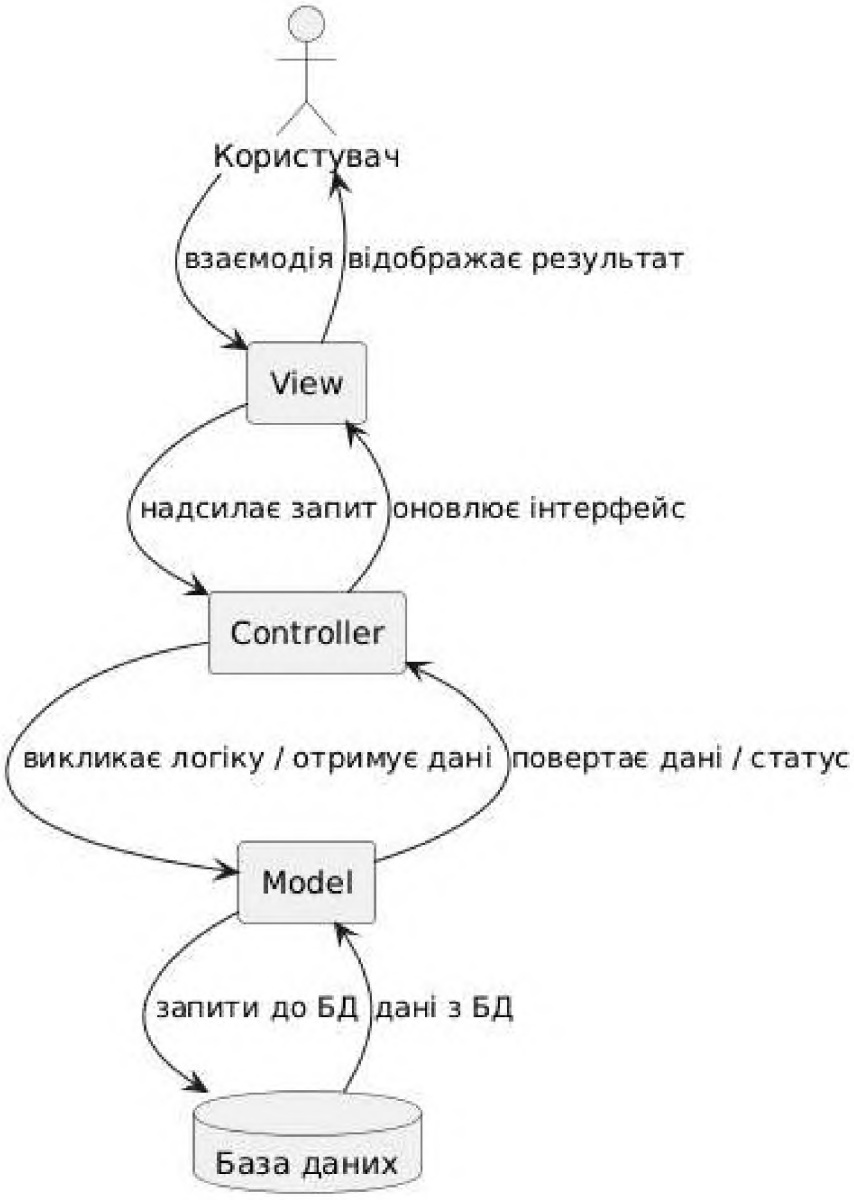

Рисунок 1.2 – Взаємодія компонент веб-додатку на основі MVC

Отже, сучасні технології створення веб-додатків для інтернет-магазинів включають комбінацію клієнтських (HTML, CSS, JavaScript), серверних (PHP, Node.js, Python) технологій, СУБД (MySQL, PostgreSQL, MongoDB) і архітектурних шаблонів, таких як MVC. Вибір конкретних інструментів залежить від вимог до продуктивності, масштабованості та зручності інтерфейсу. Для книжкових інтернет-магазинів оптимальним є поєднання PHP з MySQL для серверної частини та React для клієнтської, що забезпечує швидкість, надійність і адаптивність [24].

1.3 Аналіз вимог до інформаційних систем книжкових інтернет-магазинів

Інформаційна система (ІС) книжкового інтернет-магазину є складним програмно-апаратним комплексом, який забезпечує автоматизацію бізнес-процесів, пов'язаних із продажем книг через Інтернет [10]. Вимоги до таких систем формуються на основі потреб користувачів, бізнес-цілей компанії та сучасних технологічних стандартів. Цей підрозділ присвячено аналізу функціональних і нефункціональних вимог до ІС книжкових інтернет-магазинів, з урахуванням їх специфіки та особливостей електронної комерції.

Функціональні вимоги визначають задачі, які ІС має виконувати для задоволення потреб користувачів і бізнесу [24]. Основні функціональні вимоги до ІС книжкового інтернет-магазину наведені у таблиці 1.3.

Таблиця 1.3 – Основні функціональні вимоги до ІС книжкового інтернет-магазину

Функція	Опис	Приклад реалізації
Каталог	Відображення книг із фільтрацією та сортуванням	Сторінка з картками книг, фільтри за жанром
Пошук	Пошук за ключовими словами чи категоріями	Рядок пошуку, категорії в бічному меню
Кошик	Додавання, видалення, оформлення замовлення	Сторінка кошика з підсумковою вартістю
Авторизація	Реєстрація, вхід, відновлення пароля	Форма входу/реєстрації
Замовлення	Управління статусами замовлень	Панель адміністратора для обробки

Основні функціональні вимоги до ІС книжкового інтернет-магазину включають:

1. Управління каталогом товарів – система повинна відображати книги з детальною інформацією, такою як назва, автор, опис, ціна, обкладинка та жанр. Користувачі мають мати можливість переглядати каталог, сортувати товари за різними критеріями (наприклад, за ціною чи популярністю) та фільтрувати за жанрами чи авторами.

2. Пошук і навігація – ІС має забезпечувати швидкий пошук книг за ключовими словами (наприклад, назвою чи автором) і навігацію за категоріями (жанрами).

3. Кошик і оформлення замовлення – система повинна дозволяти додавати книги до кошика, переглядати вибрані товари, змінювати їх кількість, видаляти позиції та оформлювати замовлення.

4. Реєстрація та авторизація – для здійснення покупок користувачі повинні мати можливість зареєструватися та увійти в систему. ІС має забезпечувати безпечне зберігання даних, зокрема паролів, за допомогою сучасних методів шифрування, таких як bcrypt.

5. Управління замовленнями – система має фіксувати історію замовлень, надавати користувачам доступ до їх статусу та забезпечувати адміністраторам можливість обробки замовлень (підтвердження, відправлення, скасування).

6. Зворотний зв'язок – ІС повинна підтримувати форми для відгуків, оцінок книг і звернень до служби підтримки.

Нефункціональні вимоги визначають якісні характеристики системи, такі як продуктивність, безпека, масштабованість і зручність використання [25].

1. Продуктивність – ІС повинна обробляти запити з мінімальною затримкою. Наприклад, час відгуку сторінки каталогу не має перевищувати 2 секунди за одночасного доступу 1000 користувачів. Це вимагає оптимізації запитів до бази даних і використання кешування.

2. Безпека – система має забезпечувати захист персональних даних і транзакцій через HTTPS, шифрування та захист від SQL-ін'єкцій за допомогою підготовлених запитів.

3. Масштабованість – ІС повинна підтримувати зростання кількості користувачів і товарів без значного зниження продуктивності. Наприклад, реляційна база даних, як MySQL, підходить для малих і середніх магазинів, але для великих може знадобитися PostgreSQL.

4. Зручність використання (usability) – інтерфейс має бути інтуїтивно зрозумілим, із чіткою навігацією та адаптивним дизайном для різних пристроїв (ПК, планшети, смартфони).

5. Надійність – система повинна працювати без збоїв, з мінімальним часом простою (не більше 0,01% на рік), що вимагає регулярного резервного копіювання даних і моніторингу серверів.

Для наочного представлення взаємодії користувачів із системою розроблено діаграму варіантів використання, яка відображає основні сценарії використання ІС книжкового інтернет-магазину (рис. 1.3). Діаграма включає два основні типи акторів: Користувач (зареєстрований або гість) і Адміністратор. Користувач може виконувати дії, такі як перегляд каталогу, пошук книг, додавання до кошика, оформлення замовлення, реєстрація, авторизація та залишення відгуків. Адміністратор відповідає за управління каталогом, обробку замовлень і модерацію відгуків.

Рисунок 1.3 – Діаграма використання ІС книжкового інтернет-магазину

Книжкові інтернет-магазини мають унікальні вимоги, пов'язані з особливостями асортименту. Книги характеризуються кількома атрибутами (автор, видавництво, рік видання, ISBN), які необхідно враховувати в структурі бази даних. Користувачі можуть шукати книги за специфічними критеріями, наприклад, за серією чи мовою видання. ІС також має підтримувати рекомендаційні системи, які пропонують книги на основі попередніх покупок або переглядів, що підвищує рівень персоналізації.

Отже, вимоги до ІС книжкового інтернет-магазину охоплюють функціональні та нефункціональні аспекти, спрямовані на забезпечення зручності, безпеки та ефективності. Функціональні вимоги включають управління каталогом, пошук, кошик і авторизацію, тоді як нефункціональні акцентують увагу на продуктивності, масштабованості та зручності. Специфічні вимоги, такі як підтримка рекомендацій і детальних характеристик книг, роблять ці системи унікальними в контексті електронної комерції.

1.4 Аналіз аналогів інформаційних систем книжкових магазинів

Аналіз аналогів інформаційних систем (ІС) книжкових інтернет-магазинів є важливим етапом розробки, оскільки дозволяє оцінити сильні та слабкі сторони існуючих рішень, визначити найкращі практики та уникнути типових помилок. У цьому підрозділі розглядаються три популярні ІС книжкових інтернет-магазинів: Amazon Books, Yakaboo та Book Depository. Аналіз охоплює їх функціональність, технології, інтерфейс, безпеку та ефективність, а також визначає ключові аспекти, які можна врахувати при розробці нової ІС.

Amazon є світовим лідером у сфері електронної комерції, а його підрозділ Amazon Books пропонує широкий асортимент книг у цифровому та фізичному форматах [11]. ІС Amazon характеризується високою продуктивністю, розвиненою системою рекомендацій і потужною інфраструктурою. Основні функції включають пошук за ключовими словами, фільтрацію за жанрами,

відгуки користувачів, оформлення замовлень і персоналізовані рекомендації на основі історії покупок. Система використовує хмарні технології AWS, що забезпечує масштабованість і надійність.

Yakaboo – це українська платформа для продажу книг, яка фокусується на локальному ринку [12]. ІС Yakaboo пропонує зручний інтерфейс із підтримкою української та російської мов, каталог із фільтрами за категоріями, акціями та новинками, а також інтеграцію з платіжними системами, такими як LiqPay. Система підтримує функцію відгуків, але рекомендаційна система менш розвинена порівняно з Amazon. Yakaboo використовує реляційні бази даних (ймовірно, MySQL або PostgreSQL) і сучасні фреймворки для серверної логіки.

Book Depository – міжнародний книжковий інтернет-магазин, відомий безкоштовною доставкою по всьому світу [13]. ІС Book Depository пропонує простий інтерфейс, ефективний пошук і фільтрацію за жанрами, мовами та видавництвами. Система підтримує багатомовність і інтеграцію з платіжними системами, такими як PayPal. Проте функціонал рекомендацій обмежений, а продуктивність сайту може знижуватися під час пікових навантажень.

Таблиця 1.4 – Порівняння аналогів ІС книжкових інтернет-магазинів

Критерій	Amazon Books	Yakaboo	Book Depository
Функціональність	Висока (рекомендації, відгуки, фільтри, інтеграція)	Середня (фільтри, відгуки, базові рекомендації)	Середня (фільтри, багатомовність, обмежені рекомендації)
Зручність інтерфейсу	Висока (адаптивний дизайн, інтуїтивна навігація)	Висока (локалізований інтерфейс, адаптивність)	Середня (простий, але менш інтуїтивний дизайн)
Продуктивність	Висока (хмарна інфраструктура AWS)	Середня (залежить від локальних серверів)	Середня (можливі затримки при пікових навантаженнях)
Безпека	Висока (HTTPS, шифрування, захист від атак)	Середня (HTTPS, базовий захист даних)	Середня (HTTPS, обмежена інформація про безпеку)
Масштабованість	Висока (глобальна інфраструктура)	Середня (локальний ринок)	Висока (міжнародний ринок)

Для оцінки аналогів використано кілька критеріїв: функціональність, зручність інтерфейсу, продуктивність, безпека та масштабованість. Результати порівняння представлено в таблиці 1.4.

Перевагою Amazon Books є потужна рекомендаційна система, яка використовує алгоритми машинного навчання для аналізу поведінки користувачів. Недоліком є складність системи, що вимагає значних ресурсів для впровадження. Перевагою Yakaboo є локалізація для українського ринку, що включає підтримку української мови та місцевих платіжних систем. Недоліком є обмежена функціональність рекомендацій і менша масштабованість порівняно з глобальними платформами. Перевагою Book Depository є простота інтерфейсу та безкоштовна доставка, що приваблює міжнародних клієнтів. Недоліком є слабка персоналізація та періодичні проблеми з продуктивністю.

На основі аналізу аналогів можна визначити ключові аспекти для розробки нової ІС книжкового інтернет-магазину:

1. Функціональність – необхідно реалізувати базові функції (каталог, пошук, кошик, авторизація) та додати рекомендаційну систему, подібну до Amazon, але спрощену для зниження витрат. Схема взаємодії ключових компонентів ІС книжкового інтернет-магазину показана на рис.1.4.

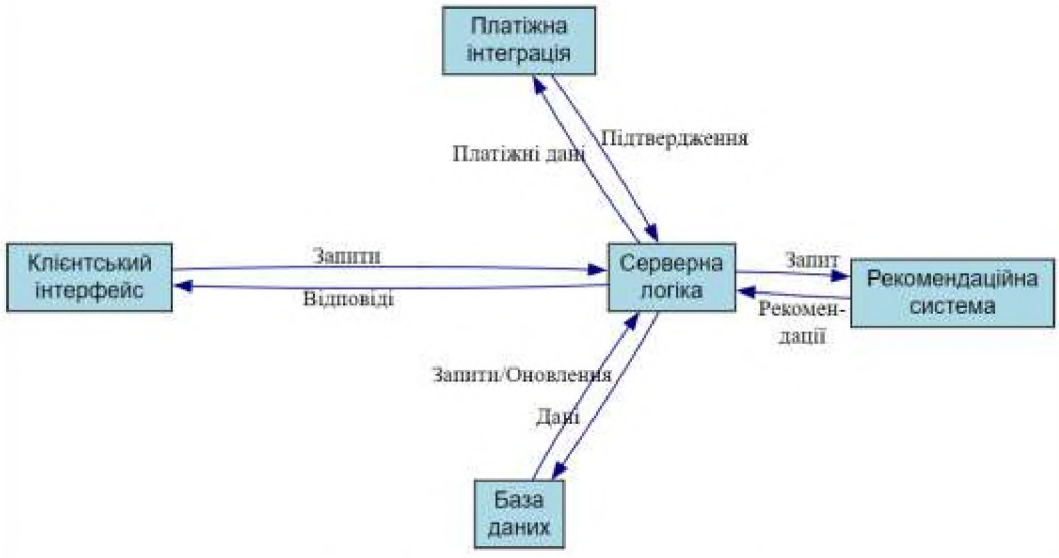

Рисунок 1.4 – Схема взаємодії ключових компонентів ІС книжкового інтернет-магазину

2. Інтерфейс – слід створити адаптивний і локалізований інтерфейс, подібний до Yakaboo, з акцентом на інтуїтивну навігацію.

3. Продуктивність і масштабованість – використання реляційних СУБД (MySQL або PostgreSQL) і хмарних сервісів (наприклад, AWS або Google Cloud) дозволить забезпечити високу продуктивність і масштабованість.

4. Безпека – впровадження HTTPS, шифрування даних і захисту від SQL-ін'єкцій є обов'язковим.

Для реалізації ІС можна використати комбінацію технологій: PHP або Python для серверної логіки, MySQL для бази даних, HTML/CSS/JavaScript для клієнтської частини. Наприклад, для обробки запитів пошуку можна застосувати SQL-запит:

```
SELECT * FROM books WHERE name LIKE '%query%';
```

Це забезпечує швидкий пошук за назвою чи автором книги. Для рекомендацій можна використати алгоритми колаборативної фільтрації, які аналізують історію покупок користувачів.

Отже, аналіз аналогів показав, що успішна ІС книжкового інтернет-магазину повинна поєднувати розвинену функціональність, зручний інтерфейс, високу продуктивність і безпеку. Amazon демонструє передові технології, Yakaboo – локалізацію, а Book Depository – простоту. Ці аспекти слід врахувати при розробці нової системи, щоб забезпечити конкурентоспроможність і зручність для користувачів.

1.5 Обґрунтування вибору технологій та інструментів для розробки

Розробка ІС книжкового інтернет-магазину вимагає ретельного вибору технологій та інструментів, які забезпечать ефективність, масштабованість, безпеку та зручність використання. Вибір залежить від функціональних і нефункціональних вимог, масштабів проєкту, а також доступних ресурсів і кваліфікації розробників. У цьому підрозділі обґрунтовується вибір технологій

для клієнтської та серверної частин, системи управління базами даних (СУБД), а також інструментів для тестування й розгортання системи.

Для клієнтської частини ІС, яка відповідає за відображення інтерфейсу та взаємодію з користувачем, обрано комбінацію HTML, CSS і JavaScript із використанням бібліотеки React. HTML забезпечує структуру веб-сторінок, дозволяючи створювати семантичні елементи (наприклад, <section>, <article>), що полегшують навігацію та SEO-оптимізацію. HTML5 підтримує адаптивність і мультимедійні можливості, що є важливим для відображення обкладинок книг і форм замовлення. CSS використовується для стилізації інтерфейсу. Для прискорення розробки та забезпечення адаптивності обрано фреймворк Tailwind CSS, який дозволяє швидко створювати сучасні, респонсивні дизайни без написання великої кількості власних стилів.

JavaScript із бібліотекою React обрано для створення динамічних, компонентно-орієнтованих інтерфейсів. React забезпечує швидке оновлення вмісту сторінки без перезавантаження (наприклад, при додаванні книги до кошика) і спрощує розробку складних інтерфейсів, таких як фільтри каталогу чи рекомендаційні блоки. Перевагою React є велика спільнота розробників і готові компоненти, що знижують час розробки.

Для серверної частини, яка обробляє запити, взаємодіє з базою даних і формує відповіді, обрано мову програмування PHP із фреймворком Laravel. Альтернативою розглядалися Node.js і Python (Django), але PHP обрано з таких причин:

1. Простота та популярність, PHP є широко поширеним і легким у вивченні, що полегшує підтримку системи. За даними статистики, PHP використовується на 20% вебсайтів, що підтверджує його надійність.

2. Фреймворк Laravel пропонує вбудовані інструменти для авторизації, маршрутизації, роботи з базами даних і захисту від атак (наприклад, CSRF і SQL-ін'єкцій). Це дозволяє швидко реалізувати функціонал кошика, авторизації та обробки замовлень.

3. Сумісність із СУБД – PHP має потужні розширення, такі як PDO, для роботи з MySQL, що забезпечує ефективну взаємодію з базою даних.

Для зберігання даних про книги, користувачів і замовлення обрано реляційну СУБД MySQL. Основні причини вибору:

– швидкість і простота;

– гнучкість;

– широка підтримка.

Альтернативою розглядалася PostgreSQL, яка пропонує розширені можливості, такі як підтримка JSON, але для малого чи середнього книжкового магазину MySQL є оптимальним вибором через простоту налаштування та нижчі вимоги до ресурсів. Результати порівняння технологій для розробки ІС наведені у таблиці 1.5.

Таблиця 1.5 – Порівняння технологій для розробки ІС

Компонент	Технологія	Переваги	Недоліки
Клієнтська частина	HTML, CSS (Tailwind), JavaScript (React)	Адаптивність, швидкість, компонентний підхід	Вимагає вивчення React
Серверна частина	PHP (Laravel)	Простота, безпека, швидка розробка	Менша продуктивність порівняно з Node.js
СУБД	MySQL	Швидкість, простота, широка підтримка	Обмежені можливості для складних даних

Для створення, тестування та розгортання ІС обрано такі інструменти:

1. Локальний сервер – OpenServer використовується для локального тестування PHP і MySQL. OpenServer забезпечує швидке розгортання Apache або Nginx-сервера, бази даних і підтримку різних версій PHP, що дозволяє гнучко налаштовувати середовище для розробки системи без підключення до Інтернету.

2. Середовище розробки – Visual Studio Code обрано як основний редактор коду завдяки підтримці розширень для PHP, JavaScript і MySQL, а також інтеграції з Git для контролю версій.

3. Контроль версій – Git із репозиторієм на GitHub використовується для управління кодом і співпраці між розробниками.

4. Тестування – для тестування інтерфейсу застосовується Cypress, який дозволяє автоматизувати перевірку функціоналу, наприклад, додавання книги до кошика. Для серверної частини використовується PHPUnit, інтегрований із Laravel.

Для організації взаємодії компонентів системи обрано шаблон MVC (Model-View-Controller), який розділяє логіку даних, інтерфейс і обробку запитів. Модель відповідає за взаємодію з MySQL, вигляд – за відображення через React, а контролер – за обробку запитів у Laravel. Схема взаємодії компонентів наведена на рис. 1.5.

Рисунок 1.5 – Схема архітектури ІС на основі MVC

Вибір технологій зумовлений їхньою ефективністю та відповідністю вимогам проєкту. HTML, CSS і React забезпечують сучасний, адаптивний інтерфейс, PHP із Laravel – швидку розробку серверної логіки, а MySQL – надійне зберігання даних. Інструменти, такі як OpenServer і Git, спрощують розробку та співпрацю. Ця комбінація дозволяє створити масштабовану, безпечну та зручну ІС для книжкового інтернет-магазину з мінімальними витратами.

Отже, обрані технології (HTML, CSS, JavaScript/React, PHP/Laravel, MySQL) та інструменти (OpenServer, Visual Studio Code, Git, Cypress, PHPUnit) відповідають вимогам до продуктивності, безпеки та зручності. Вони забезпечують швидку розробку, легке масштабування та підтримку сучасних стандартів електронної комерції, що робить їх оптимальними для створення ІС книжкового інтернет-магазину.

РОЗДІЛ 2
ПРОЄКТУВАННЯ БАЗИ ДАНИХ ІНФОРМАЦІЙНОЇ СИСТЕМИ КНИЖКОВОГО ІНТЕРНЕТ-МАГАЗИНУ

2.1 Розробка структури бази даних

Розробка структури бази даних (БД) є ключовим етапом проєктування інформаційної системи (ІС) книжкового інтернет-магазину, оскільки від правильної організації даних залежить ефективність роботи системи, швидкість обробки запитів і зручність управління інформацією. Структура БД має бути повною, нормалізованою, уникати надмірності даних і забезпечувати швидкий доступ до інформації. У цьому підрозділі описано процес створення інфологічної та концептуальної моделі БД, визначено сутності, атрибути, зв'язки, а також представлено діаграму «сутність-зв'язок» у нотації Чена з урахуванням принципів нормалізації.

Інформаційна система книжкового інтернет-магазину призначена для автоматизації процесів продажу книг через Інтернет, включаючи управління каталогом, обробку замовлень, авторизацію користувачів і підтримку зворотного зв'язку. Основна мета БД – забезпечити зберігання та швидкий доступ до даних про книги, користувачів, замовлення, категорії (жанри) та відгуки, а також підтримувати аналітичні запити, наприклад, для рекомендаційних систем. БД має відповідати вимогам ефективності, безпеки та масштабованості, щоб підтримувати зростання кількості користувачів і товарів.

Для визначення структури БД сформульовано основні прецеденти використання системи:

1. Реєстрація користувача – занесення даних нового користувача (логін, пароль, email).

2. Оформлення замовлення – створення замовлення з інформацією про книги, користувача, дату та статус.

3. Перегляд каталогу – відображення книг із фільтрацією за жанрами чи пошуком за назвою/автором.

4. Додавання книги до кошика – занесення даних про вибрані книги для певного користувача.

5. Залишення відгуку – додавання оцінки та коментаря до книги.

6. Аналіз популярності книг – генерація звіту про найпопулярніші книги за кількістю замовлень.

7. Оновлення статусу замовлення – зміна статусу (наприклад, «обробляється», «відправлено») адміністратором.

8. Управління каталогом – додавання, редагування або видалення книг адміністратором.

Опис прецедентів використання інформаційної системи книжкового інтернет-магазину наступний:

1. Реєстрація користувача: мета – додати нового користувача до системи. Завдання: ввести логін, email, зашифрований пароль; перевірити унікальність email; зберегти дані. Учасники: користувач, система.

2. Оформлення замовлення: мета – зафіксувати покупку. Завдання: обрати книги з кошика, вказати спосіб доставки, зберегти дату та статус замовлення. Учасники: користувач, система.

3. Перегляд каталогу: мета – надати доступ до книг. Завдання: відобразити список книг із фільтрами (жанр, ціна) або результатами пошуку. Учасники: користувач, система.

4. Додавання книги до кошика: мета – зберегти вибір користувача. Завдання: додати ідентифікатор книги та користувача до кошика. Учасники: користувач, система.

5. Залишення відгуку: мета – зберегти оцінку книги. Завдання: ввести рейтинг (1-5), коментар, пов’язати з книгою та користувачем. Учасники: користувач, система.

Для визначення сутностей, атрибутів і зв’язків проведено морфологічний аналіз тексту вимог з визначенням іменників та дієслів.

Іменники – кандидати в сутності та їх атрибути, наведені у таблиці 2.1:

– користувач – людина, яка реєструється та робить замовлення; атрибути – ідентифікатор, логін, пароль, email;

– книга – товар, доступний для покупки; атрибути – ідентифікатор, назва, автор, опис, ціна, обкладинка, жанр;

– замовлення – процес покупки; атрибути – ідентифікатор, дата, статус, загальна сума;

– кошик – тимчасовий набір книг користувача; атрибути – ідентифікатор книги, ідентифікатор користувача, кількість;

– категорія (жанр) – класифікація книг; атрибути – ідентифікатор, назва;

– відгук – оцінка книги користувачем; атрибути: ідентифікатор, рейтинг, коментар, дата.

Таблиця 2.1 – Сутності та їх атрибути

Сутність	Атрибути
Користувач	id (INT, PK), login (VARCHAR), password (VARCHAR), email (VARCHAR)
Книга	id (INT, PK), title (VARCHAR), author (VARCHAR), description (TEXT), price (DECIMAL), image (VARCHAR), category_id (INT, FK)
Категорія	id (INT, PK), name (VARCHAR)
Замовлення	id (INT, PK), user_id (INT, FK), date (DATETIME), status (VARCHAR), total (DECIMAL)
Кошик	book_id (INT, FK), user_id (INT, FK), quantity (INT)
Відгук	id (INT, PK), book_id (INT, FK), user_id (INT, FK), rating (INT), comment (TEXT), date (DATETIME)

Таблиця 2.2 – Зв’язки між сутностями

Сутність 1	Зв’язок	Сутність 2	Опис
Користувач	оформляє	Замовлення	Один користувач може мати багато замовлень
Користувач	додає до	Кошик	Користувач додає книги до кошика
Книга	належить до	Категорія	Одна книга належить до однієї категорії
Користувач	залишає	Відгук	Користувач може залишити відгук про книгу
Замовлення	містить	Книга	Замовлення включає одну або більше книг

Дієслова – кандидати в зв’язки, представлені у таблиці 2.2:

– користувач оформляє замовлення;

– користувач додає книгу до кошика;
– книга належить до категорії;
– користувач залишає відгук про книгу;
– замовлення містить книги.

На основі морфологічного аналізу розроблено концептуальну модель БД у нотації Чена, яка відображає сутності, їх атрибути та зв'язки з урахуванням кардинальності (рис. 2.1).

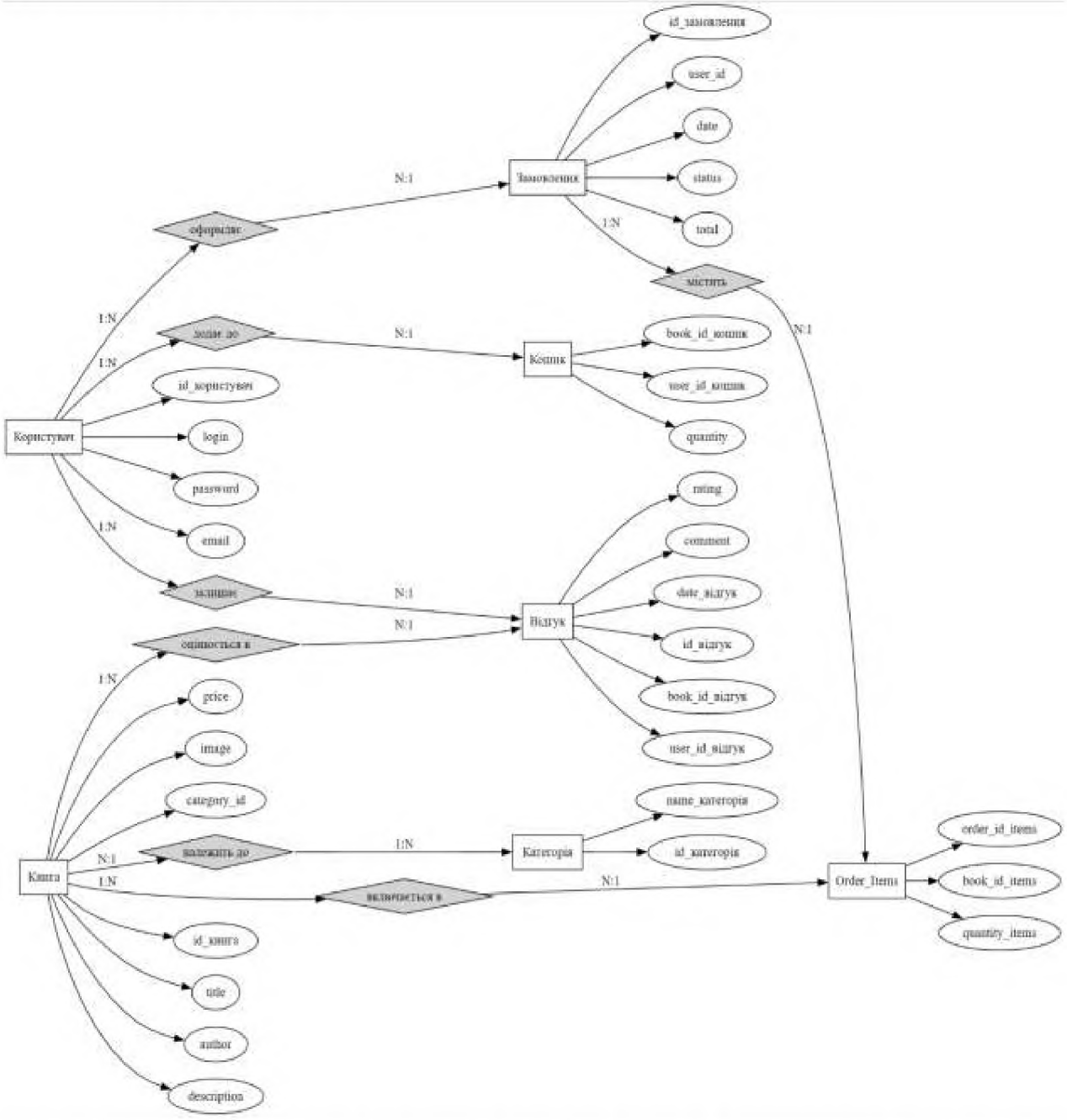

Рисунок 2.1 – Діаграма «сутність-зв'язок» для БД книжкового інтернет-магазину

Основні зв'язки:

– Користувач – Замовлення: один до багатьох (1:N). Один користувач може мати багато замовлень, але кожне замовлення належить одному користувачу.

– Книга – Категорія: багато до одного (N:1). Книга належить до однієї категорії, але категорія може містити багато книг.

– Користувач – Кошик: один до багатьох (1:N). Користувач має один кошик, який містить кілька записів про книги.

– Замовлення – Книга: Багато до багатьох (N:M). Замовлення може включати кілька книг, а книга може бути частиною кількох замовлень. Для реалізації цього зв'язку створено проміжну таблицю order_items.

Для усунення надмірності даних і забезпечення цілісності БД проведено нормалізацію до третьої нормальної форми (3NF):

1. Перша нормальна форма (1NF): усі атрибути є атомарними, таблиці не містять повторюваних груп. Наприклад, таблиця books має окремі поля для назви, автора та ціни.

2. Друга нормальна форма (2NF): усунуто часткову залежність. Наприклад, у таблиці orders атрибут total залежить від повного ключа (order_id), а не від частини ключа.

3. Третя нормальна форма (3NF): усунуто транзитивні залежності. Наприклад, поле category_id у таблиці books посилається на таблицю categories, а не дублюється.

Для підвищення продуктивності передбачено індексацію полів, які часто використовуються в запитах, наприклад, title і author у таблиці books для пошуку. Також використано зовнішні ключі для забезпечення референційної цілісності, наприклад, category_id у таблиці books посилається на id у таблиці categories.

На основі діаграми «сутність-зв'язок» для інформаційної системи книжкового інтернет-магазину розроблено EER діаграму моделі даних предметної області (рис. 2.2). EER діаграма відображає сутності (User, Book,

Category, Order, Cart, Review, Order_Items), їх атрибути та зв'язки з кардинальністю.

При побудові діаграми було внесено наступні зміни. Замінено україномовні назви на англійські: Користувач → User, Книга → Book, Категорія → Category, Замовлення → Order, Кошик → Cart, Відгук → Review. Назви зв'язків перекладено на англійські дієслова, що відповідають діям: оформляє → places, додає до → adds to, належить до → belongs to, залишає → writes, оцінюється в → is rated in, містить → contains, включається в → is included in.

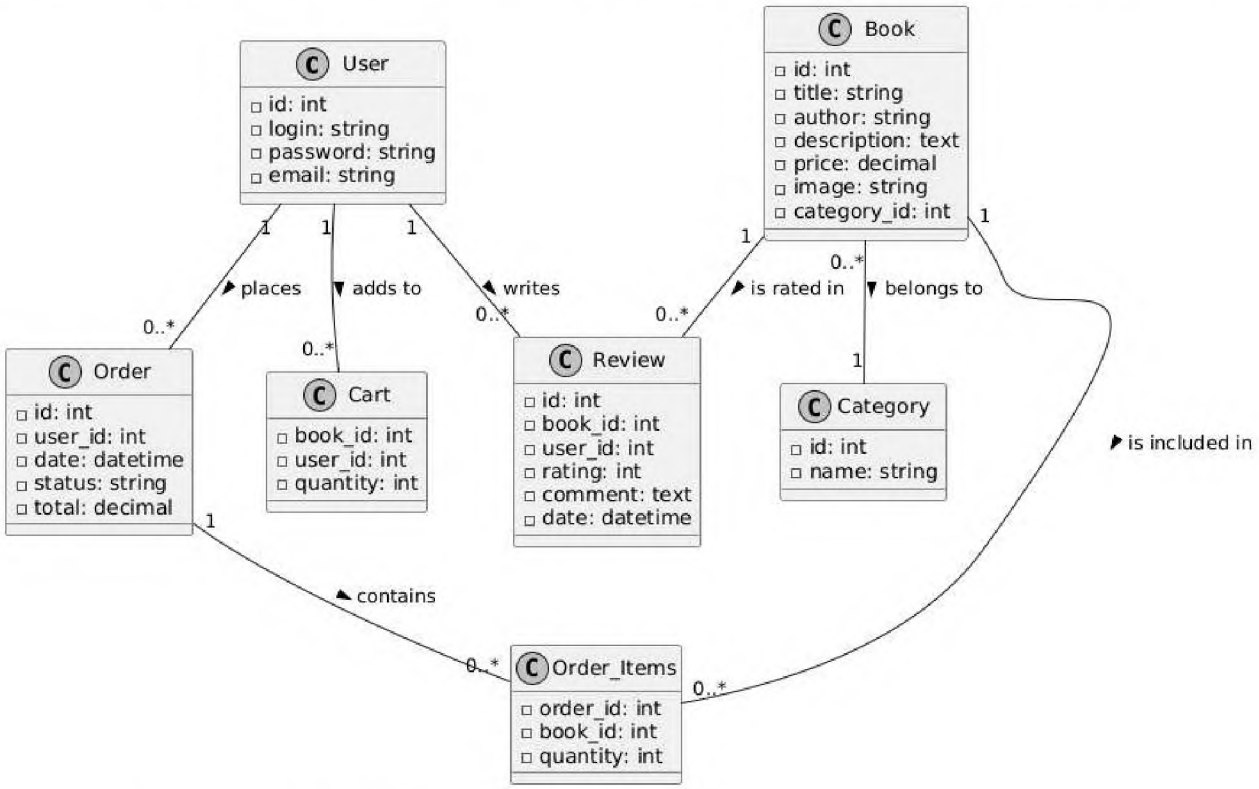

Рисунок 2.2 – EER модель БД книжкового інтернет-магазину

Зв'язки на діаграмі (рис. 2.2) відображають модель «сутність-зв'язок»:

– User – Order: 1:N (один користувач розміщує багато замовлень).
– User – Cart: 1:N (один користувач додає багато записів до кошика).
– Book – Category: N:1 (багато книг належать до однієї категорії).
– User – Review: 1:N (один користувач пише багато відгуків).
– Book – Review: 1:N (одна книга має багато відгуків).
– Order – Order_Items: 1:N (одне замовлення містить багато елементів).

– Book – Order_Items: 1:N (одна книга включається в багато елементів замовлення).

– Зв'язок N:M між Order і Book реалізовано через проміжний клас Order_Items.

Отже, розроблена структура БД включає шість основних сутностей (Користувач, Книга, Категорія, Замовлення, Кошик, Відгук) і проміжну таблицю для зв'язку замовлень із книгами. Діаграма «сутність-зв'язок» відображає зв'язки з урахуванням кардинальності, а нормалізація до 3NF забезпечує ефективність і цілісність даних. Така структура відповідає вимогам ІС книжкового інтернет-магазину, забезпечуючи швидкий доступ до даних і підтримку аналітичних функцій.

2.2 Розробка обмежень цілісності даних, індексів та тригерів

Розробка обмежень цілісності даних, індексів і тригерів є важливим етапом проєктування бази даних (БД) інформаційної системи (ІС) книжкового інтернет-магазину. Ці елементи забезпечують надійність, ефективність і цілісність даних, що критично для автоматизації процесів продажу книг, обробки замовлень і підтримки аналітичних функцій. У цьому підрозділі описано процес створення обмежень цілісності, індексів для оптимізації запитів і тригерів для автоматизації бізнес-логіки, а також представлено приклади їх реалізації в контексті структури БД, визначеної раніше.

Обмеження цілісності даних гарантують коректність і узгодженість інформації в БД. Для ІС книжкового інтернет-магазину використано три основні типи обмежень: первинні ключі (Primary Key), зовнішні ключі (Foreign Key) і обмеження унікальності (Unique). Ці обмеження застосовуються до сутностей, визначених у попередній моделі даних: User, Book, Category, Order, Cart, Review, Order_Items.

Первинні ключі (Primary Key). Кожна таблиця має унікальний ідентифікатор. Наприклад, таблиця users містить поле id типу INT з автоінкрементом, що забезпечує унікальність кожного запису. Аналогічно, таблиці books, categories, orders, reviews і order_items мають первинні ключі id. Для таблиці cart первинний ключ складається з комбінації book_id і user_id, оскільки запис у кошику унікальний для пари користувач-книга.

Зовнішні ключі (Foreign Key). Зовнішні ключі забезпечують референційну цілісність, пов'язуючи таблиці. Наприклад, у таблиці books поле category_id є зовнішнім ключем, що посилається на id у таблиці categories. Це гарантує, що книга належить до існуючої категорії. Аналогічно, у таблиці orders поле user_id посилається на id у таблиці users, а в order_items поля order_id і book_id посилаються на відповідні таблиці. Обмеження ON DELETE CASCADE застосовується до order_items, щоб автоматично видаляти елементи замовлення при видаленні замовлення.

Обмеження унікальності (Unique). У таблиці users поле email має обмеження унікальності, щоб запобігти дублюванню адрес електронної пошти. Наприклад:

```
CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  login VARCHAR(50) NOT NULL,
  password VARCHAR(255) NOT NULL,
  email VARCHAR(100) NOT NULL UNIQUE
);
```

Обмеження цілісності забезпечують, що дані залишаються узгодженими, наприклад, неможливо додати книгу до неіснуючої категорії чи оформити замовлення для неіснуючого користувача.

Індекси підвищують продуктивність БД, прискорюючи виконання запитів, особливо для пошуку та фільтрації. У контексті книжкового інтернет-магазину індекси створюються для полів, які часто використовуються в умовах WHERE, JOIN і ORDER BY.

Індекси на первинні ключі – первинні ключі (id у таблицях users, books, categories, orders, reviews, order_items) автоматично індексуються MySQL, що забезпечує швидкий доступ до записів за ідентифікатором.

Індекси на зовнішні ключі – поля category_id у books, user_id у orders, book_id і user_id у cart, order_id і book_id у order_items проіндексовано для прискорення операцій JOIN. Наприклад:

```
CREATE INDEX idx_category_id ON books(category_id);
CREATE INDEX idx_user_id ON orders(user_id)
```

Індекси для пошуку – поля title і author у таблиці books часто використовуються в пошукових запитах (наприклад, SELECT * FROM books WHERE title LIKE '%query%'). Для прискорення створюється композитний індекс:

```
CREATE INDEX idx_title_author ON books(title, author)
```

Індекси для фільтрації – поле date у таблиці orders проіндексовано для швидкого аналізу замовлень за періодами:

```
CREATE INDEX idx_order_date ON orders(date)
```

Індекси зменшують час виконання запитів, але збільшують витрати на дисковий простір і сповільнюють операції вставки чи оновлення. Тому кількість індексів обмежена лише найнеобхіднішими полями.

Тригери дозволяють автоматизувати певні дії в БД при виконанні операцій вставки, оновлення чи видалення. У книжковому інтернет-магазині тригери використовуються для підтримки бізнес-логіки, наприклад, оновлення загальної суми замовлення або логування змін.

Тригер для оновлення суми замовлення. Після вставки чи оновлення запису в таблиці order_items тригер автоматично оновлює поле total у таблиці orders, обчислюючи суму на основі цін книг і їх кількості. Наприклад:

```
DELIMITER //
CREATE TRIGGER update_order_total
AFTER INSERT ON order_items
FOR EACH ROW
```

```
BEGIN
  UPDATE orders
  SET total = (
    SELECT SUM(oi.quantity * b.price)
    FROM order_items oi
    JOIN books b ON oi.book_id = b.id
    WHERE oi.order_id = NEW.order_id
  )
  WHERE id = NEW.order_id;
END //
DELIMITER ;
```

Цей тригер забезпечує, що поле total завжди відображає актуальну суму замовлення.

Тригер для логування змін статусу замовлення. При оновленні поля status у таблиці orders тригер записує зміни до таблиці order_logs для відстеження історії. Структура таблиці order_logs:

```
CREATE TABLE order_logs (
  id INT PRIMARY KEY AUTO_INCREMENT,
  order_id INT,
  old_status VARCHAR(50),
  new_status VARCHAR(50),
  change_date DATETIME DEFAULT CURRENT_TIMESTAMP
)
```

Тригер:

```
DELIMITER //
CREATE TRIGGER log_order_status_change
BEFORE UPDATE ON orders
FOR EACH ROW
BEGIN
  IF OLD.status != NEW.status THEN
```

```
    INSERT INTO order_logs (order_id, old_status, new_status)
    VALUES (OLD.id, OLD.status, NEW.status);
  END IF;
END //
DELIMITER
```

Цей тригер допомагає адміністраторам відстежувати зміни статусів, наприклад, від «обробляється» до «відправлено».

Для перевірки обмежень, індексів і тригерів виконано тестові сценарії:

– обмеження: спроба вставки книги з неіснуючим category_id викликає помилку через зовнішній ключ;

– індекси: запит SELECT * FROM books WHERE title LIKE '%query%' виконується швидше після створення індексу idx_title_author;

– тригери: вставка запису в order_items автоматично оновлює total у orders, а зміна статусу замовлення додає запис до order_logs.

Розроблені обмеження цілісності (первинні та зовнішні ключі, унікальність), індекси та тригери забезпечують надійність, продуктивність і автоматизацію в БД книжкового інтернет-магазину. Первинні та зовнішні ключі гарантують узгодженість даних, індекси прискорюють пошукові та аналітичні запити, а тригери автоматизують бізнес-логіку, таку як обчислення суми замовлення та логування змін. Ці елементи відповідають вимогам ефективності та безпеки, підтримуючи швидкий доступ до даних і масштабування системи.

2.3 Розробка правил безпеки для бази даних книжкового інтернет-магазину

Безпека бази даних (БД) є критично важливим аспектом функціонування інформаційної системи (ІС) книжкового інтернет-магазину, оскільки вона містить конфіденційні дані користувачів, інформацію про замовлення та фінансові транзакції. Недостатній захист може призвести до витоку даних,

фінансових втрат і втрати довіри клієнтів. У цьому підрозділі розглядаються правила безпеки для БД, включаючи захист від несанкціонованого доступу, забезпечення цілісності даних, шифрування, захист від атак і резервне копіювання. Правила розроблені з урахуванням структури БД, описаної раніше, яка включає сутності User, Book, Category, Order, Cart, Review і Order_Items.

Для запобігання несанкціонованому доступу до БД застосовуються заходи з управління доступом і автентифікацією: У MySQL створюються окремі користувачі з різними рівнями привілеїв. Наприклад, для адміністраторів системи призначається користувач із повним доступом до всіх таблиць (SELECT, INSERT, UPDATE, DELETE), тоді як для клієнтської частини створюється користувач із обмеженими правами (лише SELECT для таблиць books, categories і reviews). Приклад створення користувача:

```
CREATE USER 'app_user'@'localhost' IDENTIFIED BY 'secure_password';
GRANT SELECT ON bookstore.books TO 'app_user'@'localhost';
GRANT SELECT ON bookstore.categories TO 'app_user'@'localhost'
```

Доступ до БД дозволяється лише з певних IP-адрес, наприклад, серверів додатку. Це налаштовується в конфігурації MySQL:

```
GRANT ALL PRIVILEGES ON bookstore.* TO 'admin'@'192.168.1.100'
IDENTIFIED BY 'admin_password'
```

Паролі для всіх користувачів БД мають бути складними (не менше 12 символів, включаючи літери, цифри та спеціальні символи). Паролі змінюються кожні 90 днів, а їх зберігання забезпечується в зашифрованому вигляді через механізми MySQL.

SQL-ін’єкції є однією з найпоширеніших загроз для вебдодатків. Для захисту від них застосовуються такі заходи:

1. Підготовлені запити – усі запити до БД, що містять дані від користувачів (наприклад, пошуковий запит чи дані форми), виконуються через підготовлені запити в Laravel Eloquent ORM. Наприклад, для пошуку книги:

```
$books = Book::where('title', 'LIKE', '%' . $searchQuery . '%')->get();
```

Це запобігає введенню шкідливого коду, такого як ' OR 1=1 --.

2. Екранування даних – усі вхідні дані, отримані від користувачів (наприклад, коментарі в таблиці reviews), екрануються перед збереженням у БД, щоб уникнути XSS-атак. Laravel автоматично екранує дані через middleware.

3. Обмеження привілеїв для запитів – користувач БД для клієнтської частини не має прав на виконання операцій DROP або ALTER, що знижує ризик модифікації структури БД у разі компрометації.

Для захисту конфіденційних даних, таких як паролі користувачів і фінансові дані, застосовується шифрування. Паролі в таблиці users зберігаються в зашифрованому вигляді за допомогою алгоритму bcrypt, який є стійким до атак грубої сили. Приклад у Laravel:

```
$user = new User();
$user->password = bcrypt('user_password');
$user->save();
```

Усі з'єднання між сервером додатку та БД захищені протоколом TLS. У MySQL це налаштовується через параметри SSL:

```
ssl-ca=ca.pem
ssl-cert=server-cert.pem
ssl-key=server-key.pem
```

Хоча фінансові дані (наприклад, номери карт) не зберігаються в БД, а обробляються зовнішніми платіжними системами (наприклад, Stripe), тимчасові токени транзакцій шифруються за допомогою AES-256 перед збереженням у таблиці orders.

Для забезпечення надійності та захисту від втрати даних розроблено стратегію резервного копіювання. Щоденні резервні копії БД створюються за допомогою утиліти mysqldump. Наприклад: mysqldump -u admin -p bookstore > backup_$(date +%Y%m%d).sql

Копії зберігаються на окремому захищеному сервері з доступом лише для адміністраторів. Для зменшення обсягу даних застосовується інкрементне копіювання, яке зберігає лише зміни з моменту останньої повної копії. Це налаштовується через інструменти, такі як Percona XtraBackup. Щомісяця

виконується тестування відновлення БД із резервної копії, щоб переконатися в її цілісності: mysql -u admin -p bookstore < backup_20250401.sql

Для виявлення потенційних загроз і аналізу інцидентів реалізовано журналювання та моніторинг. У MySQL увімкнено журнал повільних запитів (slow_query_log), щоб виявляти неефективні запити, які можуть бути використані для DoS-атак. Наприклад:

SET GLOBAL slow_query_log = 'ON';

SET GLOBAL long_query_time = 2

Усі спроби входу до БД записуються в журнал (general_log), що дозволяє відстежувати підозрілі дії. Лог-файли аналізуються за допомогою інструментів, таких як Logwatch. Для відстеження змін у критичних таблицях, таких як orders, створено тригер, який записує оновлення статусу до таблиці order_logs (описано в попередньому підрозділі). Це забезпечує аудит дій адміністраторів.

Для підтримки цілісності даних застосовуються обмеження, описані раніше (первинні та зовнішні ключі, унікальність). Операції, що змінюють кілька таблиць (наприклад, створення замовлення та додавання елементів до order_items), виконуються в межах транзакцій для забезпечення атомарності. Приклад у Laravel:

```
DB::transaction(function () {
  $order = Order::create(['user_id' => $userId, 'status' => 'pending']);
  OrderItem::create(['order_id' => $order->id, 'book_id' => $bookId, 'quantity' => 1]);
})
```

Перед вставкою даних до таблиць виконується валідація на стороні сервера (Laravel) для забезпечення коректності формату (наприклад, позитивна ціна в books). Ефективність розроблених правил оцінюється за такими критеріями:

– захист від витоків – шифрування паролів і TLS забезпечують конфіденційність даних.

– стійкість до атак – підготовлені запити та обмеження привілеїв знижують ризик SQL-ін’єкцій.

– надійність – регулярне резервне копіювання та транзакції гарантують збереження даних.

Тестування безпеки включає спроби SQL-ін'єкцій через форми пошуку (усі відхилені завдяки підготовленим запитам); тестування відновлення БД із резервної копії (успішне за 5 хвилин); аналіз логів для виявлення несанкціонованих спроб доступу (жодних за тестового періоду).

Розроблені правила безпеки для БД книжкового інтернет-магазину забезпечують захист від несанкціонованого доступу, атак, втрати даних і порушення цілісності. Управління доступом, шифрування, підготовлені запити, резервне копіювання та журналювання створюють надійне середовище для роботи ІС. Ці заходи відповідають вимогам безпеки електронної комерції, гарантуючи конфіденційність даних користувачів і стабільність системи.

Таким чином, у розділі розроблено архітектуру інформаційної системи книжкового інтернет-магазину, зокрема структуру бази даних, обмеження цілісності, індекси, тригери та правила безпеки. На основі аналізу вимог створено концептуальну модель БД у нотації Чена, яка включає сутності (Користувач, Книга, Категорія тощо), їх атрибути та зв'язки з урахуванням принципів нормалізації до 3NF. Діаграма «сутність-зв'язок» візуалізує взаємодію між компонентами системи, що забезпечує ефективне зберігання та обробку даних.

Також запропоновано механізми для підвищення надійності та продуктивності системи: обмеження цілісності (первинні та зовнішні ключі), індекси для прискорення пошуку, тригери для автоматизації бізнес-логіки (наприклад, оновлення суми замовлення). Розроблено правила безпеки, включаючи управління доступом, шифрування даних, захист від SQL-ін'єкцій та стратегію резервного копіювання. Ці рішення відповідають вимогам масштабованості, безпеки та ефективності, що є критичними для стабільної роботи інтернет-магазину.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КНИЖКОВОГО ІНТЕРНЕТ-МАГАЗИНУ

3.1 Реалізація бази даних книжкового інтернет-магазину за допомогою phpMyAdmin

Реалізація бази даних (БД) є важливим етапом створення інформаційної системи (ІС) книжкового інтернет-магазину, який забезпечує зберігання, управління та обробку даних про користувачів, книги, замовлення та інші сутності. Для реалізації використано інструмент phpMyAdmin – вебінтерфейс для адміністрування MySQL, що дозволяє зручно створювати таблиці, налаштовувати обмеження, індекси, тригери та керувати даними. У цьому підрозділі описано процес створення БД, її фізичну модель, нові особливості, що розширюють функціональність порівняно з концептуальною моделлю, а також продемонстровано приклади використання phpMyAdmin для реалізації та тестування.

Для реалізації БД використано локальний сервер із встановленим стеком OpenServer, який включає MySQL 8.4 і phpMyAdmin. Початковим етапом було створення бази даних із назвою bookstore_ua із кодуванням utf8mb4_unicode_ci, що підтримує україномовний контент і забезпечує коректне зберігання символів Unicode. У phpMyAdmin виконано наступні дії:

1. Увійти в інтерфейс phpMyAdmin через браузер за адресою localhost/phpmyadmin.

2. Створити нову БД: на вкладці Databases увести назву bookstore_ua, обрати кодування utf8mb4_unicode_ci і натиснути Create.

3. Вибрати створену БД для подальшої роботи.

Цей етап забезпечив основу для створення таблиць і налаштування структури.

Фізична модель БД реалізує концептуальну модель, описану раніше, з урахуванням особливостей MySQL і phpMyAdmin. Вона включає вісім таблиць: users, categories, books, orders, order_items, cart, reviews і order_logs. Кожна таблиця має чітко визначені атрибути, типи даних, обмеження цілісності, індекси та тригери. У порівнянні з концептуальною моделлю фізична модель містить нові особливості, які розширюють функціональність:

– додаткові атрибути для управління ролями, запасами та логуванням;

– обмеження для забезпечення коректності даних (наприклад, CHECK для оцінок);

– тригери для автоматизації оновлення та логування;

– РОЛІ користувачів БД для безпеки.

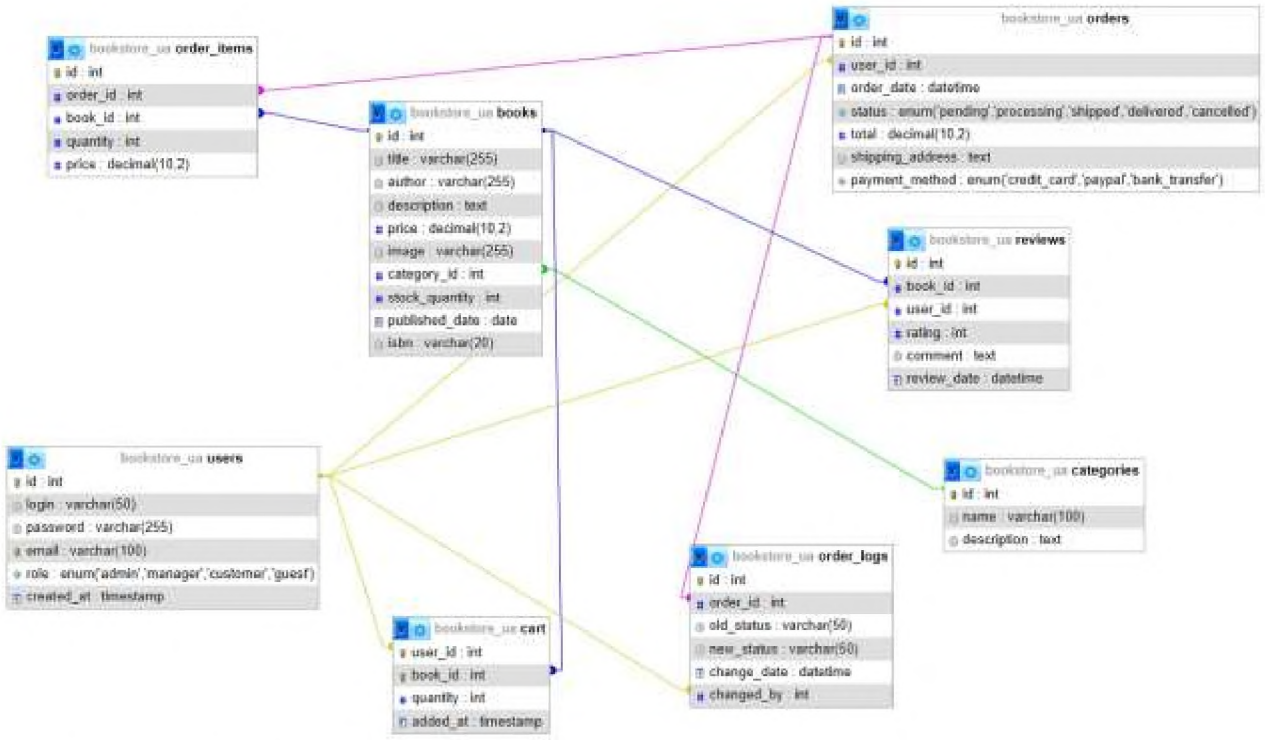

Рисунок 3.1 – Фізична модель бази даних книжкового інтернет-магазину

Фізична модель (рис. 3.1) створена в phpMyAdmin через вкладку Designer, яка дозволяє візуально визначити таблиці, зв’язки та обмеження. Для кожної таблиці вказано двигун InnoDB, який підтримує транзакції та референційну цілісність.

Таблиці створювалися через вкладку Structure у phpMyAdmin, де для кожної таблиці визначалися атрибути, типи даних і обмеження. У таблиці 3.1 наведено ключові особливості реалізації таблиць БД, які розширюють концептуальну модель.

Таблиця 3.1 – Нові атрибути у фізичній моделі порівняно з концептуальною

Таблиця	Нові атрибути	Призначення
users	role, created_at	Розмежування прав, фіксація дати реєстрації
categories	description	Детальний опис категорії
books	stock_quantity, published_date, isbn	Управління запасами, інформація про видання
orders	shipping_address, payment_method	Обробка доставки та оплати
order_items	price	Фіксація ціни на момент замовлення
cart	added_at	Аналіз часу додавання до кошика
reviews	CHECK (rating BETWEEN 1 AND 5)	Контроль коректності оцінок
order_logs	Нова таблиця	Аудит змін статусу замовлень

Таблиця users: атрибути: id (INT, AUTO_INCREMENT), login (VARCHAR(50)), password (VARCHAR(255)), email (VARCHAR(100), UNIQUE), role (ENUM: admin, manager, customer, guest), created_at (TIMESTAMP). Нова особливість: Атрибут role дозволяє розмежовувати права доступу (адміністратор, менеджер, клієнт, гість). Наприклад, адміністратор може редагувати книги, а клієнт – лише переглядати каталог. Атрибут created_at фіксує дату реєстрації.

Таблиця categories: атрибути: id (INT, AUTO_INCREMENT), name (VARCHAR(100)), description (TEXT). Нова особливість: Додано поле description для зберігання детального опису категорії, що полегшує адміністрування та відображення інформації для користувачів.

Таблиця books: атрибути: id (INT, AUTO_INCREMENT), title (VARCHAR(255)), author (VARCHAR(255)), description (TEXT), price (DECIMAL(10,2)), image (VARCHAR(255)), category_id (INT, FK), stock_quantity

(INT), published_date (DATE), isbn (VARCHAR(20)). Нова особливість: Атрибути stock_quantity, published_date та isbn дозволяють відстежувати наявність книг, дату видання та унікальний ідентифікатор. Це підтримує управління запасами та пошук за ISBN.

Таблиця orders: атрибути: id (INT, AUTO_INCREMENT), user_id (INT, FK), order_date (DATETIME), status (ENUM: pending, processing, shipped, delivered, cancelled), total (DECIMAL(10,2)), shipping_address (TEXT), payment_method (ENUM: credit_card, paypal, bank_transfer). Нова особливість: Додані поля shipping_address і payment_method для зберігання адреси доставки та способу оплати, що є необхідним для обробки замовлень.

Таблиця order_items: атрибути: id (INT, AUTO_INCREMENT), order_id (INT, FK), book_id (INT, FK), quantity (INT), price (DECIMAL(10,2)). Нова особливість: Поле price фіксує ціну книги на момент замовлення, що запобігає невідповідностям при зміні ціни книги.

Таблиця cart: атрибути: user_id (INT, FK), book_id (INT, FK), quantity (INT), added_at (TIMESTAMP). Нова особливість: Поле added_at дозволяє відстежувати час додавання книги до кошика, що корисно для аналізу поведінки користувачів.

Таблиця reviews: атрибути: id (INT, AUTO_INCREMENT), book_id (INT, FK), user_id (INT, FK), rating (INT, CHECK: 1–5), comment (TEXT), review_date (DATETIME). Нова особливість: Обмеження CHECK (rating BETWEEN 1 AND 5) гарантує коректність оцінок, що підвищує надійність даних.

Таблиця order_logs: атрибути: id (INT, AUTO_INCREMENT), order_id (INT, FK), old_status (VARCHAR(50)), new_status (VARCHAR(50)), change_date (DATETIME), changed_by (INT, FK). Нова особливість: Таблиця підтримує аудит змін статусу замовлень, а поле changed_by вказує на користувача, який вніс зміни (наприклад, менеджер чи адміністратор).

Обмеження цілісності та індекси налаштовувалися через вкладку Structure у phpMyAdmin, а також за допомогою SQL-запитів у вкладці SQL. Усі таблиці, крім cart, мають первинний ключ id (INT, AUTO_INCREMENT). Для cart

первинний ключ – комбінація user_id і book_id. У phpMyAdmin первинні ключі встановлювалися через прапорець PRIMARY у налаштуваннях поля.

Зовнішні ключі були налаштовані через вкладку Relation view у phpMyAdmin після активації InnoDB (рис. 3.2). Наприклад, books.category_id пов'язаний із categories.id із обмеженням ON DELETE RESTRICT, що запобігає видаленню категорії, якщо вона використовується. Для order_items, cart, reviews і order_logs використано ON DELETE CASCADE, щоб автоматично видаляти залежні записи.

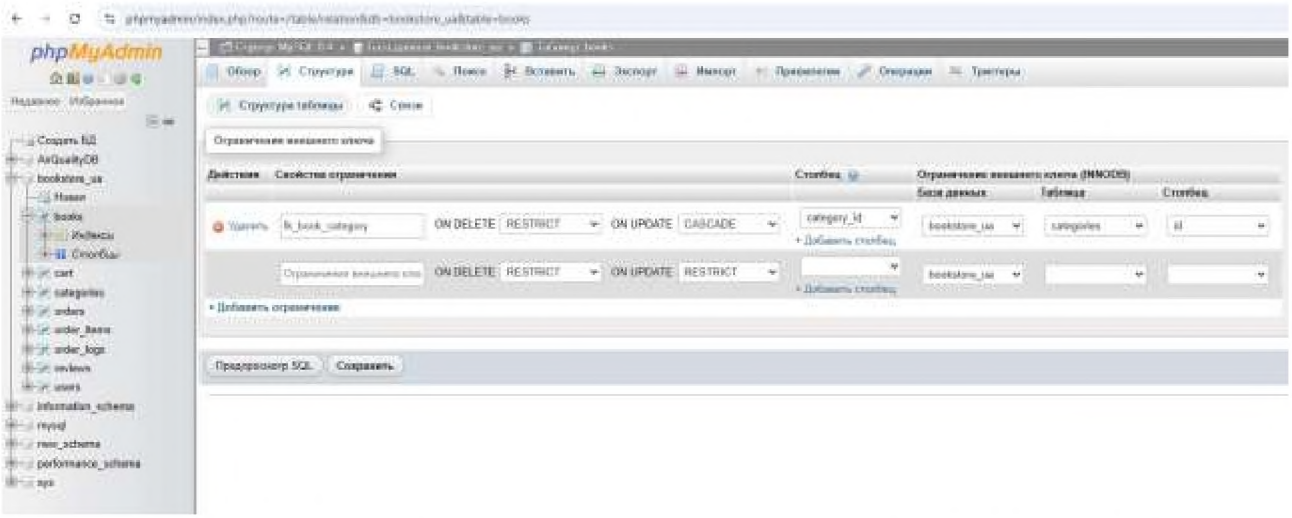

Рисунок 3.2 – Налаштування зовнішнього ключа в phpMyAdmin

Поле email у users позначено як UNIQUE через прапорець у phpMyAdmin. Обмеження CHECK для reviews.rating додано через SQL-запит:

ALTER TABLE reviews ADD CONSTRAINT check_rating CHECK (rating BETWEEN 1 AND 5)

Індекси на зовнішні ключі (наприклад, idx_category_id у books) створено через вкладку Indexes у phpMyAdmin. Композитний індекс idx_title_author для books створено через SQL:

CREATE INDEX idx_title_author ON books(title, author)

Тригери створено через вкладку Triggers у phpMyAdmin, що дозволяє визначити SQL-код для автоматизації. Тригер update_order_total оновлює поле

total у orders після вставки запису в order_items. Код тригера введено в phpMyAdmin:

```
DELIMITER //
CREATE TRIGGER update_order_total
AFTER INSERT ON order_items
FOR EACH ROW
BEGIN
 UPDATE orders o
 SET o.total = (
  SELECT SUM(oi.quantity * oi.price)
  FROM order_items oi
  WHERE oi.order_id = NEW.order_id
 )
 WHERE o.id = NEW.order_id;
END //
DELIMITER
```

Тригер update_order_total_update оновлює total після зміни запису в order_items, що забезпечує актуальність суми при зміні кількості чи ціни. Тригер log_order_status_change логує зміни статусу в order_logs, додаючи поле changed_by для вказівки користувача, який вніс зміну. Це підтримує аудит дій менеджерів. Безпека БД налаштовувалася через вкладку Privileges у phpMyAdmin. Нові особливості включають створення трьох ролей користувачів із різними привілеями:

– bookstore_admin: повний доступ до всіх таблиць (ALL PRIVILEGES);

– bookstore_manager: дозволено SELECT, INSERT, UPDATE, заборонено DELETE;

– bookstore_customer: лише SELECT для books, categories, reviews.

Приклад створення користувача в phpMyAdmin:

```
CREATE USER 'bookstore_admin'@'localhost' IDENTIFIED BY
'StrongAdminPassword123!';
```

GRANT ALL PRIVILEGES ON bookstore_ua.* TO 'bookstore_admin'@'localhost';

FLUSH PRIVILEGES

Паролі відповідають вимогам складності (не менше 12 символів, літери, цифри, спеціальні символи). Для тестування безпеки виконано спроби доступу з обмеженим користувачем bookstore_customer, який не зміг модифікувати таблиці.

Для перевірки працездатності БД використано вкладку SQL у phpMyAdmin для вставки тестових даних і виконання запитів. Наприклад, вставка книги:

INSERT INTO books (title, author, price, category_id, stock_quantity, isbn)

VALUES ('Тіні забутих предків', 'Михайло Коцюбинський', 150.00, 1, 35, '9789662355789')

Перевірка тригера update_order_total виконана після вставки запису в order_items поле total у orders оновлювалося автоматично. Результати тестування ключових операцій наведено у таблиці 3.2. Тестові дані включають україномовні книги, категорії, замовлення та відгуки, що підтверджує підтримку локалізації. Загалом вставлено 15 книг, 15 замовлень, 16 відгуків і 5 записів у кошик.

Таблиця 3.2 – Результати тестування ключових операцій

Операція	Очікуваний результат	Фактичний результат
Вставка книги з неіснуючим category_id	Помилка через fk_book_category	Помилка MySQL: Foreign key violation
Пошук за назвою книги	Швидке виконання завдяки idx_title_author	Час виконання: <0.01 сек
Оновлення order_items	Оновлення total у orders	Поле total оновлено коректно
Зміна статусу замовлення	Запис у order_logs із changed_by	Запис додано, changed_by коректний

Реалізація бази даних книжкового інтернет-магазину за допомогою phpMyAdmin дозволила створити фізичну модель із вісьмома таблицями, що відповідає концептуальній моделі та розширює її новими можливостями.

Додаткові атрибути (role, stock_quantity, shipping_address), таблиця order_logs, обмеження CHECK і тригер update_order_total_update підвищують функціональність і надійність. Налаштування безпеки через ролі користувачів і тестування підтвердили стабільність системи. Використання phpMyAdmin забезпечило зручність створення та адміністрування БД, а підтримка україномовного контенту відповідає вимогам локалізації. Фізична модель готова до інтеграції з серверною частиною ІС, забезпечуючи ефективне управління даними.

3.2 Розробка функціональних компонентів книжкового інтернет-магазину

Розробка книжкового інтернет-магазину передбачала створення повнофункціонального веб-додатку з інтуїтивно зрозумілим інтерфейсом, адаптивним дизайном та серверною логікою для обробки запитів користувачів. У цьому підрозділі розглянуто структуру проєкту, особливості реалізації клієнтської та серверної частин, а також впровадження функціоналу пошуку, фільтрації за жанрами та кошика. Проєкт було реалізовано з використанням сучасних веб-технологій: HTML, CSS, JavaScript для клієнтської частини, PHP і MySQL для серверної частини та зв'язку з базою даних.

Проєкт має модульну структуру, що забезпечує зручність розробки та підтримки. Основні компоненти організовано в окремі директорії:

1. Корінь проєкту:

– index.php – точка входу, яка перенаправляє користувача на сторінку каталогу книг (/controllers/books.php);

– db.php – містить функції для роботи з базою даних (наприклад, getBooks(), getCart(), createOrder());

– config.php – конфігураційний файл із налаштуваннями підключення до бази даних (константи DB_HOST, DB_NAME тощо);

– script.js – JavaScript-код для асинхронного додавання книг до кошика;

– style.css – базові стилі для сторінки;

– bookstore_ua.sql – SQL-дамп бази даних із таблицями та тестовими даними;

– README.md – інструкція з встановлення та запуску проєкту;

2. Папка controllers:

– books.php – обробляє запити, пов'язані з каталогом книг (список книг, перегляд окремої книги, додавання відгуків);

– cart.php – відповідає за управління кошиком (додавання, оновлення, видалення товарів);

– orders.php – обробляє замовлення (перегляд історії, оформлення замовлення);

– auth.php – реалізує авторизацію та вихід користувача;

3. Папка templates:

– layout.php – базовий шаблон, який містить загальну розмітку сторінки (навігаційне меню, контейнер для контенту);

– books.php – шаблон для відображення каталогу книг і сторінки окремої книги;

– cart.php – шаблон для кошика;

– orders.php – шаблон для історії замовлень і оформлення замовлення;

– auth.php – шаблон для сторінки авторизації;

4. Папка images містить зображення обкладинок книг із назвами, що відповідають ISBN (наприклад, 9789662355789.jpg).

Структура проєкту дозволяє легко масштабувати функціонал, додаючи нові контролери та шаблони за потреби. Клієнтська частина проєкту розроблена з акцентом на адаптивність і зручність користувача. Для розмітки використано HTML, стилі реалізовано з використанням Tailwind CSS (підключено через CDN), а JavaScript забезпечує інтерактивність, зокрема асинхронне додавання книг до кошика.

Основний шаблон layout.php містить навігаційне меню, яке адаптується до різних розмірів екранів. На десктопах меню відображається горизонтально, а на мобільних пристроях – як гамбургер-меню, що розгортається при натисканні. Навігаційне меню включає пункти: «Книжковий магазин» (головна сторінка),

«Книги» (каталог), «Кошик», «Замовлення» та «Вхід/Вихід» (залежно від статусу авторизації). Код меню виглядає так:

```
<nav class="bg-white shadow-md">
  <div class="container mx-auto px-4 py-3 flex justify-between items-center">
    <a href="/" class="text-xl font-bold text-gray-800">Книжковий магазин</a>
    <div class="lg:hidden">
      <label for="menu-toggle" class="cursor-pointer">
        <svg class="w-6 h-6" fill="none" stroke="currentColor" viewBox="0 0 24 24">
          <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M4 6h16M4 12h16M4 18h16"></path>
        </svg>
      </label>
    </div>
    <input type="checkbox" id="menu-toggle" class="hidden">
    <div class="mobile-menu hidden lg:flex lg:items-center lg:space-x-6 absolute lg:static top-14 left-0 w-full lg:w-auto bg-white lg:bg-transparent shadow-md lg:shadow-none z-10">
      <div class="flex flex-col lg:flex-row lg:space-x-6 p-4 lg:p-0">
        <a href="/controllers/books.php" class="block lg:inline-block text-gray-600 hover:text-blue-600 py-2 lg:py-0">Книги</a>
        <a href="/controllers/cart.php" class="block lg:inline-block text-gray-600 hover:text-blue-600 py-2 lg:py-0">Кошик</a>
        <a href="/controllers/orders.php" class="block lg:inline-block text-gray-600 hover:text-blue-600 py-2 lg:py-0">Замовлення</a>
        <?php if (isLoggedIn()): ?>
          <a href="/controllers/auth.php?action=logout" class="block lg:inline-block text-gray-600 hover:text-blue-600 py-2 lg:py-0">Вихід</a>
        <?php else: ?>
          <a href="/controllers/auth.php?action=login" class="block lg:inline-block text-gray-600 hover:text-blue-600 py-2 lg:py-0">Вхід</a>
        <?php endif; ?>
```

```
        </div>
      </div>
    </div>
</nav>
```

Адаптивність меню забезпечується класами Tailwind CSS: lg:hidden ховає гамбургер-іконку на десктопах, а hidden lg:flex показує пункти меню горизонтально на великих екранах. На мобільних пристроях меню розгортається за допомогою CSS-селектора #menu-toggle:checked ~ .mobile-menu, що змінює display: block при активації checkbox.

Шаблон templates/books.php відповідає за відображення каталогу книг. Список книг представлено у вигляді сітки, яка адаптується до розміру екрану: 1 колонка на мобільних, 2 на планшетах, 3 на десктопах (grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3). Кожна книга відображається як картка з обкладинкою, назвою, автором, ціною та кнопками «Деталі» і «Додати до кошика». Якщо обкладинка відсутня, відображається заповнювач із текстом «Немає зображення». Рис. 3.3 ілюструє вигляд каталогу книг на десктопі.

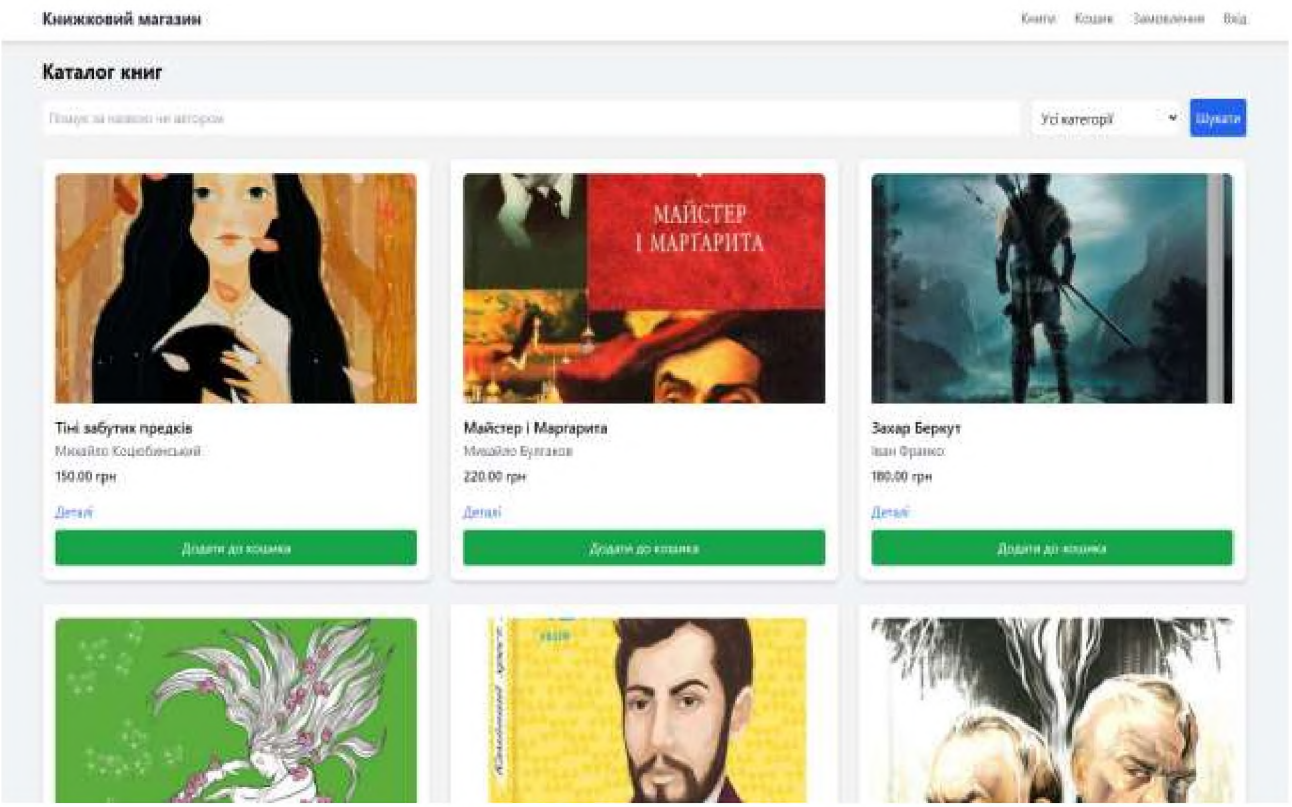

Рисунок 3.3 – Каталог книг у книжковому інтернет-магазині

На сторінці окремої книги (action=view у books.php) зображення та інформація розташовані адаптивно: вертикально на мобільних і горизонтально на десктопах (flex flex-col lg:flex-row). Користувачі можуть залишити відгук, якщо вони авторизовані. Форма відгуків включає текстове поле для коментаря та вибір рейтингу від 1 до 5.

Кнопки «Додати до кошика» мають клас add-to-cart і атрибут data-id із ідентифікатором книги. При натисканні JavaScript відправляє POST-запит до cart.php, додаючи книгу до кошика.

Серверна частина розроблена з використанням PHP і MySQL. PHP відповідає за обробку запитів, взаємодію з базою даних і генерацію HTML через шаблони. MySQL забезпечує зберігання даних про книги, користувачів, замовлення, кошик і відгуки.

Контролери в папці controllers обробляють запити користувачів і передають дані в шаблони. Наприклад, books.php обробляє три дії:

– list – відображає список книг із можливістю пошуку та фільтрації;
– view – показує сторінку окремої книги з відгуками;
– review – додає відгук до книги.

Функціонал кошика реалізовано через контролер cart.php і таблицю cart у базі даних. Користувач може додавати книги до кошика, змінювати кількість або видаляти товари. При оформленні замовлення (orders.php) створюється запис у таблиці orders, а товари з кошика переносяться в таблицю order_items.

Функціонал пошуку та фільтрації реалізовано в templates/books.php і controllers/books.php. Користувач може шукати книги за назвою або автором через текстове поле та фільтрувати за категоріями через випадаючий список. Пошук працює шляхом передачі параметра search у контролер, де він використовується в SQL-запиті. Фільтрація за категоріями додає умову AND category_id = :category до запиту, якщо обрано категорію.

Зображення обкладинок книг додано до проєкту вручну, і збережено в папці /images із назвами, що відповідають ISBN книг (наприклад, 9789662355789.jpg). Шаблон books.php відображає зображення, якщо поле image не порожнє, або заповнювач, якщо зображення відсутнє.

3.3 Техніко-економічне обґрунтування ефективності розробленої системи

Розробка книжкового інтернет-магазину передбачала кілька етапів: аналіз вимог, проєктування бази даних, розробка клієнтської та серверної частин, тестування та розгортання. Оскільки проєкт виконувався в рамках дипломної роботи, основні витрати пов'язані з робочим часом розробника, використанням програмного забезпечення та хостингу. У таблиці 3.4 наведено детальний розподіл витрат.

Таблиця 3.4 – Витрати на розробку книжкового інтернет-магазину

Стаття витрат	Опис	Вартість, грн
Робочий час розробника	200 годин × 150 грн/год (середня ставка фрилансера-початківця)	30000
Програмне забезпечення	OpenServer (безкоштовно), MySQL (безкоштовно), PHP (безкоштовно)	0
Хостинг і домен (на 1 рік)	Хостинг – 1200 грн/р., домен bookstore.ua – 300 грн/р.	1500
Технічне обладнання	Використання власного ПК (без додаткових витрат)	0
Додаткові витрати (електроенергія тощо)	Орієнтовно 50 грн/міс × 4 місяці розробки	200
Всього		31700

Для оцінки робочого часу використано середню ставку фрилансера-початківця в Україні (150 грн/год), що є типовим показником для студентських проєктів [27]. Хостинг обрано з розрахунку на базовий тариф для невеликих веб-додатків, а доменне ім'я bookstore.ua оцінено за середньою ринковою ціною.

Для оцінки доходів було проаналізовано потенційний ринок та середні показники продажів книг онлайн в Україні. За даними Асоціації видавців України (2024), середня ціна книги становить 170 грн, а середній онлайн-магазин книг продає близько 50 – 100 книг на місяць у перший рік роботи [27].

Припустимо, що книжковий інтернет-магазин продаватиме 75 книг на місяць із середньою ціною 170 грн. Тоді місячний дохід складе 75×170 = 12750 грн/міс.

Річний дохід за такого обсягу продажів: 12750×12 = 153000 грн/р.

Проте необхідно врахувати операційні витрати, такі як хостинг, оновлення контенту та маркетинг. Операційні витрати на рік оцінено в розмірі 18 000 грн (хостинг – 1200 грн/р., домен – 300 грн/р., маркетинг – 500 грн/міс × 12 = = 6000 грн, оновлення контенту – 500 грн/міс × 12 = 6000 грн, резерв на непередбачені витрати – 4500 грн).

Таким чином, чистий прибуток за рік: 153000 – 18000 = 135000 грн/р.

Економічна ефективність системи оцінюється через показник терміну окупності, який показує, скільки часу знадобиться для покриття початкових витрат за рахунок чистого прибутку [28, 29]. Початкові витрати на розробку становлять 31 700 грн, а чистий прибуток на місяць: 135000/12=11250 грн/міс.

Термін окупності розраховується як: 31700/11250 ≈ 2,82 місяці. Отже, система окупиться приблизно за 3 місяці після запуску. Для наочності на рис. 3.4 зображено графік накопичення чистого прибутку та точку окупності.

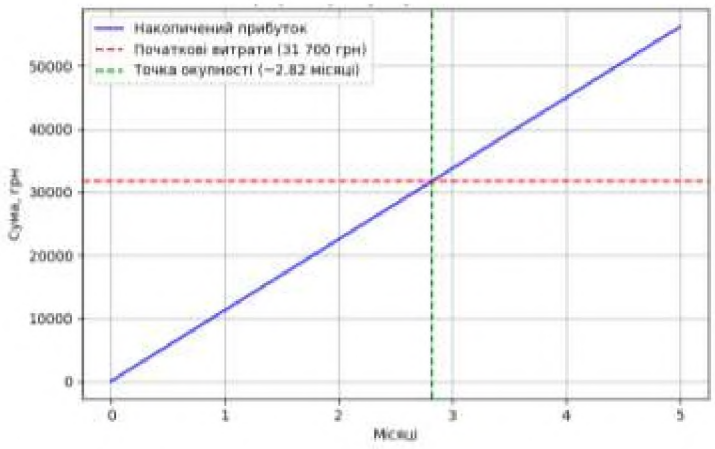

Рисунок 3.4 – Графік терміну окупності системи

Техніко-економічний аналіз показав, що розроблена система є ефективною з точки зору витрат і прибутковості. Технічні переваги, такі як адаптивність, швидкість і безпека, роблять систему конкурентоспроможною, а соціально-економічні вигоди сприяють її довгостроковій цінності.

ВИСНОВКИ

У результаті виконання дипломної роботи досягнуто поставленої мети та виконано всі завдання, спрямовані на створення інформаційної системи книжкового інтернет-магазину. Було проведено комплексний аналіз теоретичних основ, сучасних технологій, вимог та аналогів, що дозволило розробити повнофункціональний веб-додаток із зручним інтерфейсом, ефективною базою даних і високою економічною ефективністю.

На етапі теоретичного аналізу досліджено поняття інформаційної системи та її роль в електронній комерції. Визначено, що ІС є ключовим елементом автоматизації бізнес-процесів, забезпечуючи каталогізацію товарів, обробку замовлень, безпеку транзакцій і персоналізацію. Проаналізовано сучасні технології створення веб-додатків, зокрема клієнтські (HTML, CSS, JavaScript) і серверні (PHP, Node.js, Python) технології, а також СУБД (MySQL, PostgreSQL, MongoDB). Обрано оптимальний набір інструментів: HTML, CSS (Tailwind), JavaScript для клієнтської частини, PHP для серверної частини та MySQL для бази даних, що забезпечило швидкість розробки, адаптивність і масштабованість системи.

Проведено аналіз вимог до ІС книжкових інтернет-магазинів, визначено функціональні (каталог, пошук, кошик, авторизація, відгуки) та нефункціональні (продуктивність, безпека, зручність) вимоги. Досліджено аналоги (Amazon Books, Yakaboo, Book Depository), що дозволило врахувати їхні сильні сторони (зручний інтерфейс, рекомендаційні системи) та уникнути недоліків (обмежена масштабованість, слабка персоналізація). На основі аналізу зроблено висновок, що розроблена система має конкурентні переваги завдяки локалізації, адаптивності та економічній ефективності.

Розроблено структуру бази даних із використанням концептуальної моделі в нотації Чена та фізичної моделі в phpMyAdmin. БД включає вісім таблиць (users, books, categories, orders, order_items, cart, reviews, order_logs), які підтримують управління каталогом, замовленнями, кошиком і відгуками.

Додано нові атрибути (role, stock_quantity, shipping_address), що розширюють функціональність, а також реалізовано обмеження цілісності, індекси та тригери для автоматизації (наприклад, оновлення суми замовлення). Впроваджено правила безпеки: управління доступом, шифрування паролів (bcrypt), захист від SQL-ін'єкцій, резервне копіювання та журналювання. Це забезпечило надійність, безпеку та швидкий доступ до даних.

Розроблено функціональні компоненти веб-додатку: адаптивний інтерфейс (HTML, Tailwind CSS), інтерактивність (JavaScript для асинхронного додавання до кошика), серверну логіку (PHP, MySQL). Реалізовано каталог книг із пошуком і фільтрацією, кошик, оформлення замовлень і авторизацію. Модульна структура проєкту (папки controllers, templates) спрощує підтримку та масштабування. Адаптивний дизайн забезпечує коректне відображення на всіх пристроях, а індексація в БД прискорює запити.

Техніко-економічний аналіз показав високу ефективність системи: початкові витрати (31700 грн) окупаються за 3 місяці за прогнозованого продажу 75 книг на місяць. Технічні переваги (швидкість, безпека, адаптивність) і соціальні вигоди (підтримка української літератури, доступність для віддалених регіонів) роблять систему цінним інструментом для електронної комерції.

Таким чином, усі поставлені завдання виконано в повному обсязі. Розроблена ІС книжкового інтернет-магазину відповідає сучасним вимогам, забезпечуючи зручність, безпеку та економічну ефективність. Напрямками подальших досліджень є вдосконалення рекомендаційної системи (наприклад, на основі алгоритмів машинного навчання), інтеграція платіжних систем (LiqPay, Stripe) та розширення функціональності, наприклад, додавання багатомовності чи мобільної версії додатку. Також необхідно провести додаткове тестування на великих обсягах даних для оцінки масштабованості системи.