

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»
Директор інституту(декан факультету)
_____ Андрій ФОРСЮК _____
(підпис) (ім'я та прізвище)

« 02 » червня 2025р.

«До захисту допущено»
Завідувач кафедри
_____ Сергій ГРИБКОВ _____
(підпис) (ім'я та прізвище)

« 02 » червня 2025р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Комп'ютерні науки

на тему: «Розроблення веб-орієнтованої системи підтримки продажів книгарні» з _____

Виконала: здобувач 4 курсу, групи КН-4-3

_____ Куць Ганна Олександрівна _____
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник _____ Костіков Микола Павлович _____
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____
(ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) _____ (підпис)

_____ (ім'я та прізвище) _____ (підпис)

Рецензент _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ - 2025р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь Бакалавр

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма Комп'ютерні науки

(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних технологій, штучного інтелекту і кібербезпеки

_____ Сергій ГРИБКОВ

«28» квітня 2025 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

_____ Куць Ганни Олександрівни

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення веб-орієнтованої системи підтримки продажів книгарні»,

керівник роботи Костіков Микола Павлович, доцент, к. т. н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «28» квітня 2025 р. № 254-кв

2. Строк подання здобувачем роботи: 30.05.2025

3. Вихідні дані до роботи: загальні відомості про ринок книжкових магазинів.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

1. Дослідження предметної області та постановка задачі

2. Технічне завдання на проектування

3. Проектування, створення та апробація інформаційної системи

5. Перелік графічного матеріалу:

Сітка книг, Реєстрація на сайті, Мій акаунт, Картка книги з рекомендаціями

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Костіков М. П. ,доцент, кандидат технічних наук	28.04.2025	14.05.2025
2	Костіков М. П. ,доцент, кандидат технічних наук	28.04.2025	16.05.2025
3	Костіков М. П. ,доцент, кандидат технічних наук	28.04.2025	28.05.2025

7. Дата видачі завдання: 28.04.2025

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз об'єкту автоматизації., аналіз існуючих аналогів інформаційних систем	02.05.25	Виконано
2	Постановка задачі	04.05.2025	Виконано
3	Проектування системи, проектування бази даних	10.05.2025	Виконано
4	Проектування та реалізація інтерфейсу користувача., опис та реалізація функцій	15.05.2025	Виконано
5	Підготовка пояснювальної записки, презентації	29.05.2025	Виконано

Здобувач

_____ (підпис)

Куць Г. О.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Костіков М. П.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Головною метою цієї кваліфікаційної роботи є створення веборієнтованої інформаційної системи для онлайн-книгарні. Система покликана спростити процес вибору, замовлення та обробки книг як для користувачів, так і для працівників магазину, а також забезпечити ефективне управління інформацією про книги, клієнтів і замовлення.

У межах проєкту реалізовано особистий кабінет, перегляд каталогу, фільтрацію книг за параметрами, кошик, систему замовлень, а також перегляд історії покупок. Розробка здійснена з використанням Django (Python) як бекенд-фреймворку, бази даних MySQL, HTML та CSS для інтерфейсу, а також JavaScript для інтерактивних елементів.

Робота містить 58 сторінки, 4 таблиці, 30 рисунків та 25 джерел.

Ключові слова: КНИГАРНЯ, ІНФОРМАЦІЙНА СИСТЕМА, КНИГА, КОРИСТУВАЧ, PYTHON, HTML

SUMMARY

The main goal of this qualification project is to develop a web-based information system for an online bookstore. The system is designed to simplify the process of selecting, ordering, and processing books for both users and store employees, as well as to ensure efficient management of information about books, customers, and orders.

The project includes features such as a personal user account, catalog browsing, book filtering by parameters, a shopping cart, an order management system, and order history viewing. The development was carried out using Django (Python) as the backend framework, MySQL as the database, HTML and CSS for layout, and JavaScript for interactive elements.

The project consists of 58 pages, 4 tables, 30 figures and 25 sources.

Keywords: BOOKSTORE, INFORMATION SYSTEM, BOOK, USER, PYTHON, HTML

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Аналіз ринку предметної області	9
1.2 Організаційна структура книгарні.....	10
1.3 Аналіз нинішнього стану комп'ютеризації книгарні	12
1.4 Розроблення функціональної моделі та аналіз існуючих бізнес-процесів....	13
1.5 Огляд існуючих рішень для розв'язання виявлених проблем.....	18
1.6 Обґрунтування доцільності проектування й розроблення.....	22
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ.....	24
2.1 Загальні положення.....	24
2.2 Призначення і цілі створення системи.....	24
2.3 Характеристика об'єкта автоматизації	25
2.4 Вимоги до системи.....	25
2.5 Склад і зміст робіт зі створення системи.....	31
2.6. Діаграма Ганта.....	32
2.7. Порядок контролю та приймання системи.....	33
2.8. Вимоги до документації.	33
2.9. Джерела, використані під час розробки.....	33
РОЗДІЛ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	34
3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту.	34
3.2 Проектування та створення бази даних. Опис структури БД.....	36
3.3 Реалізація функцій системи	39
3.4 Інструкція користувача.....	44
3.5 Тестування програмного продукту	51
ВИСНОВКИ.....	55

	7
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТКИ.....	59

ВСТУП

Сьогодні більшість магазинів переходять на нові цифрові технології, щоб працювати швидше, зручніше й ефективніше. Це стосується і книжкових магазинів. Якщо раніше продавці записували замовлення вручну або вели облік у таблицях, то зараз цього недостатньо. Такі методи займають багато часу, можуть містити помилки й створюють незручності для покупців. Саме тому важливо впроваджувати сучасні веб-системи, які допомагають автоматизувати основні процеси — від пошуку книги до оформлення замовлення.

Книгарням потрібна зручна онлайн-система, яка дозволяє покупцям переглядати каталог книг, шукати потрібне, оформляти замовлення й залишати відгуки. Для працівників така система спрощує роботу — вони бачать усі замовлення в одному місці, можуть швидко оновлювати інформацію, контролювати склад і взаємодіяти з клієнтами.

Розробка вебсайту для підтримки продажів у книгарні є дуже актуальною темою, адже все більше людей хочуть купувати книги онлайн, не виходячи з дому. Така система робить роботу магазину сучасною, зручною та відкриває нові можливості для розвитку. Тому ця тема є важливою і відповідає вимогам часу.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз ринку предметної області

Сьогодні книжкові магазини дуже швидко змінюються. Це пов'язано з тим, що сучасні технології розвиваються, і люди почали по-іншому купувати книжки. Раніше більшість ходила в магазин, щоб обрати книгу на місці. Продавці допомагали підібрати літературу, давали поради. Тепер усе більше покупців шукають книжки самостійно — мобільні додатки або інтернет-магазини.

Що цікаво, зараз люди можуть купити книгу, не виходячи з дому. Вони знаходять потрібне видання, читають відгуки, порівнюють ціни і оформлюють доставку. Це зручно, особливо коли книжка не дуже популярна і її складно знайти у звичайному магазині. Завдяки цьому змінився сам підхід до роботи книгарень.

Ще кілька років тому більшість процесів виконували вручну: продавець приймав замовлення, перевіряв наявність книжки, записував дані про покупця. Це займало багато часу і часто спричиняло помилки. Зараз у багатьох магазинах вже є спеціальні програми, які допомагають усе це автоматизувати. Наприклад, швидке оформлення замовлень або ведення бази клієнтів.

Не всі магазини встигають за змінами. Деякі досі ведуть облік у блокноті або Excel, а замовлення обробляють вручну. Через це виникають затримки, помилки, і люди йдуть до більш зручних онлайн-магазинів. Щоб вижити на ринку, книгарням потрібно впроваджувати сучасні рішення. Деякі користуються однією програмою для замовлень, іншою — для фінансів, і ще однією — для обліку товару. Через це бувають затримки або неточності в роботі. Щоб уникнути таких проблем, потрібна одна загальна система, яка з'єднає все разом.

Також варто враховувати, що зараз дуже багато конкурентів. Крім великих мереж, є багато невеликих магазинів, які теж працюють через

інтернет. І щоб не втратити клієнтів, треба постійно вдосконалювати роботу, бути зручним і сучасним.

Можна зробити висновок, що книжковий бізнес сильно змінився. Люди очікують швидкого і якісного обслуговування, а для цього потрібні хороші інформаційні системи. Автоматизація допомагає економити час, зменшує кількість помилок і робить роботу магазину ефективнішою.

1.2 Організаційна структура книгарні

1.2.1 Загальна схема організаційної структури

Щоб магазин працював ефективно, важливо правильно організувати всі процеси. На рисунку 1.1 розглядається, як саме працює книгарня, хто за що відповідає.

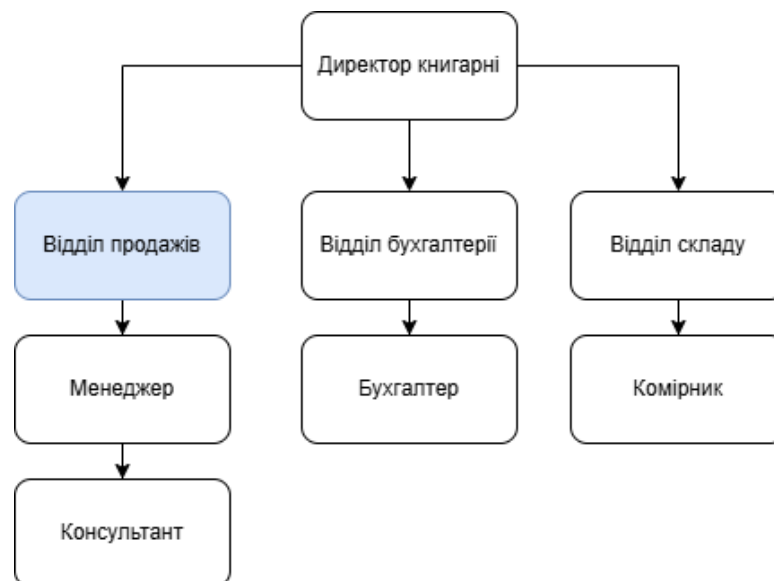


Рисунок 1.1 – Організаційна структура книгарні

Найголовнішим є директор — він керує всією роботою магазину, приймає важливі рішення і контролює, щоб усе працювало правильно. Директору підпорядковуються три основні відділи: продажів, бухгалтерії та склад.

Відділ продажів займається обслуговуванням покупців, допомагає їм у виборі книжок, оформлює замовлення та проводить продажі. На рисунку

виділено відділ продажів, оскільки саме він є об'єктом автоматизації в межах даної роботи. Автоматизація цього відділу дозволить покращити обслуговування клієнтів.

Відділ бухгалтерії відповідає за ведення фінансів магазину. Бухгалтер слідкує за доходами й витратами. Цей відділ забезпечує правильний облік усіх грошових операцій у магазині.

Відділ складу займається зберіганням книжкової продукції. Комірник приймає нові партії книжок, перевіряє їх кількість та стан, розміщує товар на складі.

1.2.2 Структура книгарні

Відділ продажів у книжковому магазині займається всім, що пов'язано з покупцями та продажем книг. Саме через цей підрозділ проходять замовлення, спілкування з клієнтами, оформлення покупок та передача товару на доставку або в торговий зал. У відділі працює менеджер і продавець-консультант. Менеджер слідкує за тим, як проходять продажі і стежить, щоб усе підпорядковано. Консультант працює з покупцями, допомагає вибрати книжку, дає поради в виборі товару, оформлює покупку.

Відділ продажів співпрацює з іншими відділами. Наприклад, зі складом — отримати потрібні книжки, і з бухгалтерією — передати інформацію про продані товари та оплату. Також відділ продажів передає дані директору, коли потрібно зробити звіт або повідомити про щось важливе.

Для автоматизації був вибраний відділ продажів, бо саме тут найбільше рутинної роботи: постійно вводити дані, шукати книги, записувати замовлення [12].

У таблиці 1.1 показано, з ким саме взаємодіє відділ продажів книжкового магазину, яку інформацію він отримує від інших підрозділів і що передає.

Таблиця 1.1. Отримання та передача інформації у відділі продажів

З ким співпрацює	Що отримує	Що передає
Склад	Дані про наявність книг	Замовлення на видачу книжок
Бухгалтерія	Дані про ціни та оплату	Інформація про продажі
Директор	Загальні вказівки	Звіти про роботу, кількість продажів
Клієнти	Інформацію про книжки	Підтвердження замовлення, статус доставки

1.3 Аналіз нинішнього стану комп'ютеризації книгарні

На сьогодні багато книжкових магазинів частково використовують комп'ютерну техніку у своїй роботі, однак рівень автоматизації залишається різним і залежить від масштабу закладу. У невеликих магазинах облік товарів, замовлень та клієнтів ведеться вручну або в простих програмах, таких як Excel. При цьому не всі процеси об'єднані в одну систему, що ускладнює роботу працівників.

Виконання окремих завдань використовуються окремі програми, наприклад: облік залишків товару — в таблицях, обробка онлайн-замовлень — через електронну пошту або месенджери, а звіти — у вигляді окремих файлів. Через це між відділами відсутній налагоджений обмін даними. Усе це призводить до того, що інформацію часто доводиться переносити вручну, що займає багато часу і може викликати помилки. Також у багатьох випадках покупець не має можливості перевірити статус свого замовлення онлайн, адже система не підтримує таку функцію.

У сучасних умовах, коли багато процесів можна виконувати за допомогою комп'ютера, продовжувати все робити вручну — це неефективно і забирає багато часу. Автоматизація допомагає зробити роботу в книжковому

магазині простішою і зручнішою. Усі бачать актуальну інформацію і не плутаються в ній [1].

1.4 Розроблення функціональної моделі та аналіз існуючих бізнес-процесів.

1.4.1 Функціональна модель процесів продажу

Перед створенням нової інформаційної системи, спочатку треба проаналізувати існуючі бізнес-процеси. Створення моделі AS-IS є результатом такого аналізу. Вона відображає, як зараз відбувається процес продажів у книгарні.

Проаналізувавши таку модель, можна визначити головні проблеми, які можна вирішити автоматизацією. Модель AS-IS показує продаж книги у книгарні.

Вхідні дані:

- Запит клієнта
- Контактна інформація
- Дані про книги

Вихідні дані:

- Оформлене замовлення
- Запис про продаж

Механізми моделі:

- Менеджер
- Продавець

Управління моделі:

- Організаційно-нормативна база
- Політика ціноутворення

Модель AS-IS, що візуально представлена в виді контексної діаграми двох рівнів декомпозиції, A-0 — зображена на рисунку 1.2 , A0 — на рисунку 1.3.

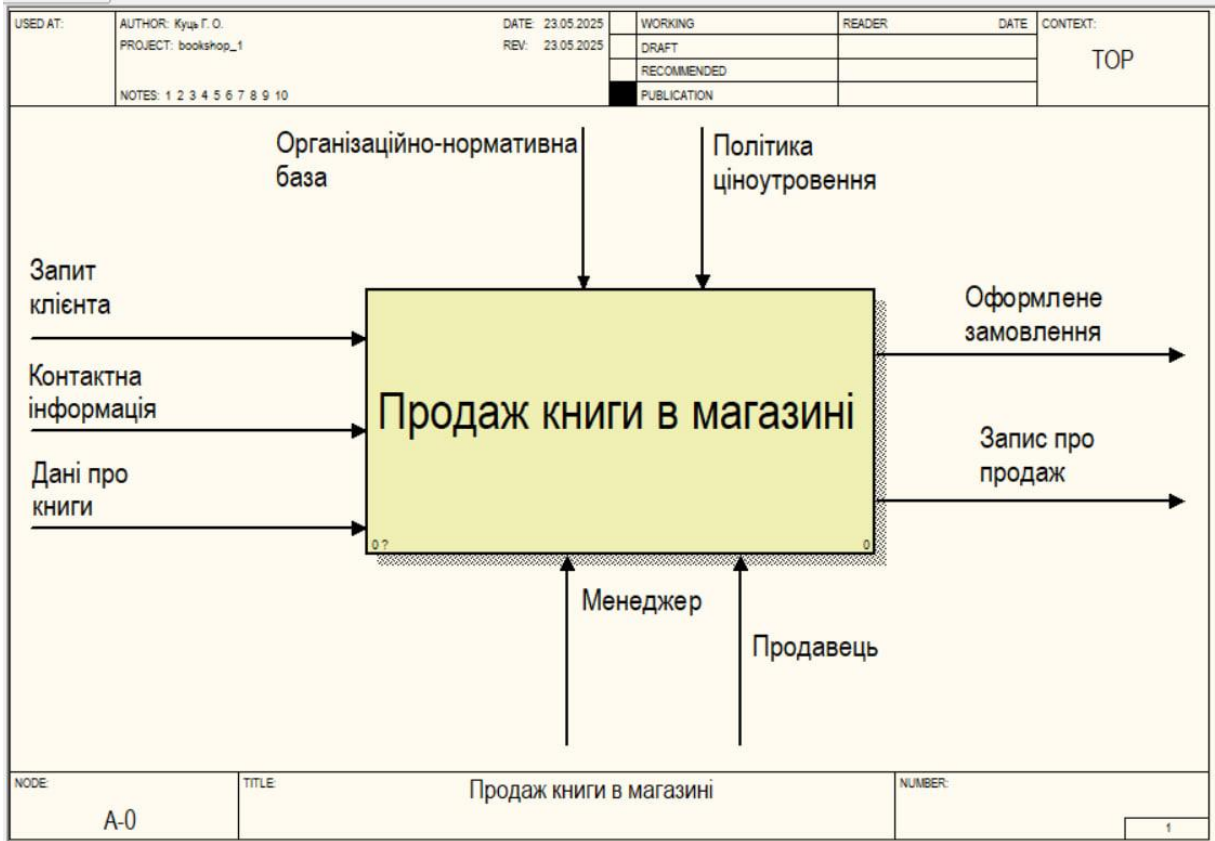


Рисунок 1.2 — Функціональна модель AS-IS

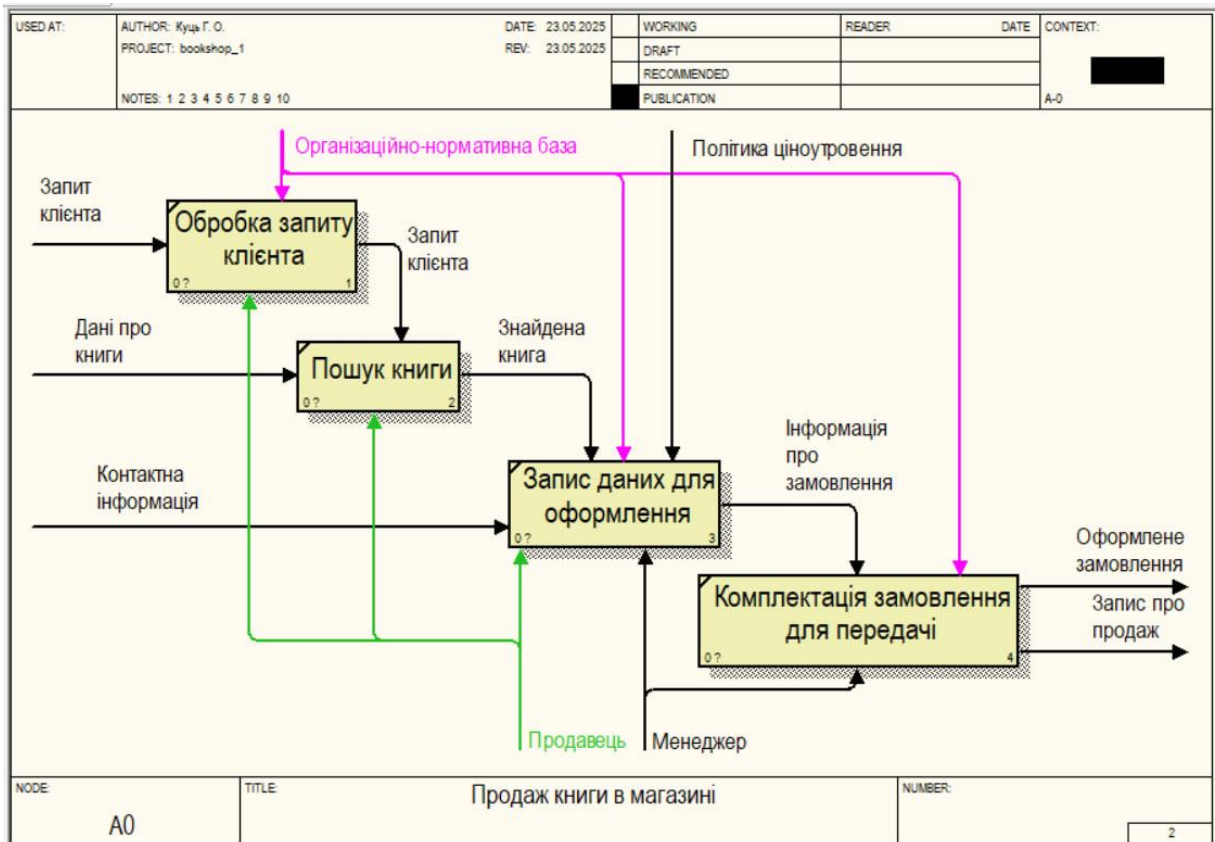


Рисунок 1.3 — Перший рівень декомпозиції AS-IS

1.4.2 Виявлені проблеми

Зараз процес продажу книги в магазині відбувається вручну, без використання єдиної зручної системи. Це створює багато незручностей як для працівників і клієнтів. Щоб дізнатися, чи є потрібна книга, продавці шукають її на полицях, відкривають Excel-файл. Це займає багато часу.

Дані про клієнтів, які залишають замовлення, записують у таблицю. І якщо потім треба знайти старе замовлення або уточнити щось — це проблематично. Немає однієї системи, де зберігаються усі замовлення.

Ще одна складність — ніхто точно не бачить, на якому етапі замовлення: знайдено книгу чи ще шукають, підготували її до видачі чи ні. Через це працівники не мають чіткої картини, і це впливає на швидкість та якість обслуговування. Також не завжди зрозуміло, хто саме з працівників відповідає за те чи інше замовлення. Якщо сталася помилка, складно з'ясувати, де саме щось пішло не так.

Задачі автоматизації:

- забезпечення зручного пошуку книг за назвою, автором, жанром або іншими параметрами в електронному каталозі;
- можливість швидко оформити замовлення без ручного записування даних клієнта;
- автоматичне збереження інформації про замовлення для подальшого перегляду або обробки;

1.4.3 Пропозиції щодо усунення наявних проблем

На сьогоднішній день процес продажу книг у магазині має багато недоліків. Працівники шукають книги вручну, це займає багато часу, особливо коли товару багато. Дані про замовлення клієнтів записують у звичайні таблиці, тому знайти старе замовлення або дізнатись, хто за нього відповідав, важко. Немає чіткого розуміння, на якому етапі зараз кожне замовлення: чи знайдена книга, чи вона вже готова до видачі. Через це виникають плутанина, затримки та помилки.

Щоб вирішити ці проблеми, пропонується застосувати нову модель для роботи — TO-BE, тобто модель, якою система буде після впровадження автоматизації.

Вхідні дані:

- Запит клієнта
- Контактна інформація
- Пошукові параметри книги

Вихідні дані:

- Оформлене замовлення
- Збережені дані клієнта

Механізми моделі:

- Продавець
- Інформаційна система

Управління моделі:

- Політика обслуговування
- Налаштування системи

Функціональна схема бізнес-процесів TO-BE зображена на рисунках 1.4 - 1.5. Після впровадження автоматизованої системи (TO-BE) всі дії виконуватимуться через вебінтерфейс: клієнт сам вводить пошукові параметри, система знаходить потрібну книгу, оформлює замовлення, зберігає контактні дані.

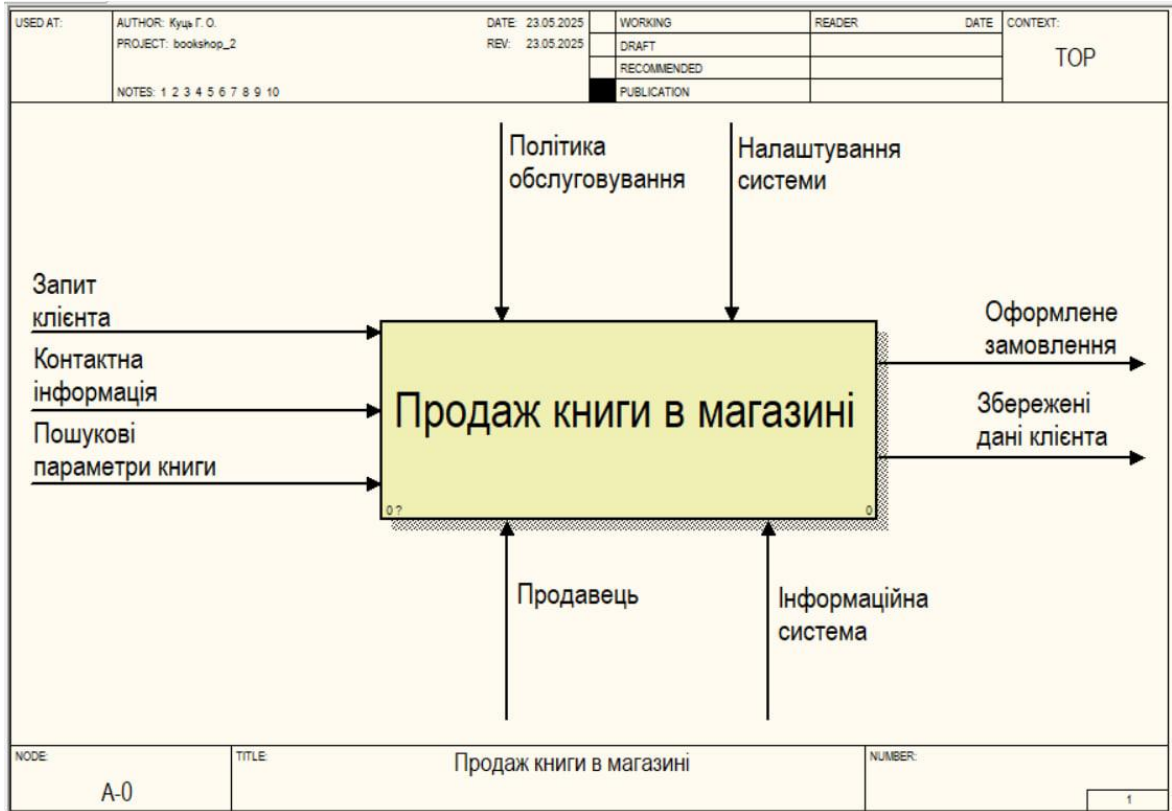


Рисунок 1.4 — Функціональна модель ТО-ВЕ

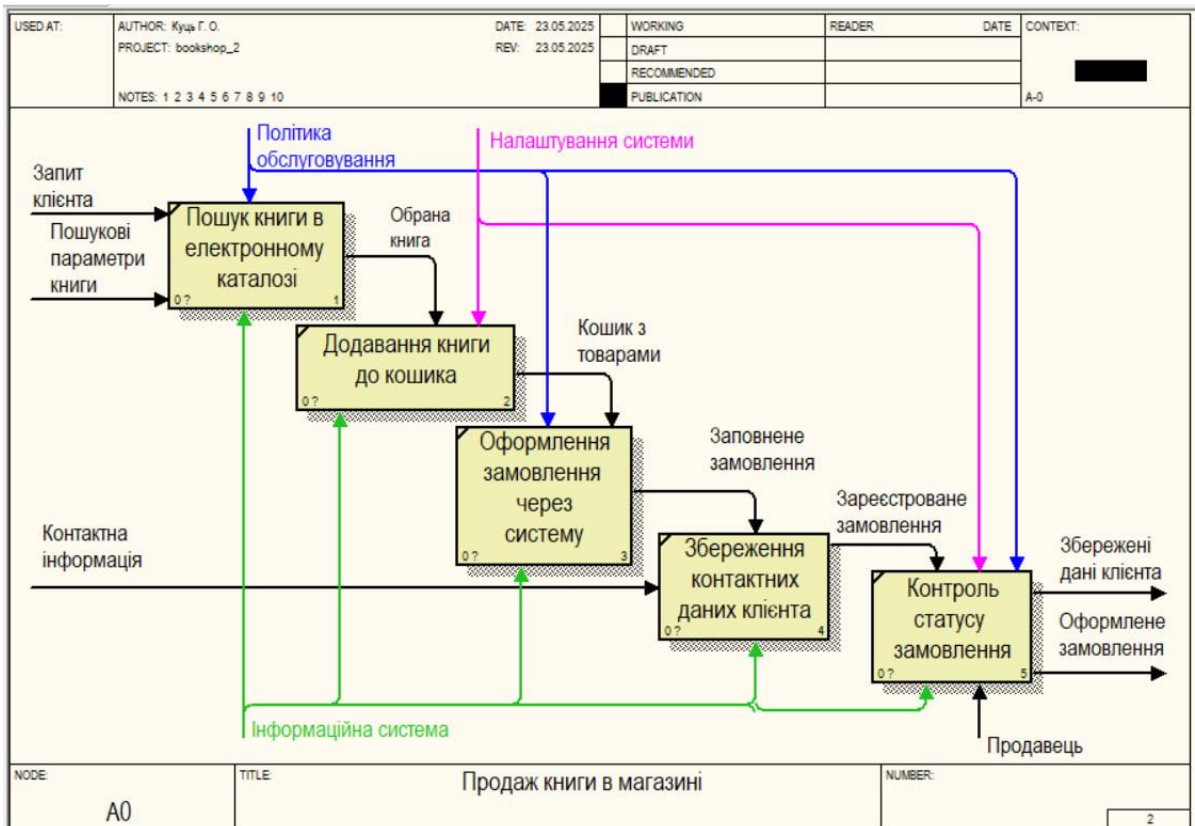


Рисунок 1.5 — Перший рівень декомпозиції ТО-ВЕ

1.5 Огляд існуючих рішень для розв'язання виявлених проблем

1.5.1 Інтернет-магазин «КСД»

Сайт Клубу Сімейного Дозвілля побудований на ідеї книжкового клубу: щоб стати покупцем, треба зареєструватися і час від часу робити замовлення. Інтерфейс (зображений на рисунку 1.6): на головній сторінці видно новинки, акції та добірки. Щоб знайти потрібну книгу, треба ввести її назву чи автора в пошук, вибрати жанр із випадаючого списку. Книги додаються в кошик, при оформленні замовлення потрібно заповнити свої контактні дані та обрати доставку [4].

Особливість КСД – це «Мій кабінет», там зберігаються всі попередні покупки, накопичені бали за програмою лояльності та обов'язкові «клубні» замовлення. Звідти можна редагувати адресу доставки, глянути, які книги були замовлені. У сайту немає мобільного додатка.

КСД дає базову автоматизацію: каталог підтримує фільтри за видавництвом, ціною і тд. , є рівень захисту особистих даних, всі замовлення зберігаються у базі.

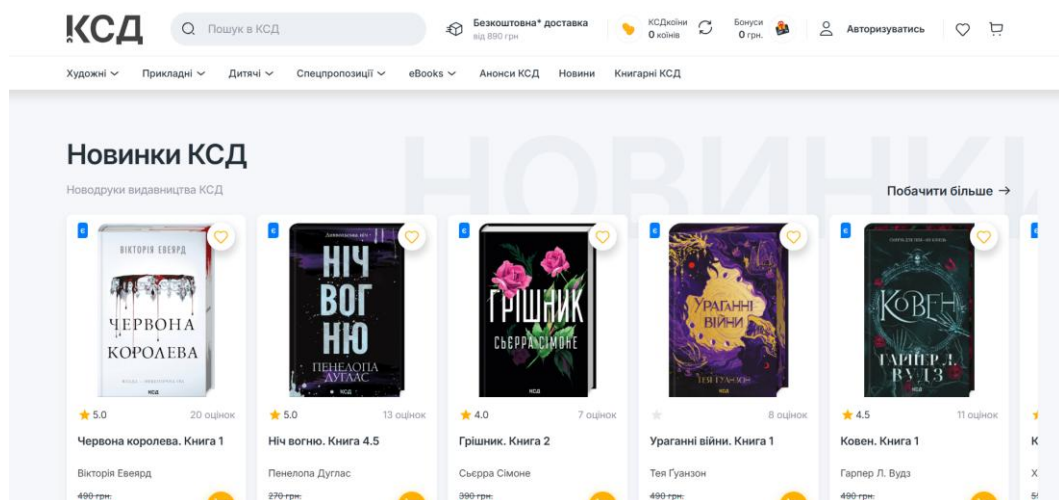


Рисунок 1.6 — Інтерфейс «КСД»

1.5.2 Інтернет-магазин «Якабу»

Yakaboo є універсальною книжковою платформою. На головній сторінці, що зображена на рисунку 1.6 — банер із новими надходженнями, популярними виданнями та акціями. Пошук: можна шукати за назвою чи автором, за ISBN, видавництвом, ключовими словами в описі. Результати можна сортувати за рейтингом, ціною, датою виходу [3].

Якщо натиснути «Додати до кошика» — книга потрапить у список покупок. У формі замовлення автоматично підтягуються дані з облікового запису, якщо зареєстровані. Пропонується швидка реєстрація через соцмережі. Yakaboo підтримує паперові, електронні й аудіокниги: після покупки електронного формату стане доступна PDF-версія, аудіокниги можна буде послухати у вбудованому плеєрі чи в мобільному додатку. Сайт автоматично надсилає повідомлення про доставку, зміну статусу замовлення та нові акції, тож вам не доводиться перевіряти вручну.

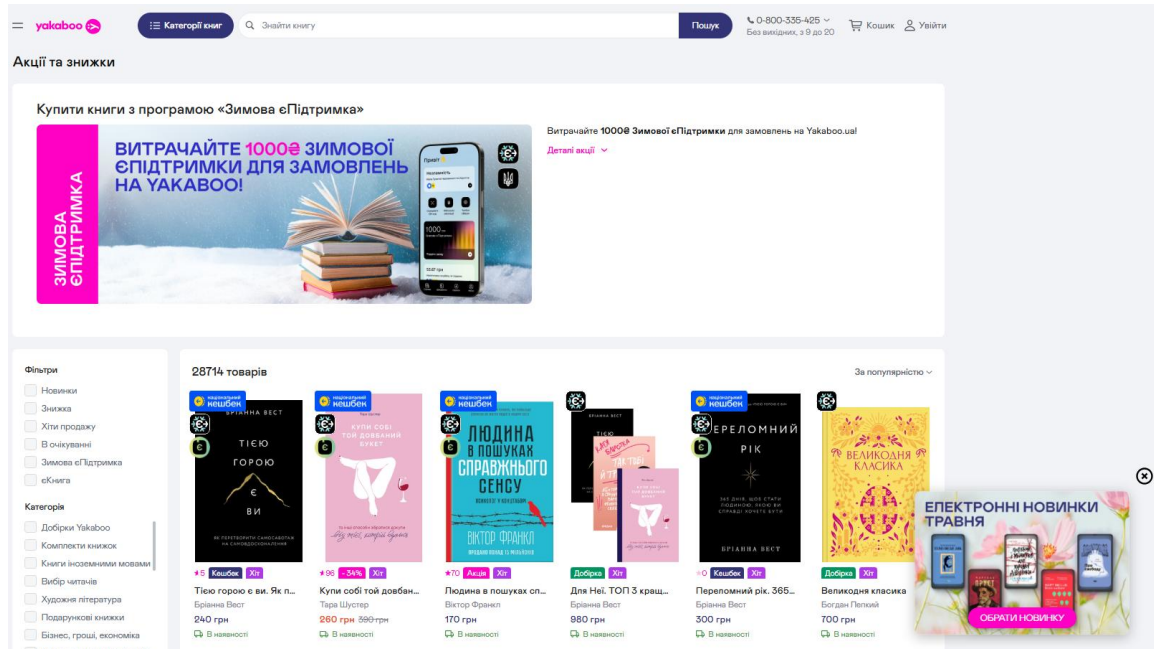


Рисунок 1.7 — Інтерфейс «Якабу»

1.5.3 Інтернет-магазин «Книгарня „Є”»

Сайт «Книгарня Є» — це інтернет-магазин, де можна купити книжки, переважно українською мовою [5]. Він простий у користуванні. Інтерфейс цього сайту мінімалістичний та зображений на рисунку 1.6.

На сайті «Книгарня Є» можна додати книгу до кошика. Для цього треба натиснути кнопку «Купити» біля потрібного товару. Книга потрапляє в кошик, покупець може або продовжити вибір, або перейти до оформлення.

Під час оформлення замовлення відкривається сторінка з усіма обраними книгами. Там показується загальна сума, кількість товарів і форма для введення контактної інформації. Покупець вказує своє ім'я, номер телефону, електронну адресу та вибирає спосіб отримання.

У зареєстрованих користувачів є особистий кабінет, де зберігається історія покупок.

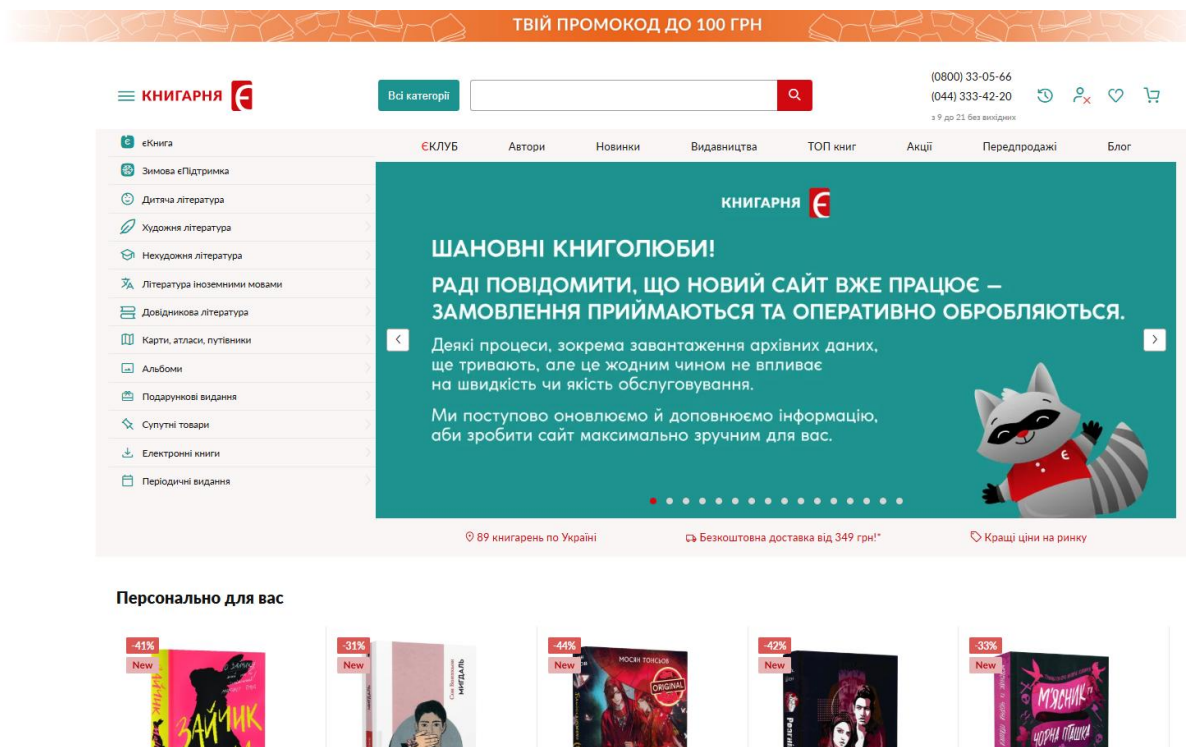


Рисунок 1.8 — Інтерфейс «Книгарня „Є”»

1.5.4 Результат огляду існуючих рішень для розв’язання виявлених проблем

Розглянувши три популярні книжкові сайти — КСД, Yakaboo та Книгарню «Є», можна зробити висновок, що кожна з них має свої особливості, переваги й недоліки.

Виділимо їхні характеристики у порівняльну таблицю 1.2.

Таблиця 1.2. Порівняння систем-аналогів

Критерій	КСД (bookclub.ua)	Yakaboo (yakaboo.ua)	Книгарня Є (book-ye.com.ua)
Швидкодія	Середня	Висока	Висока
Аудіокниги	Немає	Є	Немає
Пошук і фільтри	Прості	Складніші	Середні
Мова інтерфейсу	Українська	Українська	Українська
Мобільний додаток	Немає	Є	Немає
Особистий кабінет	Є	Є	Є
Програма лояльності	Є (клубна система)	Є (бонуси, знижки)	Є (для постійних клієнтів)
Оплата онлайн	Є	Є	Є

Продовження таблиці 1.2

Доставка	Нова пошта, кур'єр	Нова пошта, поштомати, кур'єр	Нова пошта, самовивіз
Інтерфейс	Простий	Сучасний, адаптивний	Мінімалістичний, простий у користанні

З технічної точки зору найфункціональнішою є Yakaboo — він підтримує різні формати книг, має мобільний додаток і зручний інтерфейс. КСД працює добре, підходить для постійних клієнтів, які є членами клубу. Сайт «Книгарня С» зручний, простий, має небагато можливостей. Ці недоліки ускладнюють роботу користувачам.

Порівняння показало, що всі ці сайти мають свої плюси, але жоден з них не підходить ідеально для всіх. Тому створення власної системи для книжкового магазину може бути хорошим рішенням. Це дозволить зробити сайт саме таким, як потрібно, і взяти найкраще з усіх існуючих прикладів. Власна система для магазину дозволить зробити роботу швидшою і простішою.

1.6 Обґрунтування доцільності проєктування й розроблення

У книгарні процеси не до кінця автоматизовані. Працівники самі шукають книги, записують замовлення в таблиці. Це забирає багато часу і призводить до проблем.

Аналіз існуючих книжкових сайтів показав, що кожен із них має свої переваги та обмеження. Деякі мають складне оформлення замовлень або застарілий інтерфейс. Це підтверджує доцільність створення власної системи, яка буде враховувати реальні потреби.

Потрібна система, яка буде автоматизувати основні процеси: пошук книги, оформлення замовлення, збереження даних клієнта, контроль за статусом замовлення. Уся інформація зберігатиметься в одному місці, працівники та клієнти зможуть легко її знайти.

Пропонується створити власну автоматизовану систему для книжкового магазину, яка:

- забезпечить швидкий пошук книг за різними параметрами (назва, автор, жанр);
- дасть можливість оформлювати замовлення онлайн і зберігати контактну інформацію клієнта;
- дозволить клієнту перевіряти статус свого замовлення;
- збере все в 1 базу даних, доступну для персоналу;

Запровадження такої системи зробить роботу швидшою та простішою. Клієнтам буде зручніше замовляти книжки, в той час як магазин зможе працювати ефективніше. Система дозволить врахувати потреби клієнтів і працівників, тому її проектування та розробка є доцільними.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ

2.1 Загальні положення

2.1.1. Найменування системи: «Веб-орієнтована система підтримки продажів книгарні»

2.1.2. Результати робіт зі створення системи оформлюються відповідно до вимог чинних стандартів (зокрема ДСТУ) на кожному етапі розробки. Порядок оформлення і передавання результатів визначається змістом та календарним планом реалізації проєкту.

2.1.3. У разі потреби, на подальших етапах розробки окремі положення можуть бути уточнені або доповнені.

2.2 Призначення і цілі створення системи

2.2.1. Призначення системи

Система призначена для автоматизації процесів управління продажами в книжковому магазині. Вона забезпечує ефективну взаємодію між продавцями і клієнтами.

Розроблена система дозволяє користувачам переглядати каталог книг, здійснювати пошук за різними параметрами, додавати книги до кошика, оформлювати замовлення та переглядати історію покупок у власному акаунті.

Система включає модулі обліку книг, клієнтів, замовлень і кошиків, передбачає обробку замовлень, авторизацію/реєстрацію, управління особистими даними користувача [13].

2.2.2. Цілі створення системи

Головна мета – підвищення ефективності бізнес-процесів шляхом автоматизації обліку, обробки замовлень. Система має забезпечити зменшення ручної праці, покращення обслуговування клієнтів, оперативний доступ до інформації про товари.

Впровадження системи дозволить клієнтам самостійно шукати потрібні товари, оформлювати замовлення, переглядати історію покупок у власному акаунті, а персоналу — ефективно керувати базою книг і клієнтів.

2.3 Характеристика об'єкта автоматизації

2.3.1. Короткі відомості про об'єкт автоматизації

Об'єктом автоматизації є діяльність книжкового магазину, який здійснює продаж друкованої продукції. Основні функції включають: облік книг, проведення продажів. Впровадження веб-орієнтованої системи дозволить централізовано управляти усіма процесами з єдиної платформи.

2.4 Вимоги до системи

2.4.1. Вимоги до системи в цілому

2.4.1.1. Вимоги до структури і функціонування системи

2.4.1.1.1. Система повинна реалізовуватись за клієнт-серверною архітектурою з використанням єдиної бази даних. Повинна бути інтеграція з автоматизованими робочими місцями:

- Продавець;
- Менеджер по роботі з клієнтами.

2.4.1.1.2. Система має діагностувати помилки в роботі, надавати повідомлення про збої та некоректні дії, з метою оперативного усунення проблем.

2.4.1.1.3. Архітектура має передбачати простоту масштабування, оновлення функціональності, підтримку нових технологій та сучасного програмного забезпечення.

2.4.1.1.4. Повинна підтримуватись можливість обробки онлайн-замовлень.

2.4.1.2. Вимоги до персоналу

2.4.1.2.1. Користувачі повинні мати базові знання ПК, пройти інструктаж і дотримуватись правил експлуатації програмного забезпечення.

2.4.1.2.2. Доступ до функцій системи має бути диференційований за рівнями із захистом даних за допомогою паролів.

2.4.1.3. Показники призначення

Система має забезпечувати високу точність, швидкість обробки запитів, своєчасне оновлення даних.

2.4.1.4. Вимоги до надійності

Система повинна працювати стабільно протягом усього робочого дня. Надійність визначається характеристиками СУБД, серверного середовища та мережевого з'єднання.

2.4.1.5. Вимоги до безпеки

Потрібно дотримуватись вимог з електробезпеки, безпеки даних та захисту від несанкціонованого доступу, відповідно до актуальних стандартів і нормативів.

2.4.1.6. Вимоги з ергономіки та технічної естетики

Інтерфейс системи має бути інтуїтивно зрозумілим, з урахуванням принципів юзабіліті та сучасних стандартів веб-дизайну. Загальні ергономічні і естетичні вимоги до системи повинні відповідати держстандартам ДСТУ 8604:2015, ДСТУ 7298:2013. Освітленість робочого місця повинна відповідати ДСТУ EN 12464-1:2016, ДБН В.2.5-28-2006.

2.4.1.7. Вимоги до захисту інформації від несанкціонованого доступу

Для захисту даних у веборієнтованій системі продажів необхідно впровадити такі заходи: кожен сеанс зв'язаний з замовленням користувача повинен починатися з введення унікального логіну та пароля.

2.4.1.8. Вимоги до збереження інформації

2.4.1.8.1. У розробленій системі резервне копіювання бази даних реалізується за допомогою інструменту `django-dbbackup`, який дозволяє автоматично зберігати копії бази на локальний диск або у хмарне сховище. Для поточної конфігурації копії бази даних зберігаються у спеціальній директорії на сервері, з можливістю розширення.

Команда `python manage.py dbbackup` виконує збереження копії, а у разі потреби відновлення виконується через `python manage.py dbrestore`.

2.4.1.10. Вимоги до патентної чистоти

Під час розробки веб-орієнтованої системи продажів патентні дослідження не проводились.

2.4.1.11. Вимоги до стандартизації та уніфікації

Усі інформаційні структури повинні кодуватися згідно з міжнародними стандартами та класифікаторами.

2.4.2. Вимоги до функціональності системи

2.4.2.1. Система повинна підтримувати перелік функцій в таблиці 2.1.

Таблиця 2.1. Функціональні можливості системи

Функція	Опис	Вхідна інформація	Вихідна інформація
Каталог книг	Перегляд усіх книг з назвою, автором, жанром, описом, зображенням тощо.	Запит на перегляд каталогу	Список книг із детальними даними
Пошук і фільтрація	Пошук книг за назвою, автором, жанром; фільтрація результатів.	Пошуковий запит, параметри фільтрів	Відібраний список книг
Система обліку користувачів	Реєстрація, вхід, профіль з історією замовлень.	Дані для реєстрації/входу	Доступ до особистого кабінету
Кошик	Додавання, редагування кількості, збереження для неавторизованих користувачів.	Книга та її кількість	Оновлений вміст кошика
Оформлення замовлення	Форма для створення замовлення з подальшим збереженням у базу.	Книги з кошика, контактні дані	Оформлене замовлення
Історія замовлень	Перегляд усіх попередніх замовлень у профілі користувача.	Запит від користувача	Список минулих замовлень

Продовження таблиці 2.1

Функція повторного замовлення	Кнопка «Повторити замовлення» додає всі книги з попереднього замовлення в кошик.	Ідентифікатор попереднього замовлення	Ті ж книги у поточному кошику
Відображення статусу замовлення	Статуси «Опрацьовується» або «Відправлено» відображаються у профілі.	Ідентифікатор замовлення	Статус замовлення (опрацьовується / відправлено)
Адміністративна панель	Керування книгами, користувачами та замовленнями через Django Admin.	Дані від адміністратора	Оновлена база книг, замовлень або користувачів
Повідомлення	Сповіщення про успішні чи помилкові дії (додавання, оформлення тощо).	Результат дії користувача	Відповідне повідомлення
Рекомендовані книги	Автоматичне виведення схожих книг за автором або жанром.	Інформація про поточну книгу	Список рекомендованих книг
Банери з книгами	Клікабельні банери на головній сторінці з переходом до детальної сторінки книги.	Натискання на банер	Сторінка з описом відповідної книги

Розроблена веб-орієнтована система має охоплювати всі ключові процеси продажу книжкової продукції, забезпечує взаємодію між магазином і клієнтами. Впроваджені функції: перегляд каталогу книг, пошук та фільтрація, робота з кошиком, оформлення і повторення замовлень. Система має підтримувати авторизацію, особистий кабінет з історією покупок і статусами замовлень. Механізм рекомендацій і інтерактивні банери мають створювати персоналізований досвід для покупців [2].

2.4.3. Вимоги до видів забезпечення

2.4.3.1. Вимоги до математичного забезпечення

Щоб реалізувати всі функції системи, не потрібно використовувати складні математичні обчислення або спеціальні алгоритми. Для роботи достатньо стандартних можливостей обраної бази даних, які вже підтримують збереження та обробку інформації.

2.4.3.2. Вимоги до інформаційного забезпечення

Інформаційне забезпечення має містити всі потрібні дані для того, щоб система продажів працювала повноцінно. Структура бази даних повинна бути зручною, щоб дані зберігались без зайвих дублювань і були легко доступні. Заповнювати базу будуть працівники замовника за спеціальними формами, які підготує розробник системи. Як саме буде організовано інформацію, уточнюється під час технічного проектування — згідно з логічною схемою бази.

2.4.3.3. Вимоги до лінгвістичного забезпечення

Програмне забезпечення має створюватися з використанням сучасних мов програмування, які підтримують структурований і об'єктно-орієнтований підхід. Для роботи з базою даних застосовується мова запитів, що відповідає обраній СУБД.

Інтерфейс користувача повинен бути зручним: з меню, підказками та повідомленнями, які допомагають легко виконувати потрібні дії. Більшість запитів до системи мають бути сформульовані простою, зрозумілою мовою.

2.4.3.4. Вимоги до програмного забезпечення

Загальносистемне програмне забезпечення повинно забезпечувати надійну та ефективну роботу всіх функцій системи. До нього належать:

- Операційна система: Windows;
- Система управління базами даних (СУБД): MySQL;
- Фреймворк для реалізації логіки та інтерфейсу: Django (Python).

Вимоги до операційної системи:

- економне використання оперативної пам'яті та дискового простору;
- ОС сервера: Windows Server;

- ОС клієнтських пристроїв: Windows 10 або новіші.

Вимоги до СУБД (MySQL):

- підтримка всіх операцій, необхідних для роботи функцій системи (створення, оновлення, зберігання, пошук даних);
- висока стабільність та надійність;
- швидка обробка запитів користувачів;
- невисокі вимоги до серверного обладнання;
- зручне резервне копіювання та відновлення бази даних.

Вимоги до програмної частини для користувача:

- зручне та зрозуміле візуальне подання інформації (через вебінтерфейс);
- перевірка правильності введених даних та можливість їх редагування;
- діалоговий режим роботи через форми та повідомлення;

2.4.3.5. Вимоги до технічного забезпечення

Для коректної роботи веб-орієнтованої системи продажів книжкового магазину необхідне відповідне технічне забезпечення як на боці сервера, так і на боці користувачів (клієнтів).

Серверна частина:

Сервер, на якому буде розміщено вебсайт і базу даних, повинен мати такі характеристики:

- Сервер: HP ML115 або аналогічний;
- Процесор: Intel Xeon Quad Core, 2.5 ГГц;
- Оперативна пам'ять: 16 ГБ;
- Дискова підсистема: RAID 5, обсяг 1 ТБ;
- Мережевий інтерфейс: 1 Гбіт/с для швидкої взаємодії з клієнтськими пристроями.

Клієнтська частина:

Кожен працівник або користувач, який працює з системою, повинен мати комп'ютер з такими мінімальними характеристиками:

- Процесор: Intel Core i3-6100 або аналогічний;
- Оперативна пам'ять: 4 ГБ;
- Накопичувач: SSD на 240 ГБ для швидкого завантаження системи та браузера;
- Монітор з діагоналлю від 10 до 32 дюймів для зручності перегляду;
- Стандартні пристрої введення: миша та клавіатура з USB або бездротовим підключенням.

2.4.3.6. Вимоги до метрологічного забезпечення

Оскільки система не передбачає вимірювальних пристроїв, вимоги до цього виду забезпечення відсутні.

2.4.3.7. Вимоги до організаційного забезпечення

Організаційне забезпечення системи створюється відповідно до чинних державних норм і стандартів, що стосуються автоматизованих систем управління. Для впровадження системи не потрібно збільшувати кількість працівників — вона розрахована на існуючий штат.

Робочі місця організуються на підприємстві за його власним рішенням. Хто саме матиме доступ до системи, визначає керівник підприємства — відповідний перелік осіб затверджується наказом. У разі збою або виникнення аварійної ситуації відповідальність за вирішення проблем покладається на спеціально призначеного працівника, відповідального за підтримку системи.

2.5 Склад і зміст робіт зі створення системи

Розробка веб-орієнтованої системи для підтримки продажів книжкового магазину здійснюється поетапно. Кожен етап передбачає виконання окремих завдань: від аналізу предметної області до повного впровадження проекту. У таблиці 2.2 наведено основні етапи створення системи з відповідними термінами виконання.

Таблиця 2.2. Етапи роботи

Етап роботи	Термін виконання
Дослідження об'єкта автоматизації	Від 29.04.2025 до 01.05.2025
Розробка технічного завдання	Від 30.04.2025 до 05.05.2025
Створення технічного проєкту	Від 01.05.2025 до 05.05.2025
Оформлення документації	Від 03.05.2025 до 07.05.2025
Розробка структури бази даних	Від 04.05.2025 до 09.05.2025
Створення інтерфейсу користувача	Від 06.05.2025 до 11.05.2025
Реалізація логіки обробки замовлень	Від 10.05.2025 до 16.05.2025
Тестування та відлагодження системи	Від 13.05.2025 до 19.05.2025
Впровадження системи та демонстрація	Від 18.05.2025 до 03.06.2025

2.6. Діаграма Ганта

Сформована Діаграма Ганта наведена на рисунку 2.1.



Рисунок 2.1 — Діаграма Ганта

2.7. Порядок контролю та приймання системи

Всі випробування проводяться згідно з програмою тестування, яка попередньо затверджується.

2.8. Вимоги до документації.

2.8.1. На систему розробляється комплекс документації у складі: технічне завдання та технічний проєкт.

2.9. Джерела, використані під час розробки

- ДСТУ 3008-2015 — вимоги до звітів у сфері науки і техніки;
- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

РОЗДІЛ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту.

Для розробки веб-орієнтованої інформаційної системи було використано багато програмно-технічних засобів. Зокрема:

- Мова програмування Python
- Середовище PyCharm — найрозумніший Python IDE з повним набором інструментів для ефективної розробки Python [14].
- Фреймворк Django
- Система керування базами даних MySQL
- Мови для розробки інтерфейсу та функцій: HTML, CSS та JavaScript
- Технологія AJAX

Розробка виконувалась на комп'ютері з операційною системою Windows 11.

3.1.1 Обґрунтування вибору програмно-технічних засобів

Для реалізації веб-орієнтованої системи продажів книжкового магазину було обрано набір програмно-технічних засобів, які можуть створити повноцінну веб-систему.

1. Python, Django:

Ці засоби були обрані для реалізації всієї серверної логіки:

- Каталог книг, кошик, замовлення, профіль користувача, адмін-панель — реалізовано через `views.py`, `models.py` та `forms.py`, які є складовими Django.
- За допомогою Django можна зручно створювати таблиці в базі даних, обробляти форми замовлення і виконувати перевірку даних, не пишучи все вручну [9].
- Python має читабельний синтаксис і легко підтримується [6, 16].

2. MySQL

Використовується для збереження даних:

- Інформація про книги, користувачів, замовлення та вміст кошика зберігається в базі.

- MySQL дозволяє швидко виконувати запити фільтрації, пошуку та сортування, які використовуються при виведенні каталогу книг [19, 20, 21].

3. HTML, CSS

Ці технології використовувались для створення структури та зовнішнього вигляду веб-сторінок:

- HTML — для розмітки елементів: списків книг, форм входу/реєстрації, кошика, профілю, форми оформлення замовлення.

- CSS — для створення сучасного дизайну сайту: кольори, шапка, кнопки, модальні вікна, адаптивність [7, 22].

4. JavaScript + AJAX

Засоби, що застосовані для забезпечення інтерактивної поведінки веб-сторінки:

- Додавання книг до кошика без перезавантаження сторінки (через fetch або XMLHttpRequest).

- Оновлення кількості товарів, видалення з кошика, підтвердження замовлення — реалізовані за допомогою AJAX-запитів.

- Модальні вікна відкриваються динамічно через JavaScript, що робить сайт зручнішим і швидшим у користуванні [10].

5. PyCharm

Було використано як середовище розробки, яке:

- Підсвічує синтаксис;
- Автоматично перевіряє помилки;
- Дає підказки;
- Інтегрується з Django;
- Дає можливість швидко перемикатись між моделями, шаблонами, URL і тому подібне [8, 15].

Обрані засоби охоплюють рівні роботи системи — від збереження даних у базі до взаємодії з користувачем на вебсторінці. Django зв'язаний з мовою

програмування Python забезпечує надійний серверний функціонал, MySQL — зберігання інформації, а HTML, CSS, JavaScript та AJAX — побудову інтерфейсу з інтерактивною поведінкою. Усі ці інструменти були вибрані з урахуванням вимог проєкту та для більш зручного створення веб-орієнтованої системи.

3.2 Проєктування та створення бази даних. Опис структури БД

Для реалізації бази даних було обрано систему управління базами даних MySQL. Базу даних створено за допомогою ORM Django, що дозволяє автоматично генерувати SQL-структури на основі моделей, написаних у Python.

Програмний код для моделей (models.py):

```
from django.db import models
from django.contrib.auth import get_user_model
from django.conf import settings

User = get_user_model()

class Book(models.Model):
    title = models.CharField(max_length=200)
    author = models.CharField(max_length=200)
    genre = models.CharField(max_length=100, default='Unknown')
    publisher = models.CharField(max_length=100, default='Unknown
Publisher')
    price = models.DecimalField(max_digits=5, decimal_places=2)
    publication_year = models.IntegerField(default=2020)
    language = models.CharField(max_length=50, default='Unknown')
    cover_type = models.CharField(max_length=100,
default='Paperback')
    pages_count = models.IntegerField(default=100)
    description = models.CharField(max_length=500, default='Onuc')
```

```

        image = models.ImageField(upload_to='book_images/',
blank=True, null=True)

```

```

    def __str__(self):
        return self.title

```

```

class Order(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE,
related_name='orders')
    full_name = models.CharField(max_length=100)
    email = models.EmailField() # ← ДОДАНО ЦЕЙ РЯДОК
    address = models.TextField()
    phone = models.CharField(max_length=20)
    created_at = models.DateTimeField(auto_now_add=True)
    sent = models.BooleanField(default=False,
verbose_name='Відправлено')

```

```

    def __str__(self):
        return f"Замовлення #{self.id} від {self.user.username}"

```

```

class OrderItem(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE,
related_name='items')
    book = models.ForeignKey(Book, on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(default=1)

```

```

    def __str__(self):
        return f"{self.book.title} x {self.quantity}"

```

```

class Cart(models.Model):

```

```

user = models.OneToOneField(settings.AUTH_USER_MODEL,
on_delete=models.CASCADE)

```

```

def __str__(self):
    return f"Кошик користувача: {self.user.username}"

```

```

class CartItem(models.Model):
    cart = models.ForeignKey(Cart, on_delete=models.CASCADE,
related_name='items')
    book = models.ForeignKey(Book, on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(default=1)

```

```

def __str__(self):
    return f"{self.book.title} x {self.quantity}"

```

Схема бази даних зображена на рисунку 3.1.

Таблиці бази даних:

- User — зберігає дані користувачів (логін, пароль, email, ім'я).
- Book — містить інформацію про книги (назва, автор, жанр, ціна, опис, рік).
- Order — зберігає замовлення, які зробили користувачі.
- OrderItem — окремі позиції в замовленні (які книги були замовлені і скільки).
- Cart — кошик користувача (тимчасове зберігання книг перед замовленням).
- CartItem — книги, які додано в кошик користувача.

Один користувач може зробити багато замовлень. Одне замовлення може містити багато книг. Один користувач має один кошик. У кошику може бути багато книг [23].

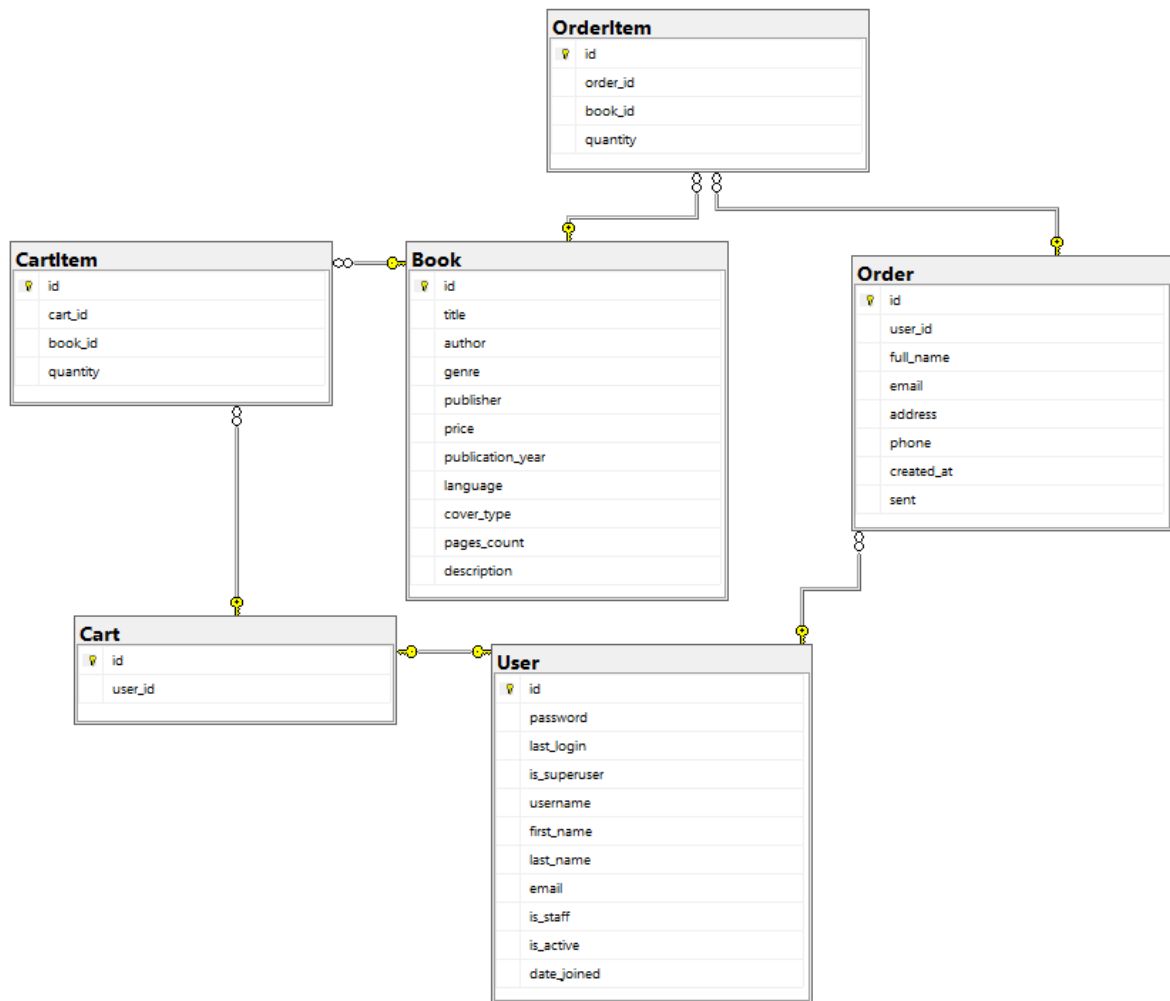


Рисунок 3.1 — Схема бази даних

3.3 Реалізація функцій системи

3.3.1 Каталог книг, пошук і фільтрація

Головна сторінка сайту виконує функцію каталогу, де відображаються всі книги з бази даних. Кожна книга подається у вигляді картки з назвою, автором, жанром, описом, ціною та картинкою. Всі книги автоматично витягуються з бази за допомогою функції у файлі `views.py`, а їх відображення реалізується через HTML-шаблон `book_list.html`.

Користувач може знайти потрібну книгу за допомогою пошукового поля, яке розміщене у верхній частині сторінки. Пошук реалізовано за назвою, автором або жанром. Після введення запиту користувач бачить відібраний список книг. Окремо реалізована розширена фільтрація, яка дозволяє обмежити результати за мовою, жанром, ціною (від – до), роком видання (від

– до), типом обкладинки та видавництвом. Уся логіка обробки фільтрів реалізована у `views.py`, де параметри передаються через URL-параметри, обробляються та фільтрують вибірку книг з бази.

Одним із прикладів використаного елемента графічного інтерфейсу є сітка карток книг, що зображена у додатку А, на рисунку А.1. Дизайн кожної картки розроблений так, щоб відображати всю важливу інформацію на обмеженому просторі. Код для стилізації картки книги наведено в додатку Б. При натисканні на картку спрацьовує JavaScript-функція, яка відкриває модальне вікно [11]. У ньому користувач бачить повний опис книги, тип обкладинки, мову, рік, кількість сторінок та кнопку "Додати в кошик".

На сторінці каталогу також реалізовано посторінковий перегляд (пагінацію), що дозволяє користувачу зручно переглядати великий список книг. Виводиться по 12 книг на одній сторінці. Внизу сторінки з'являються кнопки з номерами сторінок («1», «2», «3» і т.д.), які дозволяють переходити між сторінками. Реалізація пагінації виконується через стандартний механізм Django — клас `Paginator`. У шаблоні сторінки виводиться відповідний блок із номерами сторінок і кнопками «Назад» / «Вперед» залежно від поточної сторінки.

3.3.2 Реєстрація, профіль користувача, кошик

Відвідувач веб-сторінки може зареєструватися, увійти у свій обліковий запис та отримати доступ до особистого кабінету. Реєстрація (див. Додаток А, рисунок А.2) здійснюється через окрему сторінку, де користувач заповнює просту форму: вводить своє ім'я, email та пароль. Реалізація виконана на основі кастомної форми Django (`RegisterForm`) і збереження даних у базу через модель користувача [17]. Після натискання кнопки «Зареєструватися» система автоматично створює профіль користувача в базі даних і перенаправляє його на сторінку входу. Після успішного входу користувач потрапляє до особистого кабінету.

Увійшовши в акаунт, інтерфейс якого зображено у додатку А, на рисунку А.3, користувач бачить сторінку «Мій акаунт», де відображаються всі

попередні замовлення. Дані витягуються з бази через зв'язки між моделями User, Order та OrderItem, а логіка побудована на фільтрації замовлень за користувачем. Для кожного замовлення вказано дату створення, перелік книг, загальну суму та статус: «Опрацьовується» або «Відправлено». Біля кожного замовлення знаходиться кнопка «Повторити замовлення яка викликає функцію у views.py, що копіює склад попереднього замовлення у поточний кошик.

Кнопка «Кошик» доступна у верхній частині кожної сторінки сайту, тож користувач у будь-який момент може переглянути додані товари. Якщо користувач не авторизований, кошик працює на основі сесії браузера — це реалізовано через збереження списку товарів у request.session. Якщо ж користувач увійшов у систему, книги зберігаються в базі даних і пов'язуються з обліковим записом через моделі Cart та CartItem.

Книги додаються до кошика з модального вікна — після натискання кнопки «Додати в кошик» у картці книги. Додавання обробляється через JavaScript, який надсилає асинхронний запит (AJAX) до представлення add_to_cart. У кошику користувач бачить список вибраних книг, може змінити кількість або видалити непотрібні. Також відображається загальна сума замовлення та кнопка «Оформити замовлення», яка веде до сторінки з формою заповнення контактних даних і підтвердженням покупки.

Усі дії супроводжуються візуальними повідомленнями — наприклад, повідомлення про успішну реєстрацію, помилку при вході, додавання книги в кошик або підтвердження замовлення. Вони реалізовані за допомогою системи messages у Django та виводяться у відповідних шаблонах. Коди реалізації функцій наведені у додатку Б [18].

3.3.3 Рекомендовані книги, рекламні банери з книгами

На головній сторінці реалізовано функцію банерів із книгами та автоматичну підбірку рекомендованих видань. Це зроблено для того, щоб одразу привернути увагу користувача до популярних чи тематичних книг і спростити вибір. Реалізація банерів відбувається через спеціальний блок

carousel, який оновлюється за допомогою JavaScript через кожні кілька секунд. Для цього використовуються HTML-елементи з класом banner-card, в які динамічно підставляється інформація про книги.

Банери з книгами розміщено у верхній частині головної сторінки. Кожен банер — це велика ілюстрація з обкладинкою книги, яка клікабельна. При натисканні виконується JavaScript-функція, що відкриває модальне вікно з детальним описом книги, її ціною, мовою, типом обкладинки та кнопкою «Додати в кошик». Усі дані підтягуються автоматично з data-* атрибутів елементів. Слайдер банерів працює плавно, з авто-перелистуванням, що створює ефект галереї.

Під банерами розміщені спеціальні підбірки — це блоки з рекомендованими книгами. Наприклад, підбірка «Класика» або «Книги українською». Кожна підбірка виводиться у вигляді горизонтального слайдера з кількома книжками. Користувач може гортати ці каруселі ліворуч і праворуч за допомогою кнопок або свайпів. Усі книги подані у вигляді карток (book-card), які працюють так само, як у каталозі: натиснувши на картку, можна переглянути детальну інформацію та додати книгу до кошика.

Рекомендовані книги формуються автоматично на основі жанру або автора поточної книги, яку переглядає користувач. У backend-частині (views.py) система визначає жанр і автора відкритої книги та формує запит до бази, щоб вибрати кілька схожих. Ці книги передаються в шаблон у вигляді окремого списку та виводяться в нижній частині модального вікна або на сторінці пошуку. Це забезпечує динамічне оновлення рекомендацій залежно від вибору користувача.

Коди реалізації функцій наведені у додатку Б.

3.3.4 Оформлення замовлення, адмінпанель

Коли користувач натискає кнопку «Оформити замовлення» в кошику, система перевіряє, чи він авторизований. Якщо ні — перенаправляє на сторінку входу. Якщо так — відкривається форма для введення контактних даних.

Форма реалізована через OrderForm і зв'язана з моделлю Order. Після заповнення та підтвердження, у функції `make_order` (`views.py`) створюється новий об'єкт замовлення в базі, а всі книги з кошика (`CartItem`) автоматично додаються до таблиці `OrderItem`. Для кожного запису вказується книга, кількість та зв'язок із замовленням. Одразу після цього кошик очищується.

Між моделями реалізовано зв'язок «один-до-багатьох»: одне замовлення — багато книжок. Усі перевірки полів форми, як-от порожні значення чи некоректні email-адреси, обробляються стандартною валідацією Django.

Після успішного створення замовлення користувач бачить підтвердження, а замовлення автоматично зберігається зі статусом «Опрацьовується» і відображається у профілі.

Адміністративна панель реалізована на основі вбудованого інструменту Django Admin. Вона доступна лише для авторизованих адміністраторів і дозволяє повноцінно керувати вмістом сайту без додаткової розробки інтерфейсу.

У `admin.py` проєкту зареєстровані моделі `Book`, `Order`, `OrderItem`, `User`, `Cart`, `CartItem`. Це автоматично створює в адмінпанелі окремі розділи для кожної сутності.

Адміністратор може:

- додавати або редагувати книги (із заповненням усіх полів — назва, автор, жанр, опис, обкладинка тощо);
- переглядати замовлення користувачів із деталями;
- змінювати статус замовлень (наприклад, з «Опрацьовується» на «Відправлено»);

Усі ці можливості реалізовані за допомогою підключення моделей до `admin.site.register(...)`. Django автоматично формує зручні форми й таблиці для управління даними, з підтримкою фільтрів, пошуку та сортування. Коди реалізації функцій наведені у додатку Б.

3.4 Інструкція користувача.

Коли користувачі заходять на сайт, перед ними відкривається головна сторінка, що зображена на рисунку 3.2.

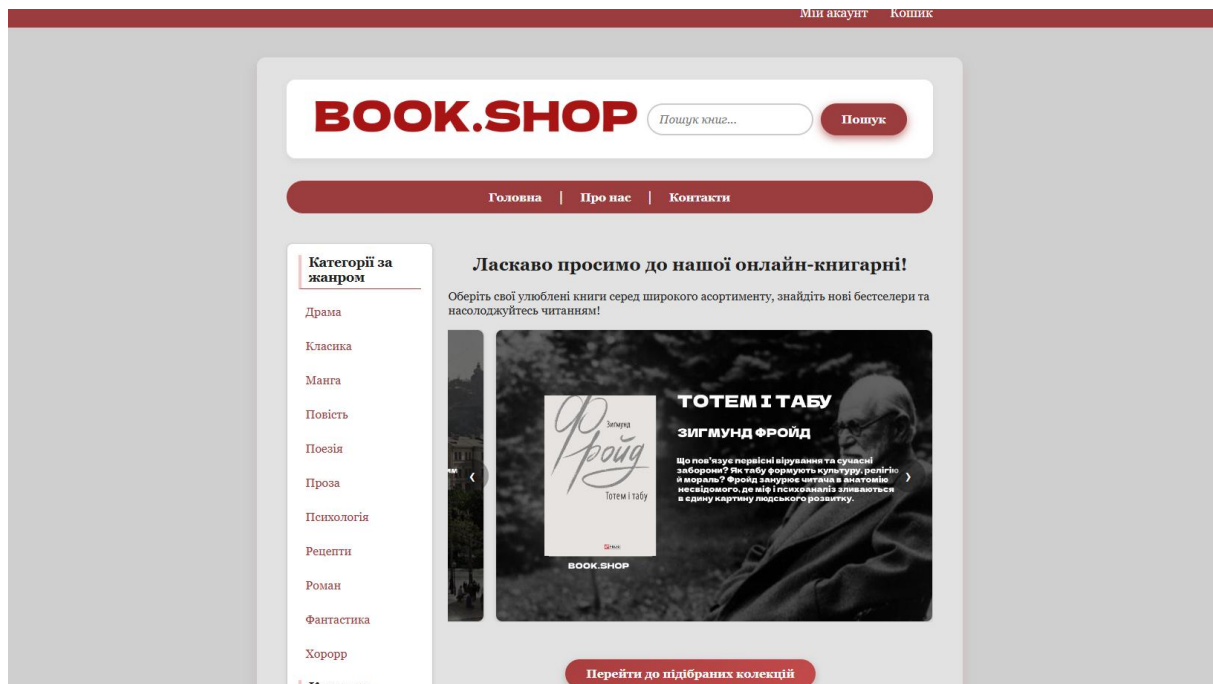


Рисунок 3.2 — Головна сторінка

Якщо натиснути на категорію відкриється сторінка з відповідними книгами та фільтрами для пошуку по цій категорії .

Якщо натиснути на банер, з'явиться картка книги відповідної. Кожна книга подається у вигляді окремої картки — із зображенням, назвою, автором, жанром, описом, ціною та роком видання. Приклад такої картки зображено на рисунку 3.3.



Рисунок 3.3 — Картка книги

В шапці сайту є пошукове поле, що зображене на рисунку 3.4.



Рисунок 3.4 — Пошук

Туди можна вводити запити по назві, автору, видавництву. Можна застосовувати фільтри та скидати їх. Це можна зробити за допомогою відповідних кнопок.

Коли книга додається до кошика, користувачі можуть переглянути її у відповідному розділі, що зображений на рисунку 3.5. Кнопка «Кошик» завжди доступна у верхньому меню.

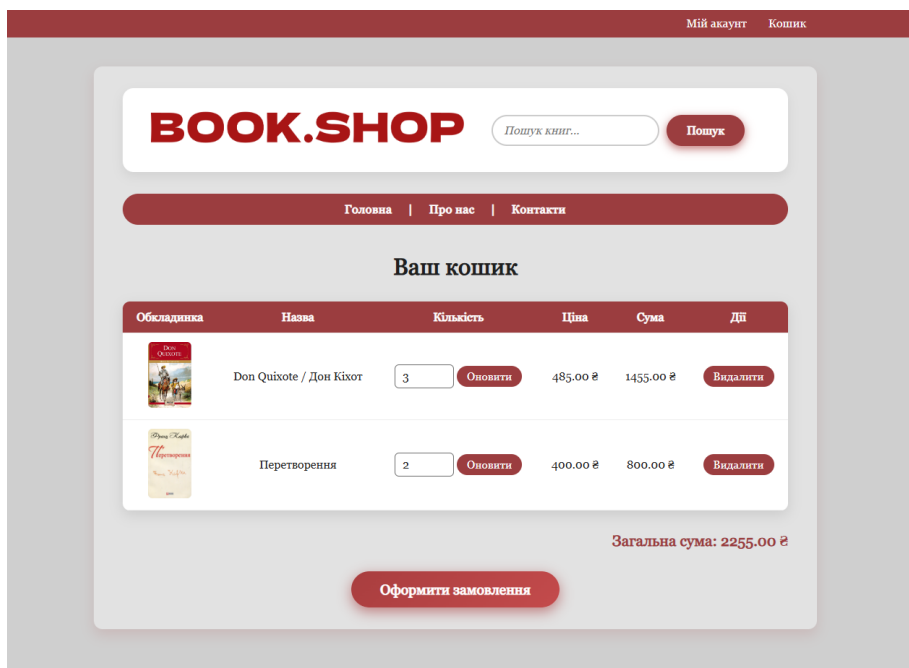


Рисунок 3.5 — Кошик

У кошику видно всі вибрані книги, їх кількість і ціни. Можна змінювати кількість книг або видаляти зайві. Під списком є загальна сума й кнопка «Оформити замовлення». Якщо її натиснути не авторизованому користувачу то з'явиться вікно, яке зображене на рисунку 3.6.

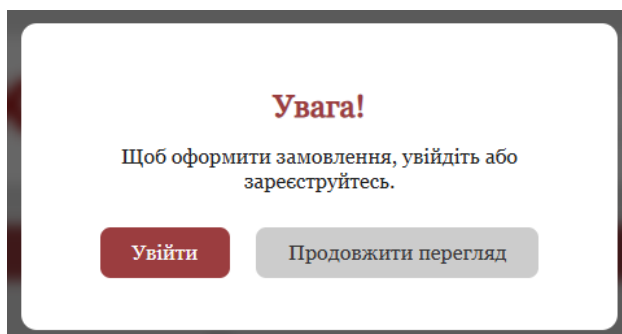



Рисунок 3.6 — Повідомлення користувачу про те що, тільки авторизовані користувачі можуть оформляти замовлення

Якщо натиснути «Продовжити перегляд», ми залишимося у кошику.
Якщо натиснути «Увійти», перекине на форму входу, що зображена на
рисунок 3.7.



Вхід

Логін:

Книголюб Петро Вакарюк

Пароль:

Пароль

Увійти

Ще не маєте акаунта? **Зареєструватися**

Рисунок 3.7 — Форма для входу

Можна натиснути «Зареєструватись», після чого відкриється форма для
реєстрації. Форма реєстрації зображена на рисунку 3.8.

Реєстрація

Логін:

Ім'я:

Прізвище:

Email:

Телефон:

Пароль:

- ► Мінімум 8 символів
- ► Хоча б одна цифра
- ► Хоча б один спец. символ (!@#\$...)

Підтвердження пароля:

Зареєструватися

Вже маєте акаунт? [Увійти](#)

Рисунок 3.8 — Форма реєстрації

Після реєстрації ми переходимо на сторінку входу. Тут ми вводимо логін та пароль, які щойно вказували під час реєстрації. Після натискання кнопки «Увійти» нас перенаправляє до особистого кабінету. Його вигляд зображений на рисунку 3.9.

Якщо натиснути кнопку «вихід», система перенеправить на форму для входу. Можна також відразу потрапити до кошика натиснувши на кнопку «Перейти до кошика».

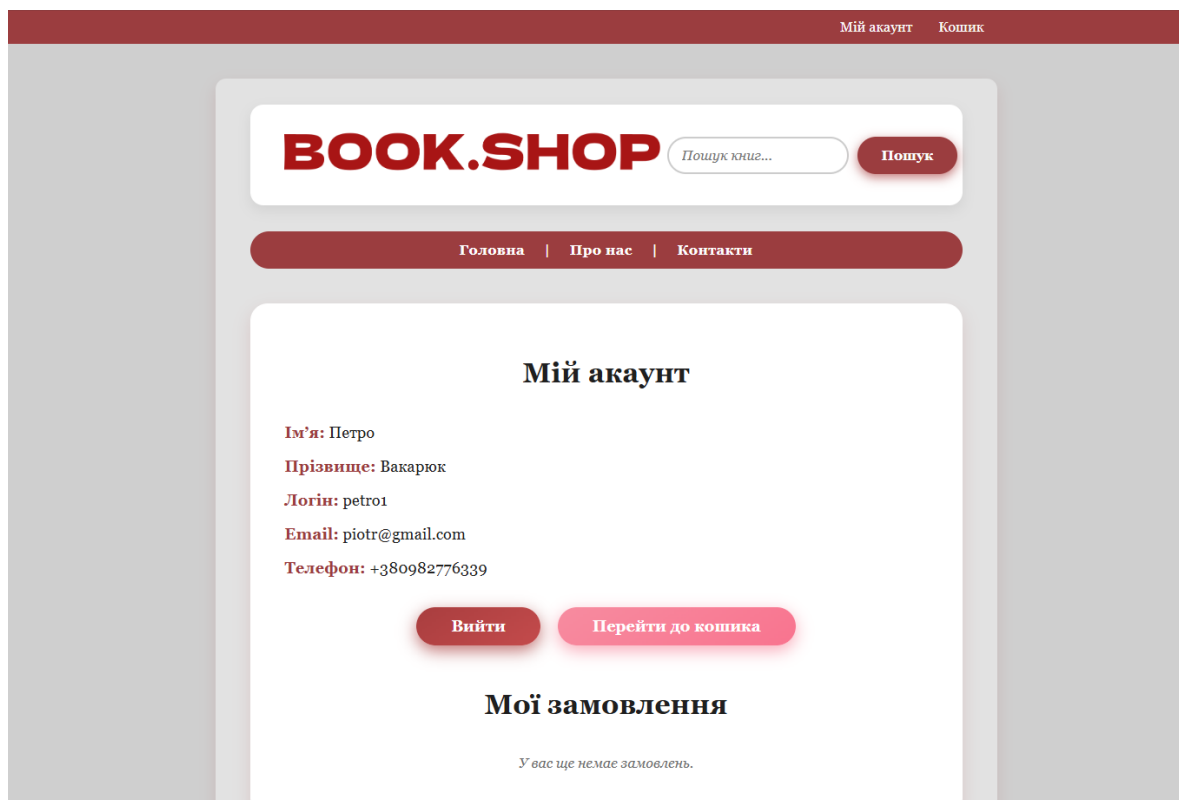


Рисунок 3.9 — Особистий кабінет

Якщо користувач авторизований та хоче зробити замовлення, слід натиснути на кнопку «Оформити замовлення». Це при умові якщо хоча б 1 товар є в кошику.

У перше поле вводять ім'я та прізвище отримувача, у друге — електронну пошту. Далі потрібно вказати адресу доставки: місто, вулицю, номер будинку або іншу інформацію, яка допоможе доставити замовлення. У останньому полі вписують номер телефону — на нього можуть зателефонувати, якщо потрібно буде уточнити.

Коли всі поля заповнені, користувач натискає кнопку «Підтвердити замовлення». Якщо ж хочемо змінити замовлення — натискаємо кнопку «Повернутись до кошика» і повертається назад. Після підтвердження замовлення система зберігає дані, очищує кошик і додає це замовлення до профілю користувача. Замовлення одразу отримує статус «Опрацьовується», і його можна буде побачити у своєму акаунті (Рис. 3.10)



Рисунок 3.10 — Мої замовлення

Якщо натиснути на кнопку «Повторити замовлення», з'явиться вікно підтвердження дії, воно зображене на рисунку 3.11.

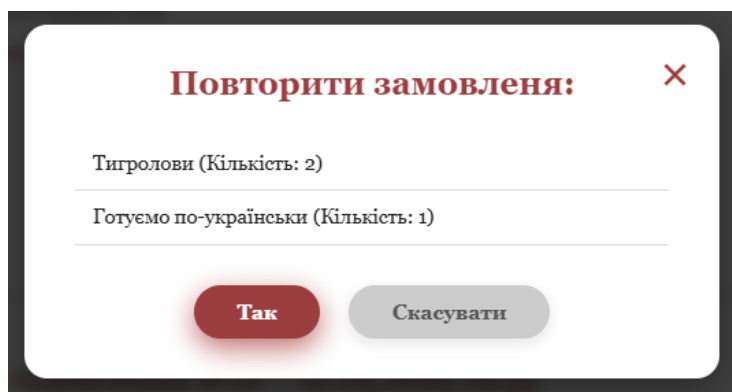


Рисунок 3.11 — Повторити замовлення

Якщо користувач вибирає «Так», перелічені книги у цьому вікні додаються у кошик, у тій же самій кількості.

Статус замовлення «Замовлення опрацьовується» означає що, замовлення прийняте та розглядається менеджером. Він може зателефонувати щоб уточнити подробиці або готує товар на відправлення. Коли замовлення буде відправлено, статус зміниться у профілі на «Відправлено» (Рис. 3.12).

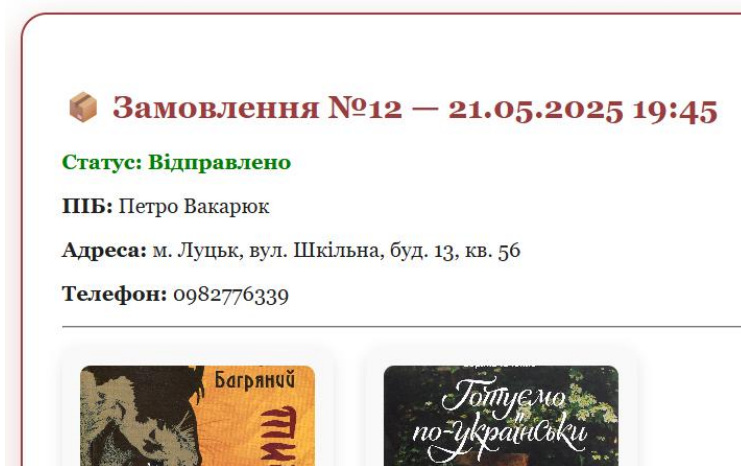


Рисунок 3.12 — Змінений статус замовлення

Для цього в адмінпанелі існує поле, яке можна позначити «пташкою» (Рис. 3.13).

ID	USER	FULL NAME	EMAIL	PHONE	ADDRESS	СУМА ЗАМОВЛЕННЯ	CREATED AT	ВІДПРАВЛЕНО
12	petro1	Петро Вакарюк	piotr@gmail.com	0982776339	м. Луцьк, вул. Шкільна, буд. 13, кв. 56	920.00 ₪	May 21, 2025, 7:45 p.m.	<input type="checkbox"/>

Рисунок 3.13 — Зміна статусу замовлення у адмін панелі

3.5 Тестування програмного продукту

Тестування є невід’ємною складовою розробки інформаційної системи, воно дозволяє впевнитися в її стабільній роботі. Тестування проводитиметься в браузерях — зокрема, Google Chrome, Microsoft Edge та Opera (Браузер 1, Браузер 2, Браузер 3) [24, 25].

Під час тестування була проведена перевірка наважливих і основних функцій, що розроблені в інформаційній системі відповідно до технічного завдання.

Тестування «Пошук книг та фільтрація» зображено на рисунку 3.14.

№	Пріоритет	Перевірка	Очікуваний результат	Результат перевірки			
1	Назва	Пошук книг					
2	Опис	Перевірка пошуку книг та фільтрації					
3	Вимоги						
4	Статус	Виконано					
5							
6	№	Пріоритет	Перевірка	Очікуваний результат	Результат перевірки		
7	Відображення відгука в списку				Браузер 1	Браузер 2	Браузер 3
8	<i>Перевірки для відображення відгуків з різними властивостями</i>						
9	1	високий	Відкрити сторінку кнопки "Пошук" працюють	Відображення сторінки "Пошук"	Passed	Passed	Passed
10	2	високий	Фільтри працюють	Правильне відображення пошуку та вводу даних для пошуку	Passed	Passed	Passed
11	3	високий	Фільтри працюють	Коректне та зрозуміле відображення книг за фільтрами	Passed	Passed	Passed

Рисунок 3.14 — Результат тестування «Пошук та фільтрація»

Тестування «Мій акаунт» зображено на рисунку 3.15.

№	Пріоритет	Перевірка	Очікуваний результат	Результат перевірки			
	Назва	Мій акаунт					
	Опис	Перевірка роботи елементів					
	Вимоги						
	Статус	Виконано					
	№	Пріоритет	Перевірка	Очікуваний результат	Результат перевірки		
	Відображення відгука в списку				Браузер 1	Браузер 2	Браузер 3
	<i>Перевірки для відображення відгуків з різними властивостями</i>						
	1	високий	Відкриття сторінки "Мій акаунт"	Відображення сторінки "Мій акаунт"	Passed	Passed	Passed
	2	високий	Кнопки "Вийти", "Повернутись до кошика", "Повторити замовлення" працюють	Правильне відображення та робота кнопок "Вийти", "Повернутись до кошика", "Повторити замовлення"	Passed	Passed	Passed
	3	високий	Статуси замовлень	Статуси замовлень працюють коректно	Passed	Passed	Passed

Рисунок 3.15 — Результат тестування «Мій акаунт»

Тестування «Мій акаунт» зображено на рисунку 3.16.

№	Пріоритет	Перевірка	Очікуваний результат	Результат перевірки			
	Назва	Кошик					
	Опис	Перевірка роботи елементів					
	Вимоги						
	Статус	Виконано					
	№	Пріоритет	Перевірка	Очікуваний результат	Результат перевірки		
	Відображення відгука в списку				Браузер 1	Браузер 2	Браузер 3
	<i>Перевірки для відображення відгуків з різними властивостями</i>						
	1	високий	Відкриття сторінки "Кошик". Статуси замовлень працюють коректно	Відображення сторінки "Кошик". Статуси замовлень працюють коректно	Passed	Passed	Passed
	2	високий	Кнопка "Оформити замовлення" та відповідне модальне вікно працює коректно	Правильне відображення та робота елементів	Passed	Passed	Passed
	3	високий	Правильний обрахунок суми замовлення	Статуси замовлень працюють коректно	Passed	Passed	Passed

Рисунок 3.16 — Результат тестування «Кошик»

Тестування «Увійти/зареєструватись» зображено на рисунку 3.17.

Назва		Увійти/реєстрація				
Опис		Перевірка роботи елементів				
Вимоги						
Статус		Виконано				
№	Пріоритет	Перевірка	Очікуваний результат	Результат перевірки		
<i>Відображення відеореєстрації в списку</i>				Браузер 1	Браузер 2	Браузер 3
<i>Перевірки для відображення відеореєстрації з різними властивостями</i>						
		Відкриття сторінки "Увійти"	Відкриття сторінки "Увійти"	Passed	Passed	Passed
1	високий	Відкриття сторінки "Увійти"	Відкриття сторінки "Увійти"	Passed	Passed	Passed
2	високий	Кнопка "Увійти" працює коректно"	Кнопка "Увійти" працює коректно	Passed	Passed	Passed
3	високий	Відкриття сторінки "Реєстрація"	Відкриття сторінки "Реєстрація"	Passed	Passed	Passed
4	високий	Введення даних та кнопка "Зареєструватись" працюють коректно	Введення даних та кнопка "Зареєструватись" працюють коректно	Passed	Passed	Passed

Рисунок 3.17 — Результат тестування «Увійти/реєстрація»

У межах проєкту було проведено тестування основних функціональних модулів вебсайту онлайн-книгарні. Перевірці підлягали такі ключові розділи системи, як вхід і реєстрація користувачів, робота з кошиком, особистий кабінет, система пошуку та фільтрації книг.

Функціонал пошуку книг та застосування фільтрів показав себе надійно. Всі сторінки завантажувалися без помилок, поля форми працювали згідно з очікуваннями, а результати відображались коректно після вибору фільтрів. Візуально сторінка залишалась зручною та зрозумілою для користувача.

Сторінка «Мій акаунт» продемонструвала стабільну роботу. Всі важливі кнопки — «Вийти», «Повернутись до кошика», «Повторити замовлення» — функціонували згідно з поставленими завданнями. Інформація про замовлення виводилася в актуальному вигляді, зміна статусів працювала без збоїв.

Перевірялася робота кошика, зокрема функції обрахунку загальної суми, відображення замовлень і взаємодії з модальними вікнами при оформленні покупки. Усі ці елементи функціонували згідно з очікуваним результатом, і жодних помилок або збоїв виявлено не було.

Вхід через форму «Увійти» працює стабільно, реєстрація нового користувача виконується без затримок і завершується створенням облікового запису з подальшим автоматичним перенаправленням до особистого кабінету.

Поля для введення даних перевіряються на валідність, а помилки коректно підсвічуються.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи ставилася головна мета — створити веб-орієнтовану систему для підтримки продажів книжкового магазину, яка б дозволила автоматизувати ключові бізнес-процеси та зробити обслуговування клієнтів зручнішим.

Для реалізації ідеї спочатку було здійснено аналіз роботи книгарні без використання автоматизованих засобів. У результаті було виявлено основні процеси, які потребували покращення. На основі цього сформульовано перелік функцій-вимог до веб-системи: визначено необхідну функціональність, особливості структури інтерфейсу та загальні принципи побудови. Наступним кроком було проектування бази даних, що дає зберігання інформації про книги, клієнтів і замовлення. Після цього реалізовано ключові сторінки сайту: головну сторінку, каталог книжок, модулі реєстрації та авторизації, кошик, оформлення замовлень.

Сайт вирішує важливі питання: автоматизує продаж книг, спрощує обробку замовлень, надає зручний інтерфейс для клієнтів і дає змогу зручніше більше працювати з базою товарів.

Очікується, що впровадження цієї системи позитивно вплине на роботу магазину, зробить її сучаснішою, ефективнішою та доступнішою для покупців.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки» денної та заочної форми здобуття освіти [Електрон. ресурс] / уклад. С. В. Грибков, Н. В. Ліманська, М. П. Костіков. – К.: НУХТ, 2025. – 43 с.
2. Література з аналізу інформаційних систем збірник наукових праць / за ред. І. І. Даниленка. – Київ : КПІ ім. Ігоря Сікорського, 2019.– 224 с.
3. Yakaboo. Офіційний сайт [Електронний ресурс] – Режим доступу: <https://www.yakaboo.ua> (дата звернення: 19.05.2025)
4. Книгарня "Є". Офіційний сайт [Електронний ресурс] – Режим доступу: <https://book-ye.com.ua> (дата звернення: 19.05.2025)
5. Клуб Сімейного Дозвілля. Офіційний сайт [Електронний ресурс] – Режим доступу: <https://ksd.ua> (дата звернення: 19.05.2025)
6. Python (мова програмування). Вікіпедія – вільна енциклопедія [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Python> (дата звернення: 19.05.2025)
7. W3Schools. HTML, CSS, JavaScript tutorials [Електронний ресурс] – Режим доступу: <https://www.w3schools.com> (дата звернення: 19.05.2025)
8. JetBrains PyCharm. Середовище розробки [Електронний ресурс] – Режим доступу: <https://itpro.ua/product/jetbrains-pycharm>(дата звернення: 19.05.2025)
9. Django Girls. Створення сайту на Django [Електронний ресурс] Режим доступу: <https://tutorial.djangogirls.org/uk/django/>(дата звернення: 19.05.2025)
10. Сучасний підручник JavaScript. Навчальний ресурс [Електронний ресурс] – Режим доступу: <https://uk.javascript.info/>(дата звернення: 19.05.2025)
11. Stack Overflow. Онлайн-спільнота програмістів [Електронний ресурс] – Режим доступу:<https://stackoverflow.com>(дата звернення: 20.05.2025)

12. Чучаєв А. П. Управління ІТ-проектами. – Київ : КНУ, 2019. – 320 с.
13. Шевченко В. І. Проектування інформаційних систем : підручник. – Харків : ХНУРЕ, 2021. – 280 с.
14. PyCharm: Документація користувача [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/pycharm/documentation/> . (дата звернення: 20.05.2025)
15. Гринь, О. М. Розробка програмних проєктів у середовищі PyCharm : навч. посіб. / О. М. Гринь. – Київ : КНУ, 2022. – 97 с.
16. Кузнецова І. В. Розробка програмного забезпечення засобами Python. – Харків : ХНУРЕ, 2023. – 212 с.
17. Django documentation [Електронний ресурс]. – Режим доступу: <https://docs.djangoproject.com/> (дата звернення: 20.05.2025).
18. Django for Professionals. William S. Vincent. – 3rd ed. – WelcomeToCode, 2022. – 400 p.
19. MySQL Documentation [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/> (дата звернення: 21.05.2025).
20. Когут, І. В. Проектування баз даних у середовищі MySQL Workbench : метод. рек. / І. В. Когут. – Львів : ЛНУ ім. І. Франка, 2021. – 56 с.
21. Ханенко Н. С. Основи баз даних : навч. посіб. – Дніпро : НГУ, 2019. – 185 с.
22. Жук Ю. О. Основи HTML, CSS та JavaScript : навч. посіб. – Київ : Ліра-К, 2022. – 198 с.
23. MySQL Workbench: Офіційна документація [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/workbench/en/> . (дата звернення: 21.05.2025)
24. Канер, С. Тестування програмного забезпечення: фундаментальні принципи : пер. з англ. / С. Канер, Дж. Фолкс, Г. Нгуєн. – Львів : БаК, 2017. – 320 с.

25. Шевченко, І. В. Якість програмного забезпечення та тестування : навч. посіб. / І. В. Шевченко. – Харків : ХНУРЕ, 2022. – 145 с.

ДОДАТКИ

Додаток А. Реалізація функцій

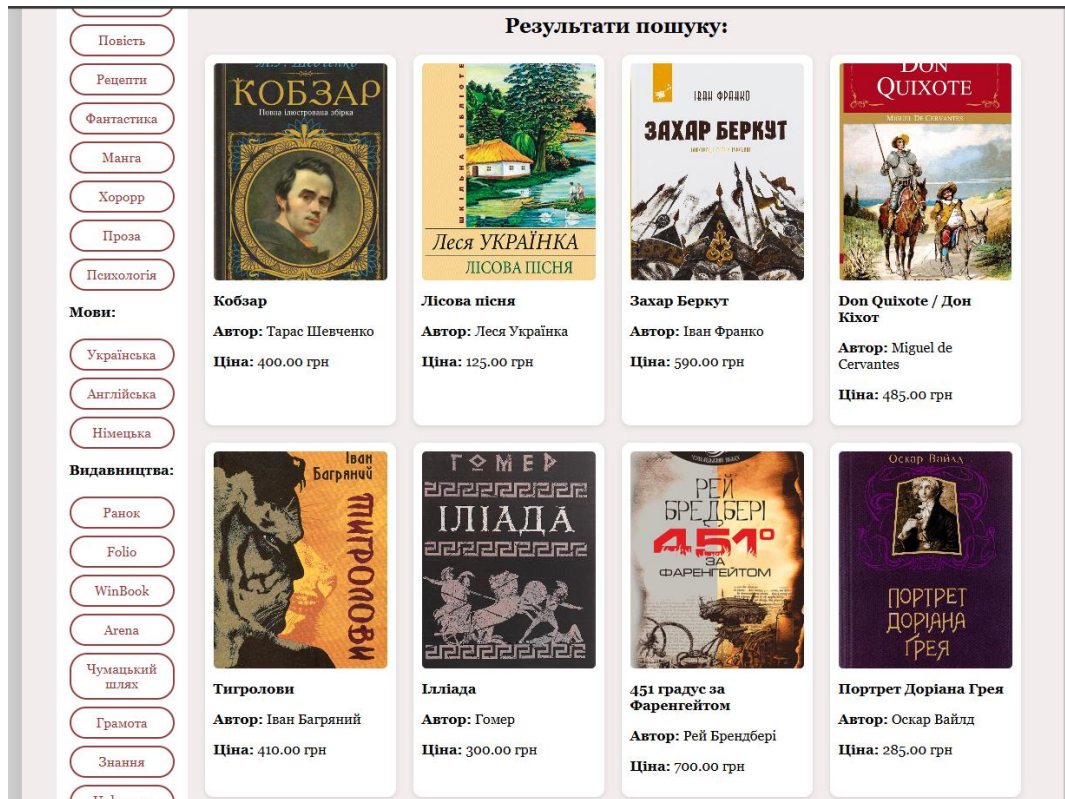


Рисунок А.1 — Сітка книг

Реєстрація

Логін:

Ім'я:

Прізвище:

Email:

Телефон:

Пароль:

- Мінімум 8 символів
- Хоча 6 одна цифра
- Хоча 6 один спец. символ (!@#\$%&...)

Підтвердження пароля:

[Вже маєте акаунт? Увійти](#)

Рисунок А.2 — Реєстрація на сайті

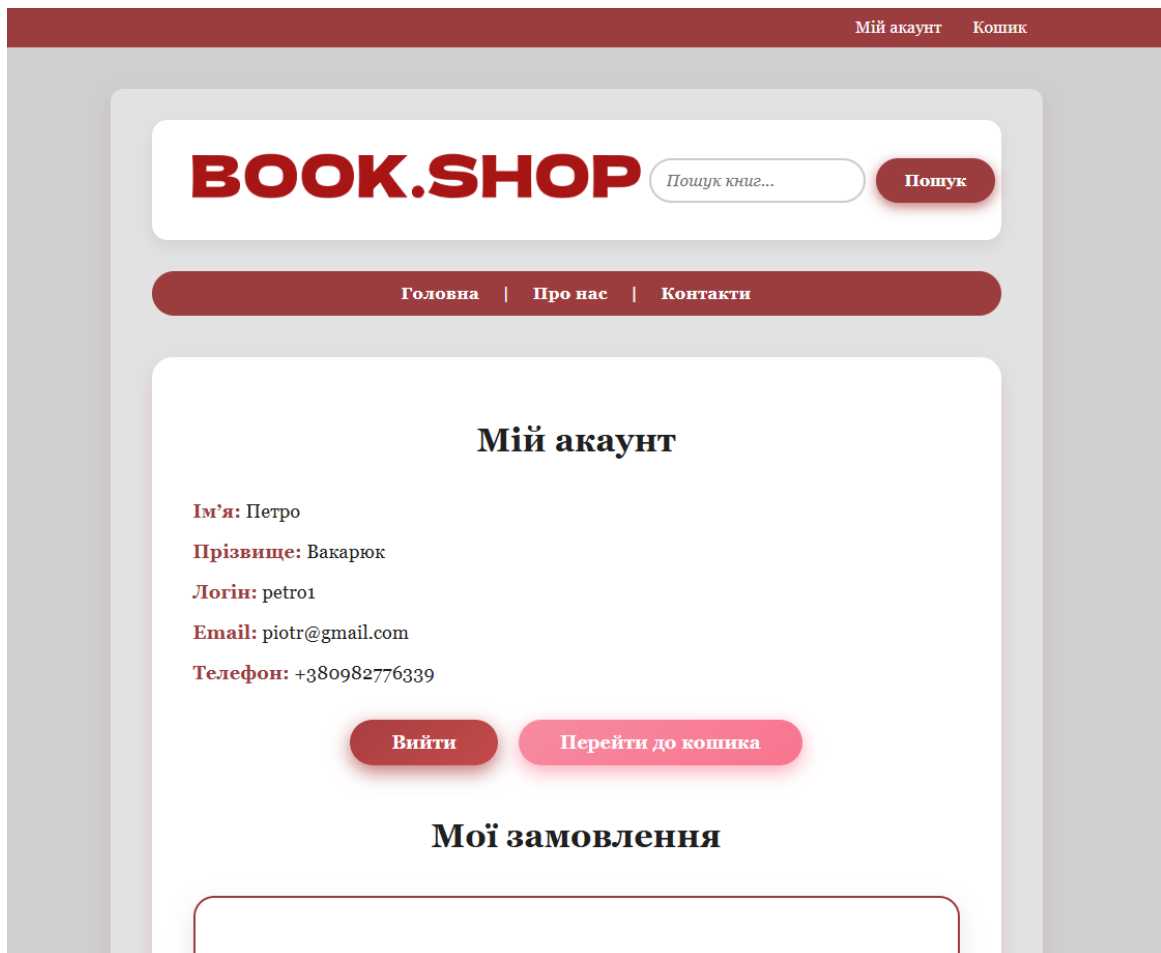


Рисунок А.3 — Мій акаунт



Рисунок А.4 — Картка книги з рекомендованими книгами

Додаток Б. Фрагменти коду***Б.1. Фрагмент коду для налаштування дизайну картки книги:******CSS:***

```
.book-card {  
  
    width: 230px;  
  
    background: #ffffff;  
  
    border-radius: 12px;  
  
    box-shadow: 0 6px 18px rgba(0, 0, 0, 0.1);  
  
    padding: 16px;  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    font-family: 'Georgia', serif;  
  
    transition: transform 0.3s, box-shadow 0.3s;  
  
    cursor: pointer;  
  
    margin: 15px;  
  
}  
  
.book-card:hover {  
  
    transform: translateY(-5px);  
  
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.15);
```

```
}
```

```
.book-card img {  
  width: 100%;  
  height: 260px;  
  object-fit: cover;  
  border-radius: 10px;  
  margin-bottom: 12px;  
}
```

```
.book-card h4.book-title {  
  font-size: 1.1em;  
  color: #9b3d3f;  
  margin: 6px 0;  
  text-align: center;  
}
```

```
.book-card p {  
  margin: 4px 0;  
  font-size: 0.95em;  
  color: #333;
```

```
text-align: center;

}

.book-card .book-description {

font-size: 0.85em;

color: #666;

height: 3.6em;

overflow: hidden;

text-overflow: ellipsis;

}

.book-card .book-price {

font-weight: bold;

color: #1f1f1f;

margin-top: auto;

}

.book-card .add-to-cart {

margin-top: 12px;

background: #9b3d3f;

color: white;
```

```
border: none;

padding: 10px 20px;

border-radius: 25px;

font-size: 0.95em;

font-family: 'Georgia', serif;

transition: background 0.3s, transform 0.3s;

}
```

```
.book-card .add-to-cart:hover {

background: #7c2d30;

transform: scale(1.05);

}
```

Б.2. Фрагмент коду для RegisterForm

```
from django import forms

from django.contrib.auth.forms import UserCreationForm

from .models import CustomUser

class RegisterForm(UserCreationForm):

    email = forms.EmailField(required=True)

    phone = forms.CharField(max_length=20, required=True)

    first_name = forms.CharField(max_length=30, required=True,
label="Имя")
```

```
last_name = forms.CharField(max_length=30, required=True,  
label="Прізвище")
```

```
class Meta:  
  
    model = CustomUser  
  
    fields = [  
  
        'username',  
  
        'first_name',  
  
        'last_name',  
  
        'email',  
  
        'phone',  
  
        'password1',  
  
        'password2',  
  
    ]
```

Б.3. Функція для реєстрації

```
def register(request):  
    if request.method == 'POST':  
        form = RegisterForm(request.POST)  
        if form.is_valid():  
            form.save()  
            return redirect('login')  
    else:  
        form = RegisterForm()  
    return render(request, 'registration/register.html', {'form': form})
```

Б.4. Фрагмент коду, що показує замовлення користувача на сторінці «Мій акаунт»

```
@login_required
def profile_view(request):
    user_orders = Order.objects.filter(user=request.user).order_by('-
created_at')
    return render(request, 'accounts/profile.html', {
        'user': request.user,
        'orders': user_orders
    })
```

Б.5. Додавання книги у кошик через AJAX

```
@login_required
@require_POST
def add_books_to_cart(request):
    try:
        data = json.loads(request.body)
        books = data.get('books', [])
    except json.JSONDecodeError:
        return JsonResponse({'success': False, 'error': 'Невірний формат
даних'})

    if not books:
        return JsonResponse({'success': False, 'error': 'Список книг
порожній'})

    cart, _ = Cart.objects.get_or_create(user=request.user)

    for book_info in books:
        book_id = book_info.get('id')
        quantity = book_info.get('quantity', 1)
```

```

if not book_id or quantity <= 0:
    continue

try:
    book = Book.objects.get(pk=book_id)
except Book.DoesNotExist:
    continue

cart_item, created = CartItem.objects.get_or_create(cart=cart,
book=book)

if not created:
    cart_item.quantity += quantity
else:
    cart_item.quantity = quantity
cart_item.save()

return JsonResponse({'success': True})

```

Б.6. Додавання кількох книг у кошик одночасно

```

@csrf_exempt

@require_POST

@login_required

def add_multiple_to_cart(request):

    try:

        data = json.loads(request.body)

        books = data.get('books', [])

    except json.JSONDecodeError:

```

```
return JsonResponse({'success': False, 'error': 'Некоректний JSON'})
```

```
cart, _ = Cart.objects.get_or_create(user=request.user)
```

```
for book_info in books:
```

```
    book_id = book_info.get('id')
```

```
    quantity = book_info.get('quantity', 1)
```

```
    book = get_object_or_404(Book, pk=book_id)
```

```
    cart_item, created = CartItem.objects.get_or_create(cart=cart,
book=book)
```

```
    if not created:
```

```
        cart_item.quantity += quantity
```

```
    else:
```

```
        cart_item.quantity = quantity
```

```
    cart_item.save()
```

```
return JsonResponse({'success': True})
```

Б.7. JavaScript-функція автопрокрутки банерів

```
document.addEventListener("DOMContentLoaded", function () {
    const imageCarousel = document.getElementById('image-carousel');
    if (!imageCarousel) return;
```

```

const banners = imageCarousel.children;
let currentBannerIndex = 0;

function scrollToBanner(index, smooth = true) {
  const banner = banners[index];
  if (banner) {
    imageCarousel.scrollTo({
      left: banner.offsetLeft,
      behavior: smooth ? 'smooth' : 'auto'
    });
  }
}

setInterval(() => {
  currentBannerIndex++;
  if (currentBannerIndex >= banners.length) {
    currentBannerIndex = 0;
  }
  scrollToBanner(currentBannerIndex);
}, 3500); // кожні 3.5 секунди
});

```

Б.8. Видалення книги з кошика

```

@require_POST

def remove_from_cart(request, item_id):

    if request.user.is_authenticated:

        try:

            item = CartItem.objects.get(id=item_id, cart__user=request.user)

            item.delete()

```

```
        messages.success(request, f'«{item.book.title}» видалено з кошика.")
except CartItem.DoesNotExist:
    messages.error(request, "Товар не знайдено в кошику.")
else:
    cart = request.session.get('cart', {})
    if str(item_id) in cart:
        del cart[str(item_id)]
        request.session['cart'] = cart
        messages.success(request, "Книга видалена з кошика.")
    else:
        messages.error(request, "Книга не знайдена в сесійному кошику.")
return redirect('cart')
```

Б.9. Оновлення кількості книг у кошику

```
@require_POST
@login_required
def update_cart_quantity(request, item_id):
    try:
        item = CartItem.objects.get(id=item_id, cart__user=request.user)
        quantity = int(request.POST.get('quantity', 1))
        if quantity > 0:
            item.quantity = quantity
```

```

    item.save()

    messages.success(request, f"Кількість \"{item.book.title}\" оновлено.")

else:

    item.delete()

    messages.info(request, f"Книгу «{item.book.title}» видалено з
кошика.")

except (CartItem.DoesNotExist, ValueError):

    messages.error(request, "Помилка оновлення кількості.")

return redirect('cart')
```

Б.10. Оформлення замовлення через AJAX

```

@login_required

def place_order(request):

    if request.method == 'POST':

        try:

            cart_data = json.loads(request.body.decode('utf-8'))

        except json.JSONDecodeError:

            return HttpResponseRedirect('Невірний формат JSON')

    form = OrderForm(request.POST)

    if form.is_valid() and cart_data.get('items'):

        order = form.save(commit=False)

        order.user = request.user
```

```

order.save()

for item in cart_data['items']:

    try:

        book = Book.objects.get(pk=item['book_id'])

        OrderItem.objects.create(order=order, book=book,
quantity=item['quantity'])

    except Book.DoesNotExist:

        continue

    return JsonResponse({'success': True, 'message': 'Замовлення
оформлено'})

else:

    return JsonResponse({'success': False, 'error': 'Невірна форма або
порожній кошик'})

else:

    form = OrderForm()

return render(request, 'catalog/place_order.html', {'form': form})

```

Б.11. Обробляє звичайне оформлення замовлення із кошика, якщо форма відправлена

```

@login_required(login_url='/accounts/login/')
def make_order(request):

```

```

cart, created = Cart.objects.get_or_create(user=request.user)
cart_items = CartItem.objects.filter(cart=cart)
total_price = sum(item.book.price * item.quantity for item in cart_items)

```

```

if request.method == 'POST':
    form = OrderForm(request.POST)
    if form.is_valid():
        # створення замовлення
        ...

```

Б.12. Функція для перевірки чи користувач зареєстрований

```

<script>
function checkLoginBeforeOrder() {
    const isAuthenticated = {{ request.user.is_authenticated|yesno:"true,false"
}};
    if (isAuthenticated) {
        window.location.href = "{% url 'make_order' %}";
    } else {
        document.getElementById('authPromptModal').style.display = 'block';
    }
}

function redirectToLogin() {
    window.location.href = "{% url 'login' %}?next={% url 'make_order'
%}";
}

function closeModal() {
    document.getElementById('authPromptModal').style.display = 'none';
}
</script>

```