МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ РОБОТА**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни Web-технології Ч.2

на тему:

**«CMS-система інтернет-магазину книг “Psalterium”»**

студента II курсу групи ІПЗ-20-4

спеціальності 121 «Інженерія програмного забезпечення»

Кормиша Романа Івановича

(прізвище, ім’я та по-батькові)

Керівник: Морозов А.В .

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Граф М.С .

(підпис) (прізвище та ініціали) \_\_\_Кузьменко О.В\_ .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чижмотря О.Г .

(підпис) (прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о зав кафедри ІПЗ

\_\_\_\_\_\_\_\_\_\_\_\_ А.В Морозов

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2021 р.

ЗАВДАННЯ

НА КУРСОВУ РОБОТУ СТУДЕНТУ

Кормишу Роману Івановичу

1. Тема роботи: Розробка інтернет-магазину книг

керівник роботи: Морозов А.В .

1. Строк подання студентом: “ ” 2021р.
2. Вихідні дані до роботи: Розробити інтернет-магазин книг – розробити та реалізувати систему адміністрування для сайту
3. Зміст розрахунково-пояснювальної записки (перелік питань, що підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми
       4. Опис роботи програми
4. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація до КР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Посилання на репозиторій: https://gitlab.com/2020-2024/ipz-20-4/kormysh-roman/course-work-2.git

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
| 1,2 | Чижмотря О.Г., ст. викладач каф. ІПЗ |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ ” 2021 р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсової роботи | Термін виконання етапів роботи | Примітки |
| 1 | Постановка задачі | 4.12.21 – 5.12.21 | Виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 5.12.21 – 6.12.21 | Виконано |
| 3 | Формулювання технічного завдання | 6.12.21 – 8.12.21 | Виконано |
| 4 | Опрацювання літературних джерел | 8.12.21 – 12.12.21 | Виконано |
| 5 | Проектування структури | 12.12.21 – 14.12.21 | Виконано |
| 6 | Написання програмного коду | 14.12.21 – 31.12.21 | Виконано |
| 7 | Відлагодження | 01.01.22 –  10.01.22 | Виконано |
| 8 | Написання пояснювальної записки | 05.01.22 –  10.01.22 | Виконано |
| 9 | Захист | 13.01.22 | Виконано |

**Студент***\_\_\_\_\_\_\_\_\_\_\_\_\_\_* Кормиш Р.І

(підпис) (прізвище та ініціали)

**Керівник проекту** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Морозов А.В

(підпис) (прізвище та ініціали)

**РЕФЕРАТ**

Пояснювальна записка до курсового проекта на тему «Розробка інтернет магазину книг» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 33 сторінках друкованого тексту.

Пояснювальна записка має 1 сторінки додатків. Список використаних джерел містить 12 найменувань і займає 1 сторінку. В роботі наведено 23 рисунів. Загальний обсяг роботи – 36 сторінок.

У першому розділі було обґрунтовано створення сайту на тему Розробка інтернет-магазину книг.

У другому розділі проведено проектування і розробка сайту.

У третьому розділі проведено тестування програмного продукту.

Висновок містить в собі результати виконаної роботи при створенні сайту на тему «Розробка інтернет-магазину книг».

У додатку представлений лістинг розробленого програмного продукту.

Ключові слова:

MYSQL, JAVASCRIPT, PYTHON, FAST API, VUE JS, PET-ПРОЕКТ, WEB-САЙТ, MVC, СЕРВЕР, АДМІН, АВТОРИЗАЦІЯ, МАГАЗИН, ІНТЕРНЕТ, ДАНІ.

# 

**Зміст**

[**ВСТУП** 7](#_Toc92540836)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ 8](#_Toc92540837)

[**1.1** **Аналіз задачі, засобів та методів її вирішення** 8](#_Toc92540838)

[**1.2** **Аналіз існуючого програмного забезпечення за тематикою курсової роботи.** 8](#_Toc92540839)

[**Висновки до першого розділу :** 14](#_Toc92540840)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 16](#_Toc92540841)

[**2.1 Проектування загального алгоритму роботи веб-додатку** 16](#_Toc92540842)

[Загальна схема роботи програми : 17](#_Toc92540843)

[**2.2 Розробка функціональних алгоритмів роботи програми** 18](#_Toc92540844)

[**2.3 Розробка програмного забезпечення** 19](#_Toc92540845)

[РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ 24](#_Toc92540846)

[**3.1 Опис роботи з програмним додатком (Опис інтерфейсу)** 24](#_Toc92540847)

[**3.2** **Тестування роботи програмного забезпечення** 30](#_Toc92540848)

[**Висновки до 3 розділу :** 31](#_Toc92540849)

[**ВИСНОВКИ** 32](#_Toc92540850)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 33](#_Toc92540851)

[**ДОДАТКИ** 34](#_Toc92540852)

**ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

БД – База даних

КР – курсова робота

MVC – Модель–вигляд–контролер (або Модель–представлення–контролер, Model-view-controller, MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

ПЗ – Програмне забезпечення

# **ВСТУП**

У цій курсовій роботі буде наведено процес розробки інтернет-магазину книг. Інтернет-магазин книг – це інтернет-магазин, де корситувач може замовляти книги. Взагалом останнім часом інтернет-магазини ствють все більш і більш популярними, тому що дозволяють здійснювати покупки не виходячи з дому, тому дана тема курсової є актуальною.

**Метою** курсової роботи є дослідження особливостей проектування та реалізації cистеми адміністрування інтернет-магазину книг.

Завданням на курсову роботу є:

* аналіз теоретичних засад проектування та реалізації CMS;
* аналіз існуючих веб-сайтів на обрану вами тематику, визначення унікальності;
* визначення інформаційних потреб предметної області дослідження;
* розробка адаптивного інтерфейсу веб-сайту за допомогою HTML5, CSS3, JavaScript;
* розробка математичної та алгоритмічної моделі функціонування системи на основі БД;
* проектування бази даних за визначеною предметною областю;
* написання серверної частини;
* реалізація CMS.

**Об’єктом дослідження** є методи та засоби розробки CMS для інтернет-магазину книг.

**Предмет дослідження** є використання веб-технологій для забезпечення інформаційних потреб в сферах освіти та саморозвитку.

# РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## **Аналіз задачі, засобів та методів її вирішення**

Завдання полягає в тому, що потрібно:

1. Визначити потреби звичайного користувача
2. Зрозуміти основні принципи та алгротитми роботи інтернет-магазину
3. Продумати можливість додавання книжок та їх повного редагування
4. Реалізувати сортування та пошук книг
5. Реалізувати userfriendly інтерфейс
6. Продумати розподілення ролей та можливість створення нових адмінів
7. Створити корректну базу даних
8. Продумати дизайн сайту
9. Реалізувати адаптивність для різних пристроїв

Для розробки сайту буде використано мови програмуваня Python(+фраймворк Fast API) та JS(+фреймворк Vue js)

Для розробки КР використано такі середовища розробки, як PyCharm та WebStorm, тому що вони підтримують всі необхідні в ході розробки інструмети.

Для написання серверної частини сайту було обрано мову програмування Python і фреймворк FastAPI. Python доволі легка мова програмування, на якій можна швидко створювати проекти, а FastAPI використовує найновітніші практики, які дозволяють швидко та зручно асинхронні проекти.

У якості БД було обрано Sqlite.

Для написання клієнтської частини сайту було обрано фреймворк Vue js та мову програмування JavaScript. Такий вибір було здійснено, бо Vue дозволяє створювати сайти, які працюють повністю асинхронно і не потребують перезавантаження сторінки.

Також, для стилізування сайту було обрано CSS-фреймворк Bootstrap 5, оскільки він надає вже готові стилі і при цьому дозволяє повну їхню кастомізацію.

## **Аналіз існуючого програмного забезпечення за тематикою курсової роботи.**

При аналізі вже існуючих *веб-сайтів* за тематикою курсової роботи було виявлено декілька проектів. Всі вони схожі і мають подібну реалізацію, адже всі вони є інтернет магазинами книжок, які призначені для продажу книжок.

Aналогічні інтренет-магазини за заданою темою:

**Yakaboo**

Посилання : https://www.yakaboo.ua/

Розглянемо даний аналог:

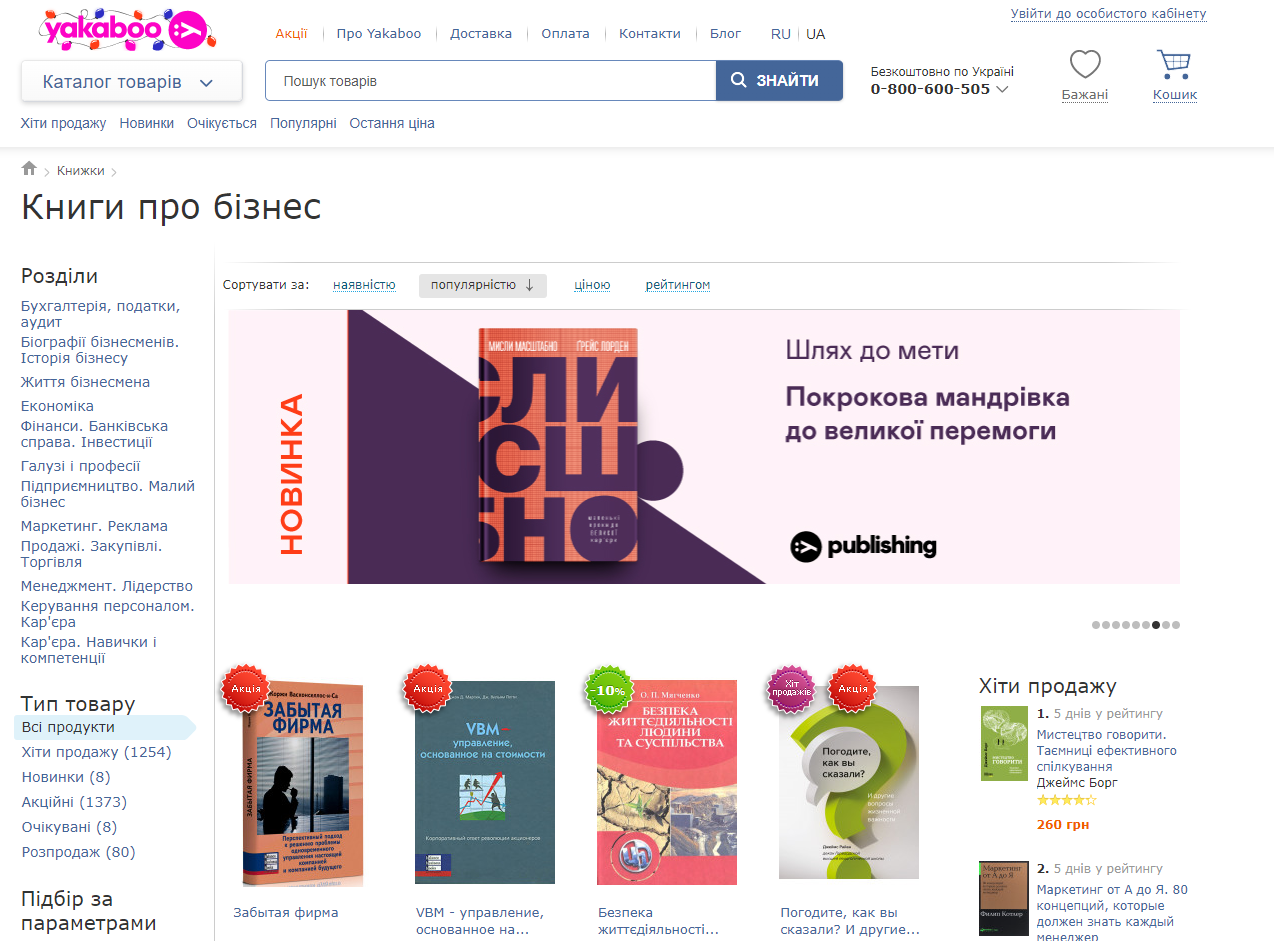


Рис 1.1 Yakaboo

Інтернет-магазин Yakaboo є одним із найпопулярніших інтернет-магазинів про продажу книг, адже має інтуїтивно зрозумілий користувацький інтріейс та широкий асортимент книг. Також має широкий вибір параметрів для сортування товарів. Серед мінусім можна виділити відсутність адаптиву.

Основні плюси:

* Інтуїтивно зрозумілий користувацький інтерфейс
* Широкий вибір параметрів для сортування товарів
* Широкий асортимент книг

Основні мінуси:

* Відсутність адаптиву

**Amazon**

Посилання : https://www.amazon.com/

Розглянемо даний аналог:

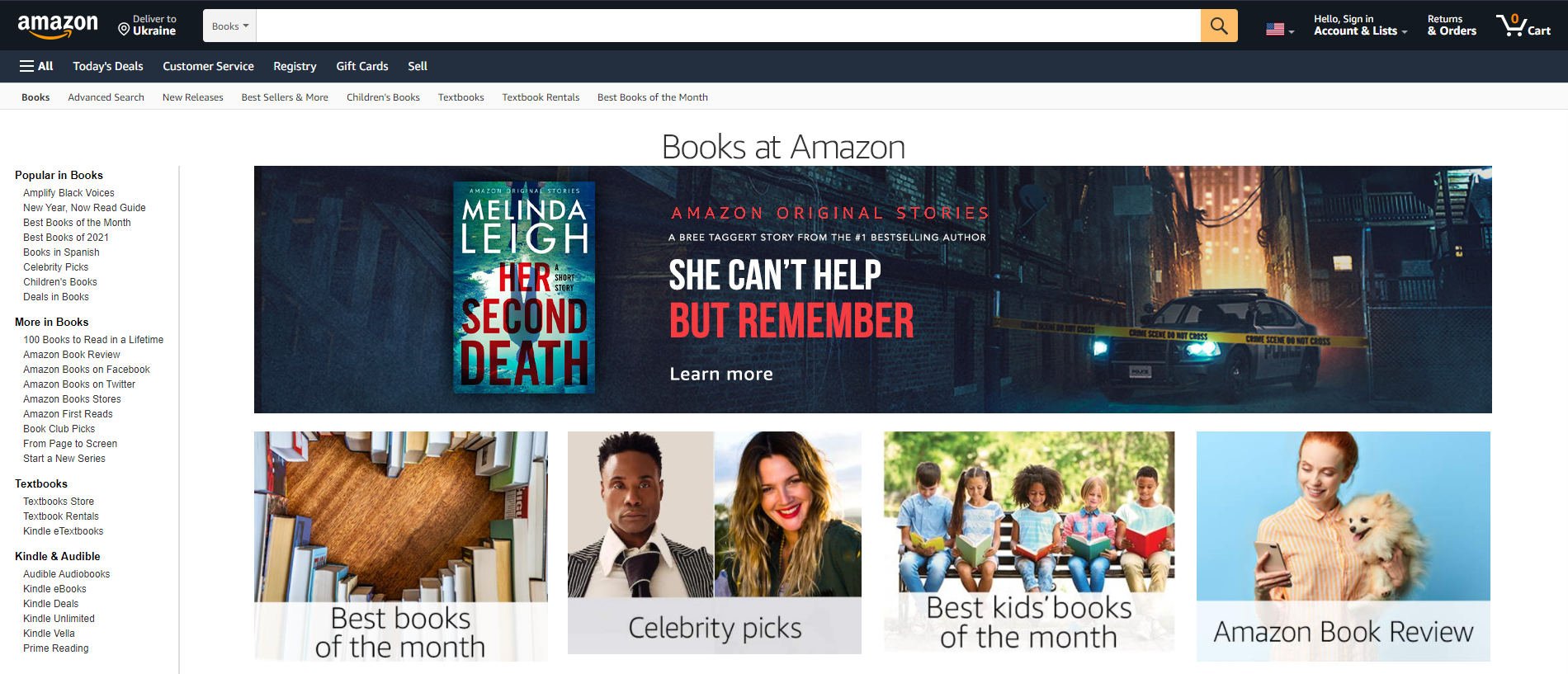


Рис 1.2 Amazon

Інтернет-магазин Amazon є одним із найперших інтернет-магазинів.

Спочатку даний сайт розроблявся саме для продажу книг, а вже згодом розширювавля для інших видів товарів. Також має широкий вибір параметрів для сортування товарів. Серед мінусім можна виділити відсутність адаптиву та незначно надмірну уількість елементів користувацького інтерфесу.

Основні плюси:

* Широкий вибір параметрів для сортування товарів
* Широкий асортимент книг
* Один з найперших інтернет-магазин книг

Основні мінуси:

* Відсутність адаптиву
* Незначно надмірний користувацький інтерфейс

**British Book**

Посилання: https://www.britishbook.ua/

Розглянемо даний аналог:

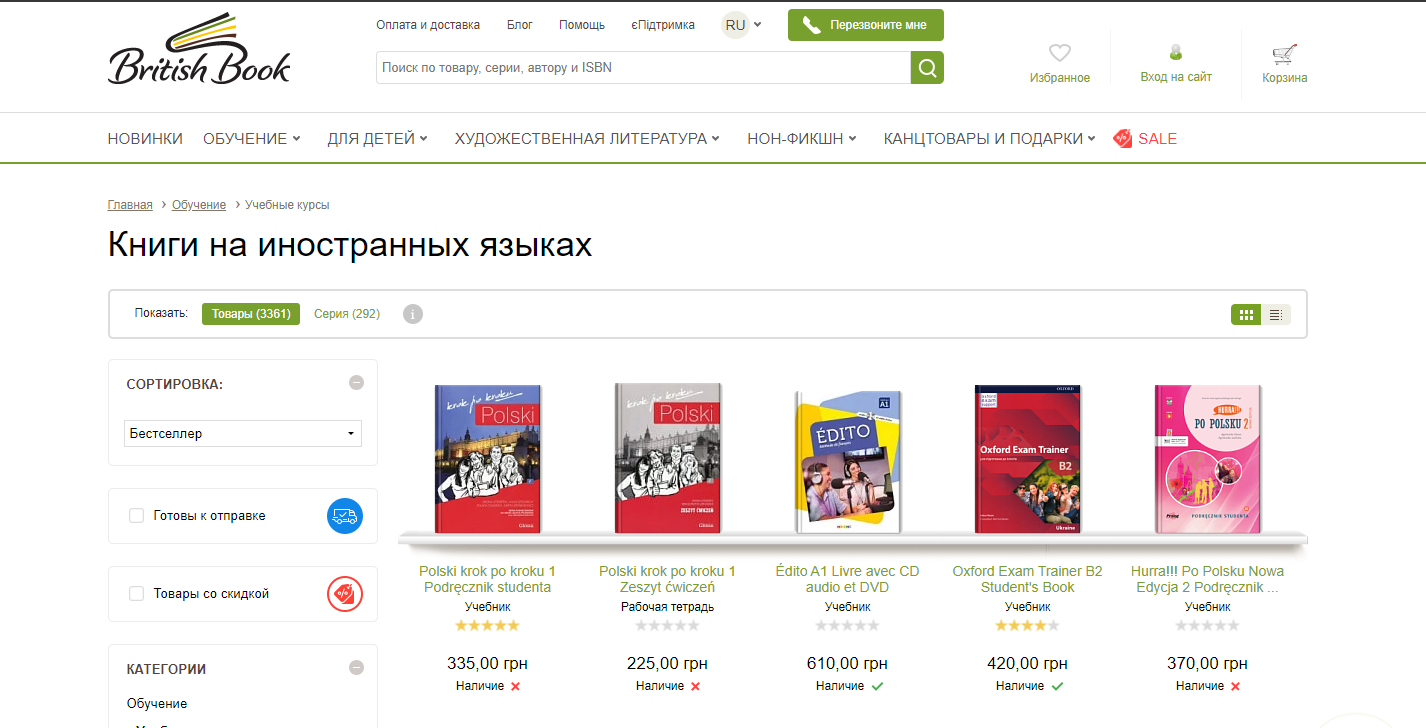


Рис 1.3 British Book

Інтернет-магазин British Book це інтернет-магаизин, який має стильний дизайн та можливості попередніх розглянутих продуктів. Навідміну від попередньо-розглянутих сайтів, має також адаптив.

Основні плюси:

* Всі, що і в попередньо-розглянутих інтерент-магазинів
* Адаптив для різних пристроїв

Основні мінуси:

* Не такий відомий як попередньо-розглянуті магазини

Актуальним напрямком реалізації власного інтернет-магазину є створення веб-сайту з бізнес логікою подібною до реалізованої у всіх представлених продуктах, реалізувати користувацький інтерфейс по принципам yakaboo, уникаючи надмірності Amazon, але послідувати основним аспектам інтерфейсу цього продукту. Також продумати сучасний дизай сайту, звертаючи увагу до дизайну British Book та реалізувати адаптив.

## **1.3 Технічне завдання на курсову роботу**

**1. Загальне положення**

***1.1Найменування програмного засобу***

Повне найменування програмної системи: "Розробка інтернет-магазину книг" (надалі "веб-додаток"). Коротка назва програмної системи - " Інтернет-магазин"

***1.2. Призначення розробки та область застосування***

Веб-сайт "Інтернет-магазин книг" призначений для розміщення каталогу книг, їх пошуку, та продажу. Система може бути впроваджена у роботу інтернет-магазину. Веб-сайт " Інтернет-магазин книг" дозволить швидко переглядати інформацію про книги, та забезпечить керівництву підприємства своєчасною та повною інформацією про актуальний стан замовлення покупок інтернет-магазину у повному обсязі, а також надасть користувачам можливість вибору книг різних видавництв в залежності від їх смаку.

***1.3. Найменування розробника та замовника.***

Розробник даного продукту - студент групи ІПЗ-20-4 Кормиш Роман Іванович (надалі "розробник"). Замовник програмного продукту – кафедра інженерія програмного забезпечення Житомирського державного технологічного університету в межах виконання курсової з дисципліни «Web-тееології» Чижмотря Олена Геннадіївна, Морозов Андрій Васильович **2. Підстава для розробки**

***2.1. Документ на підставі якого ведеться розробка***

Робота ведеться на підставі навчального плану за напрямом 121 «Інженерія програмного забезпечення».

**3. Вимоги до програми**

***3.1. Вимоги до функціональних характеристик.***

***3.1.1. Загальні вимоги***

Веб-додаток має забезпечувати:

• можливість дистанційної роботи з робочих станцій локальної та глобальної мережі підприємства;

• постійний доступ користувачів веб-додатку;

• оформлення замовлення;

• організацію управління сайтом;

• можливість доступ до бази даних;

***3.1.2. Склад виконуваних функцій***

Розробити інтернет-магазин книг, що підтримує виконання наступних операцій:

1. Пошук книг.
2. Зміна інформації про книгу.
3. Додавання запису про книгу:

* Назва
* Дата видання
* Ціна
* Автор
* ISBN
* Жанри
* Короткий опис

1. Фільтрація книг за параметрами:

* Назва
* Автор
* Дата видання
* Кількість
* Ціна

1. Перегляд інформації про книги

***3.1.3. Організація вхідних і вихідних даних***

Вхідними даними є інформація про книгу (Назва, Автор, ціна, кількість, дата видання, ISBN, опис)

***3.1.4. Часові характеристики і розмір пам'яті, необхідної для роботи програми.***

* Час реакції програми на дії користувача (маніпуляції з пристроями введення даних) не повинен перевищувати 0,25 с.
* Час виконання команд меню не більше 1 с.
* Відображення масивів даних за запитами не більше 3 хвилин.
* Доступність БД – 90% цілодобово.
* Операції з’єднання з БД не більше 1 хвилини.
* Обсяг оперативної пам'яті, необхідний для роботи програми не менше 1Гб.
* Дисковbй простір, необхідний для збереження програми і файлів даних не більше 300 Мбат для робочої станції та 20 ГБайт.
* Інсталяційний пакет програми, що містить у складі БД не повинні перевищувати 100 Мбайт.

***3.2. Вимоги до методів рішення і мов програмування***

Вибір методів рішення здійснюється розробникам без узгодження з замовником.

***3.2.1. Вимоги до системи програмних засобів.***

OpenServer, PHP 5, MySQL, HTML 5, CSS 3, JavaScript: Вимоги до програмного забезпечення робочої станції: PHP Storm

## **Висновки до першого розділу :**

У ході виконання першого розділу було поставлено завдання КР, а саме який має бути функціонал і визначено актуальність теми. Також визначено мови і фреймворки для розробки та актуальний напрямок розробки власного продукту. Розглянуто існуючі програмні продукти зі схожою тематикою та більш детально проаналізовано деякі із них, а саме : Amazon, British Book та Yakaboo.

# РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **2.1 Проектування загального алгоритму роботи веб-додатку**

На даному етапі розробки розглянемо загальний алгоритм роботи інтернет-магазину. Загальна схема роботи приведена на рисунку 2.1.

Для початку роботи потрібно створити моделі бази даних. Потум спроектувати схеми з унаслідуванням, адже саме це потрібно для API, який буле реалізовуватися на мові програмування Python та його фреймворку Fast API.

Зголом потрібно реалізувати основні роути для виконання запитів, що дозволить протестувати отримання та обробку інформації отриманої від даного API. Потім реалівувати всі роути та сервіси для кожної сутності, з використання схем які ми описали раніше.

Для авторизації використаємо JWtoken, для забезпечення безпеки даних – хешування паролів.

Тепер спроектуємо візуальну частину. Для цього використаємо мову програмування JS та її фреймворк Vue js. Напишемо декілька компонентів та першу сторінку, з використанням vue-store, що протестувати з’єднання зі створеною API. Потім потрібно написати всі шаблони сторінок, стилі та інші компоненти з сторінками.

Для взаємодії з користувачем потрібно реалізувати “діалогові вікна” (форми). Для того, щоб гість, зареестрований користувач і адмін мали різні права, реалізуємо розділення на ролі з різними прававми. Де адмін має найбільші права (право створювати адмінів включно), користувач – набір повноцінних прав, а гісті може тільки переглядати деякі сторінки сайту.

При запуску сайту користувач буде бачити основну сторінуку, сторінку магазину. Для йому буду пропонуватися зареєструватися або аутентифікуватися. Після цих дій він отримає можливість замовляти книги та виставляти на продаж власні.

## Загальна схема роботи програми :

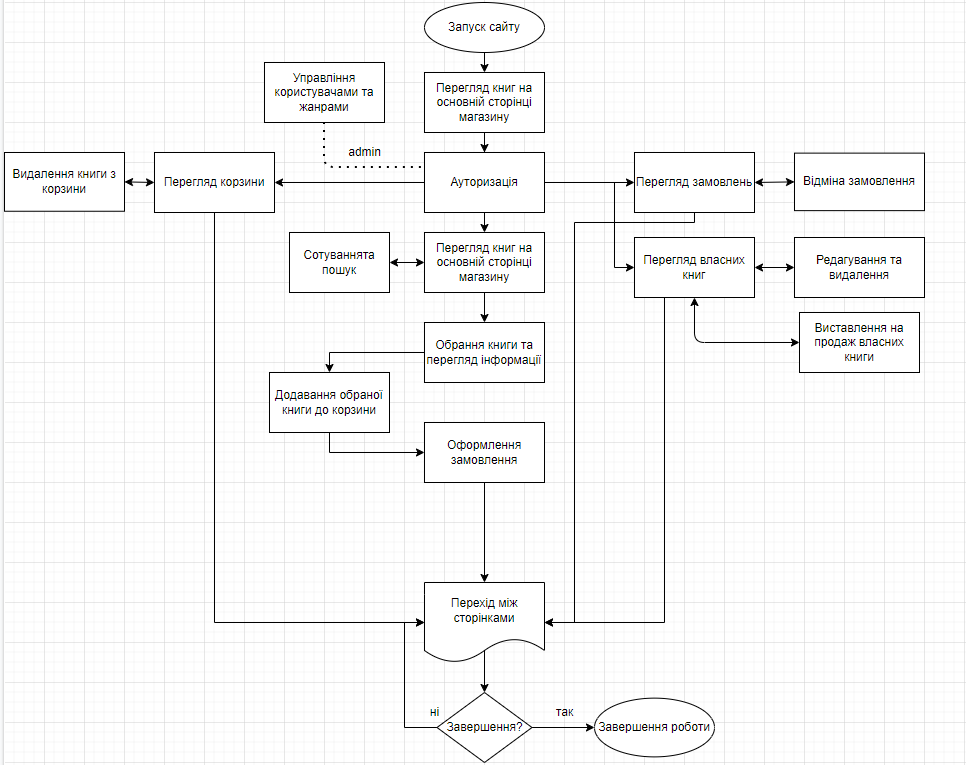


Рисунок 2.1 – Загальна схема роботи програми

## **2.2 Розробка функціональних алгоритмів роботи програми**

Залишилося розглянути детальніше алгоритми робити основних функцій.

Звичайно, основне призначення інтренет-магазину книг – продаж книг, тому розглянемо саме це. Для висталвення книги на продаж потрібно заповнити відповідну форму, після чого книга розміститься на сайті. При заповненні форми будуть з’являтися підзказки про коректність вводу. Після розміщення інший користувач зможе замовити цю книги в максимальній кількості вказаній продавцем. Користувач додає книгу в корзину, звідки можна оформити замовлення. Для цього потрібно заповнити відповідну форму, вказавши адрессу та дату прибуття, не раніше сьогоднішньої дати.

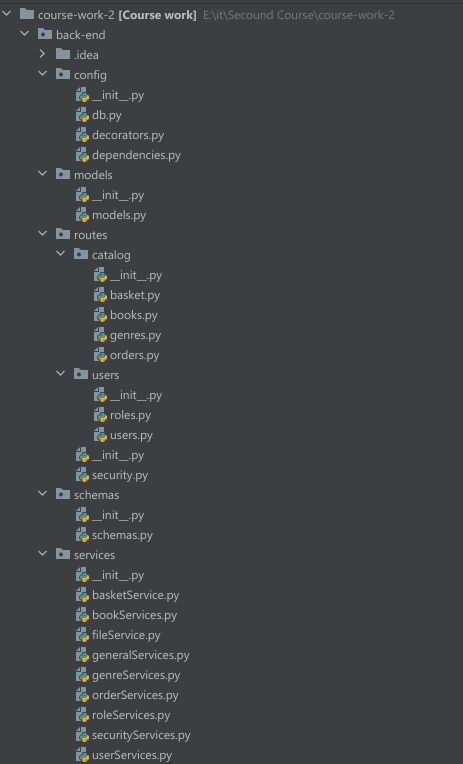
Для корректонсті обробки даних реалізована подвійна валідація, як бекенді так і на фронтенді. При будь-якій віправці форми висвітлбється вікно з оповіщення про статус успішності дії. При редагування дані автоматично заповнюють відповідні поля, для облегчення їх зміни.

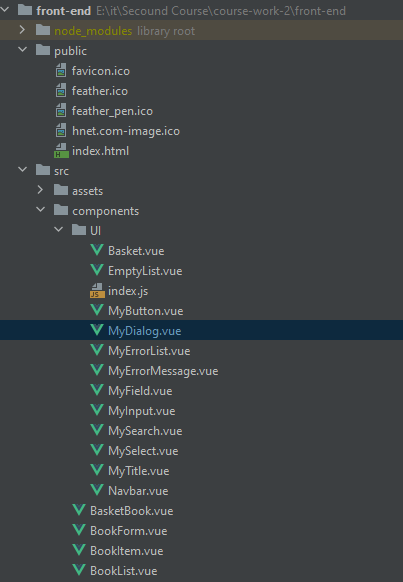
Основним алгоритмом кожної реалізованої функції є:

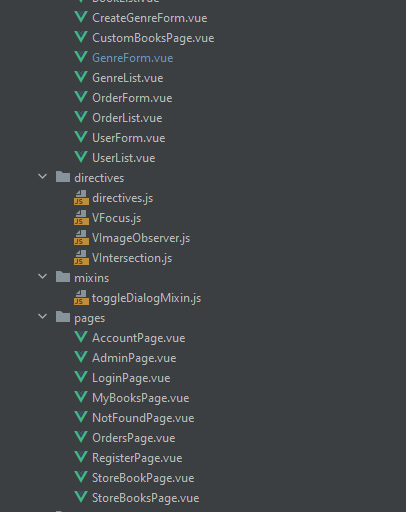
отримання даних на фронтенді -> створення запиту -> передача запиту на бекенд -> обробка отриманих даних у відповідних сервісах(взаємодія з БД) -> передача оброблених даних на фронтенд.

## **2.3 Розробка програмного забезпечення**

Структура рішення курсової роботи :





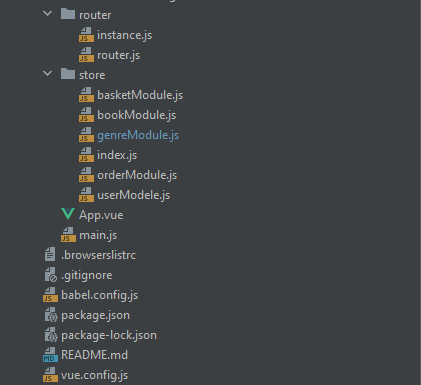


Рисунок 2.2 – Структура КР

Більша частина програмного забезпечення розробляється відповідно до алгоритму описаного вище, тому розглянемо цей принцип на прикладі реалізованих методів книжок. Методи книжок розмістимо в файлі bookServises, що згодом будуть використовуватись в books-routes, що надають можливість приймати запити з фроненду.

Розглянемо приклад коду створення книжки:

*def* create\_book(*db*: Session, *model*: schemas.BookCreate, *current\_user*: models.User) -> Any:  
 expression = \_model.ISBN == *model*.ISBN  
 generalServices.check\_in\_use\_expression(db=*db*, model=\_model, expression=expression)  
 book = \_model(  
 name=*model*.name,  
 author=*model*.author,  
 content=*model*.content,  
 price=*model*.price,  
 owner=*current\_user*,  
 count=*model*.count,  
 publication\_date=*model*.publication\_date,  
 ISBN=*model*.ISBN  
 )  
 *db*.add(book)  
 *db*.commit()  
 *return* book.id

Маємо метод для створення книжки, який повертає id новостворенної книги та приймає базу даних, з якою ми працюємо, схему з даними та поточного користувача. Використовуємо дані, які передаються в метод, та створюємо книгу. Книга додається в базу, що дає змогу отримати її id згодом.

Розглянемо роут для даногго методу:

@router.post('', response\_model=int)  
*async def* create\_book(*bookCreate*: schemas.BookCreate,*db*: Session = Depends(get\_db),  
 *current\_user*: models.User = Depends(get\_current\_user)):  
 *return* bookServices.create\_book(db=*db*, model=*bookCreate*, current\_user=*current\_user*)

Як модемо бачити код методу не великий, тому що весь функціонал винесений в сервіс.

Далі перейдемо до створення запиту з фронтеду:

async createBook({*state*, *commit*, *rootState*, *rootGetters*}) {  
 await *commit*('setFormLoading', true)  
 *rootState*.errors = []  
 const book = *state*.book  
 book.publication\_date = new *Date*(book.publication\_date.toLocaleString()).toISOString()  
 await instance  
 .post(`${*state*.defaultRoot}`, book, {headers: *rootGetters*.getHeaders})  
 .then(*response* => {  
 book.id = *response*.data

book.image = 'default.png'  
 *commit*('pushBook', book)  
 *commit*('clearBook')  
 })  
 .catch(*error* => {  
 *rootState*.errors.push(*error*.response.data.detail)  
 })  
 await *commit*('setFormLoading', false)  
},

Тут ми можемо бачити формування запиту, де instance містить основний шлях до нашого API

export const *instance* = *axios*.create({  
 baseURL: 'http://192.168.0.105:8000/api/',  
});

, що дозволяє уникнути дубляції коду. Виконується метод post, який відпраляє post запит на сервер та приймає як параметри частину шляху, в даному випадку defaultRoot(‘/books’), book – обєкт з даними про книгу config, який містить в headers токен доступу. Далі отриманий результат виконаня обробляється в then(), де ми виконуємо даданя даних до моделі книги та додаємо її в список книг, які показуються користувачу.

Інші запити виконуються по схожому алгоритму, тому вважаю надмірним демонтрувати їх код.

Також важливим компонетом веб додатку є аутентифікація, реалізована на основі JWT. Роглянемо її код:

@router.post("/token", response\_model=schemas.Token)  
*async def* login\_for\_access\_token(*form\_data*: OAuth2PasswordRequestForm = Depends(), *db*: Session = Depends(get\_db)):  
 user = authenticate\_user(db=*db*, email=*form\_data*.username, password=*form\_data*.password)  
 *if not* user:  
 *raise* HTTPException(  
 status\_code=status.HTTP\_401\_UNAUTHORIZED,  
 detail="Incorrect username or password",  
 headers={"WWW-Authenticate": "Bearer"},  
 )  
 access\_token\_expires = timedelta(minutes=ACCESS\_TOKEN\_EXPIRE\_MINUTES)  
 access\_token = create\_access\_token(  
 data={"sub": user.email, "role": user.role.name}, expires\_delta=access\_token\_expires  
 )  
 *return* {"access\_token": access\_token, "token\_type": "bearer"}

Вище приведений код роута для аутентифікації, який приймає форму з даними. В методі authenticate\_user виконується аутентифікація і якщо вона не успіша користувачу передається повідомлення про помилку.

Розглянемо дедальніше цей метод:

*def* authenticate\_user(*db*: Session, *email*: str, *password*: str) -> Any:  
 model = models.User  
 expression = model.email == *email* user = generalServices.get\_by\_expression(db=*db*, model=model, expression=expression)  
 *if not* verify\_password(*password*, user.password):  
 *return False  
 return* user

В даному методі ми звертаємося до бази даних і перевіряємо на співпадіння паролі та повертаємо відповідний реультат.

Після того як ми отримали результат аутентифікації, при успішності ми створюємо токен доступута записуємо в нього дані про email, role та дату закінчення токену. І в фіналі повертаємо на фронтенд результат виконання методу, а саме новостворений токен доступу.

## **Висновки до 2 розділу :**

Спроектовано загальну схему програми. Визначено та детально описано основні можливості програми. Пояснено алгоритми найважливіших методів інтернет-магазину та приведено код їх основних частин. Приведено приклади коду деяких методів. Роз’яснено деякі нюанси роботи програми. Показано структуру КР.

# РОЗДІЛ 3 ОПИС РОБОТИ З ДОДАТКОМ

## **3.1 Опис роботи з додатком (Опис інтерфейсу)**

Після запуску програми з’являється головна сторінка (Рис 3.1), де користувач бачить список книг та може тільки їх проглядати.

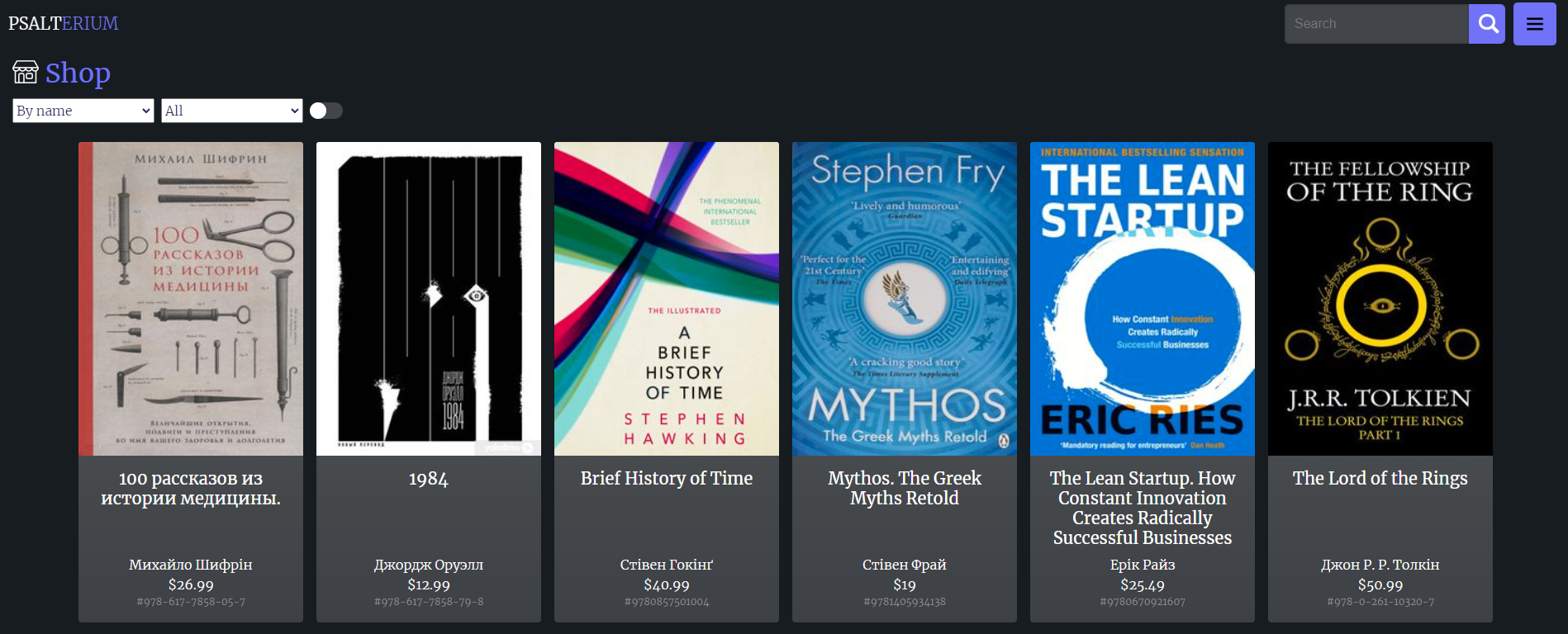


Рис 3.1 Головна сторінка

Головна сторінка має функції переходу до конкретних книжок, сортування та пошуку. Також реалізована динамічна пагінація у виді бескінечної ленти. У верхньому правому куті розміщений пошук та меню, де користувач може побачити наявні сторінки.

Перейдемо до конкретного алгоритму роботи х веб додатком.

Спочатку користувач обирає книгу, наприклад, книгу “Чистий код”. При натиску на карточку з відповідною книгою, він переходить на сторінку книги (Рис 3.3), де він може прочитати інформацію про книгу більш детально та додати деяку кількість книг до власної корзини. Після успішного додання книги до корзини, користувач бачить повідомлення:

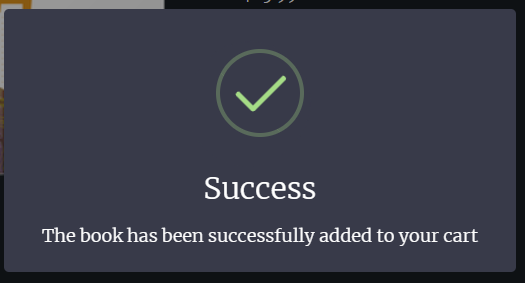


Рис 3.2 Сторінка конкретної книги

Далі користувач може переглядати інші книги та додавати до корзини або оформити замовлення.

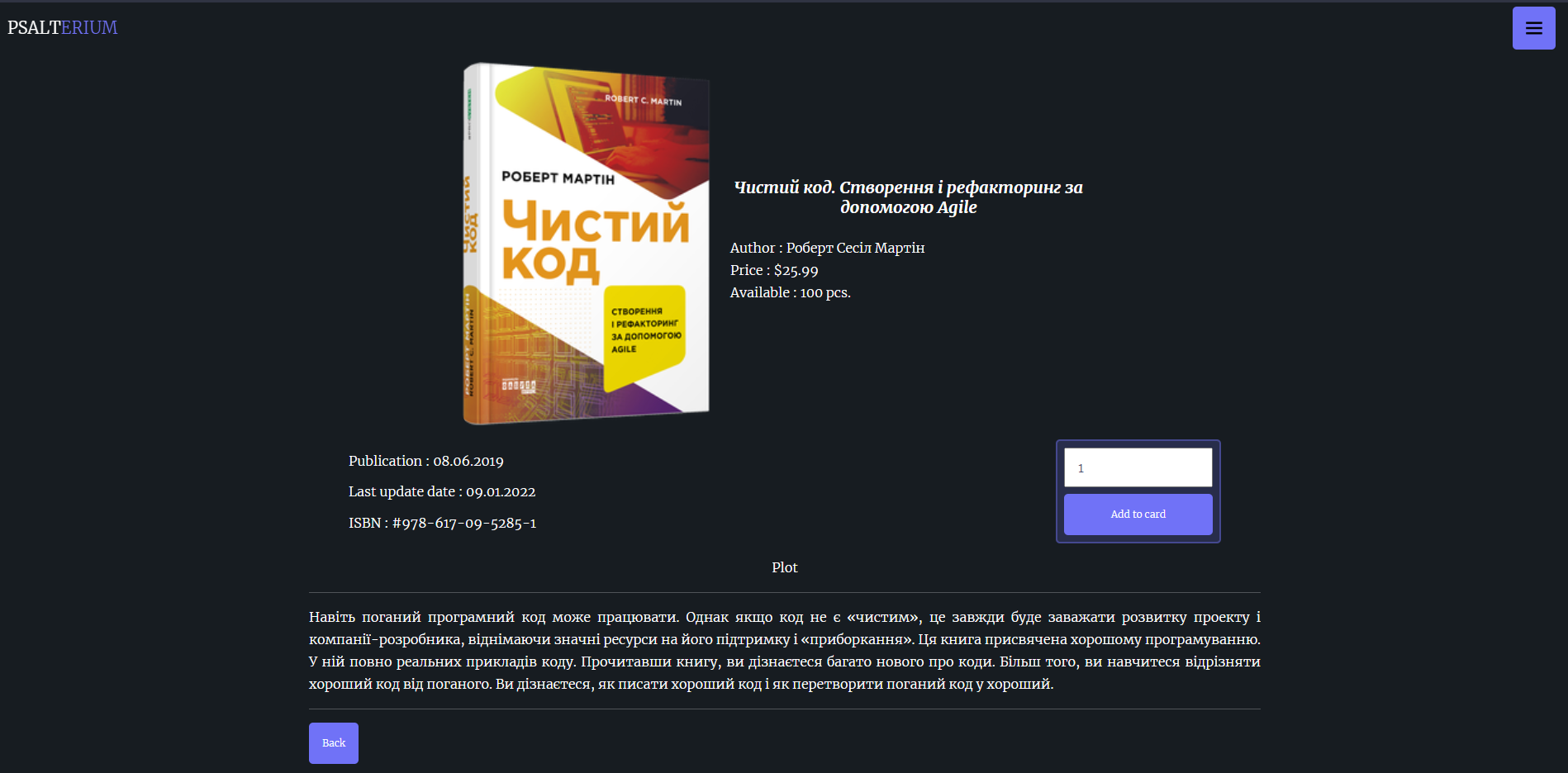


Рис 3.3 Сторінка конкретної книги

Для прикладу оберемо ще одну книгу(“1984”). Додали до корзини 3 екземпляри. Далі користувач переходить до корзини для оформлення замовлення(Рис 3.4). Як можемо бачити, тут відображаються всі нами обрані книги.

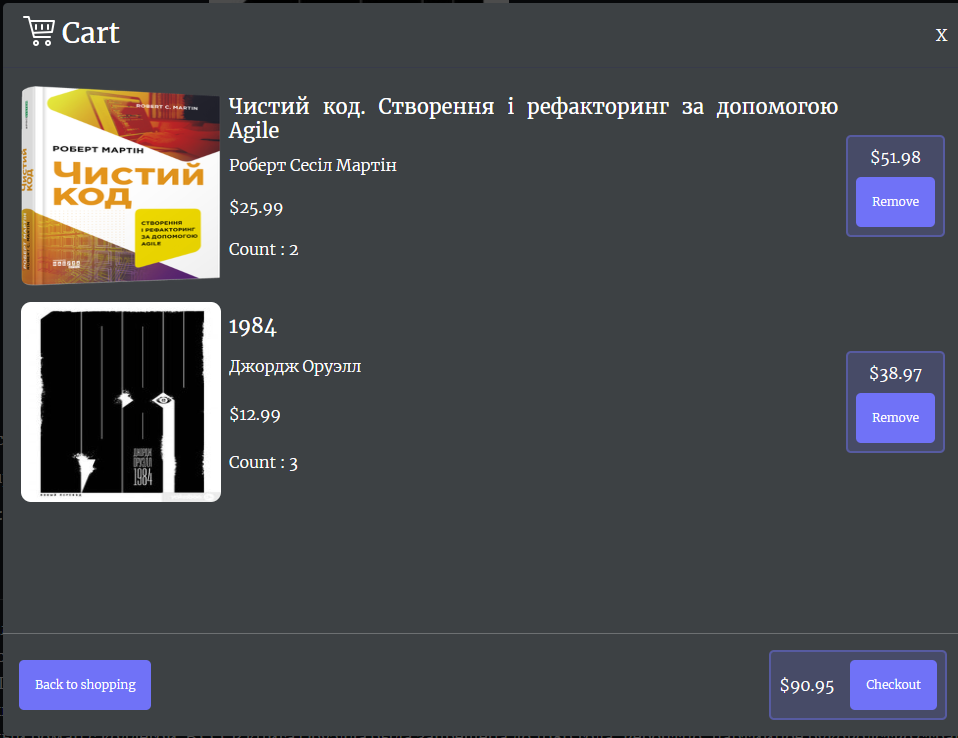


Рис 3.4 Корзина

Також користувач може видалити з корзини книгу, натиснувши на кнопку зліва від книги. Над кнопкою також відображена ціна за всі екземпляри книги. Якщо користувач впевнений, що йому саме ці книги потрібні, він переходить до оформлення замовлення(Рис 3.5). Після оформлення форми замовлення додається в список замовлень(Рис 3.6).

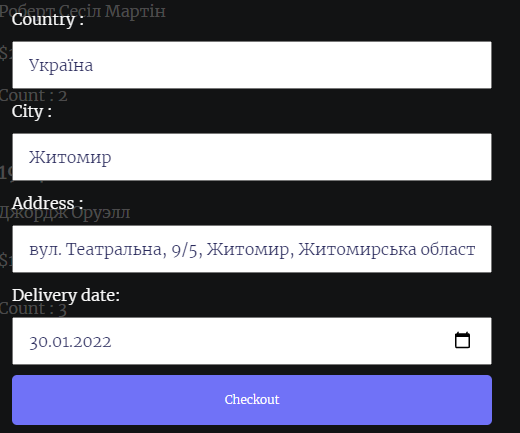


Рис 3.5 Форма оформлення замовлення

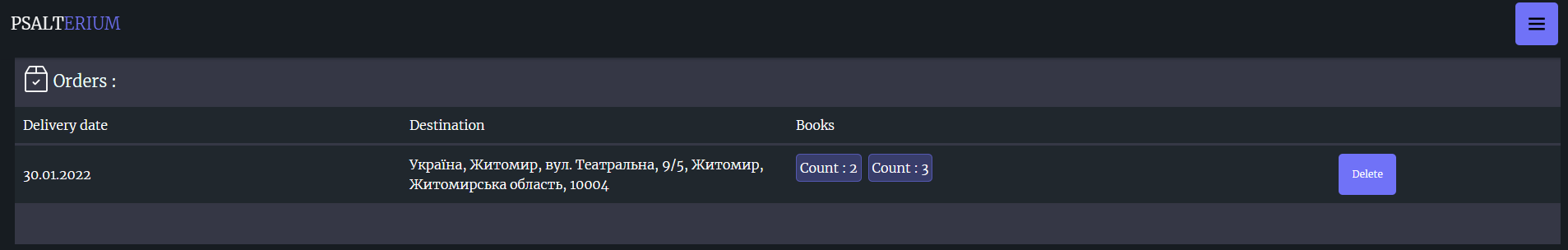


Рис 3.6 Список замовлень

На сторінці список замовлень користувач може відміняти замовлення та переглянути замовлені книги при кліку на підсвічені елементи, які ведуть на сторінки відповідних книжок. На даній сторінці адміну будуть показуватися всі замовлення, а звичайному користувачу тільки його власні.

Далі перейдемо до дадаткової можливості інтернет-магазину – виставлення на продаж власних книг.

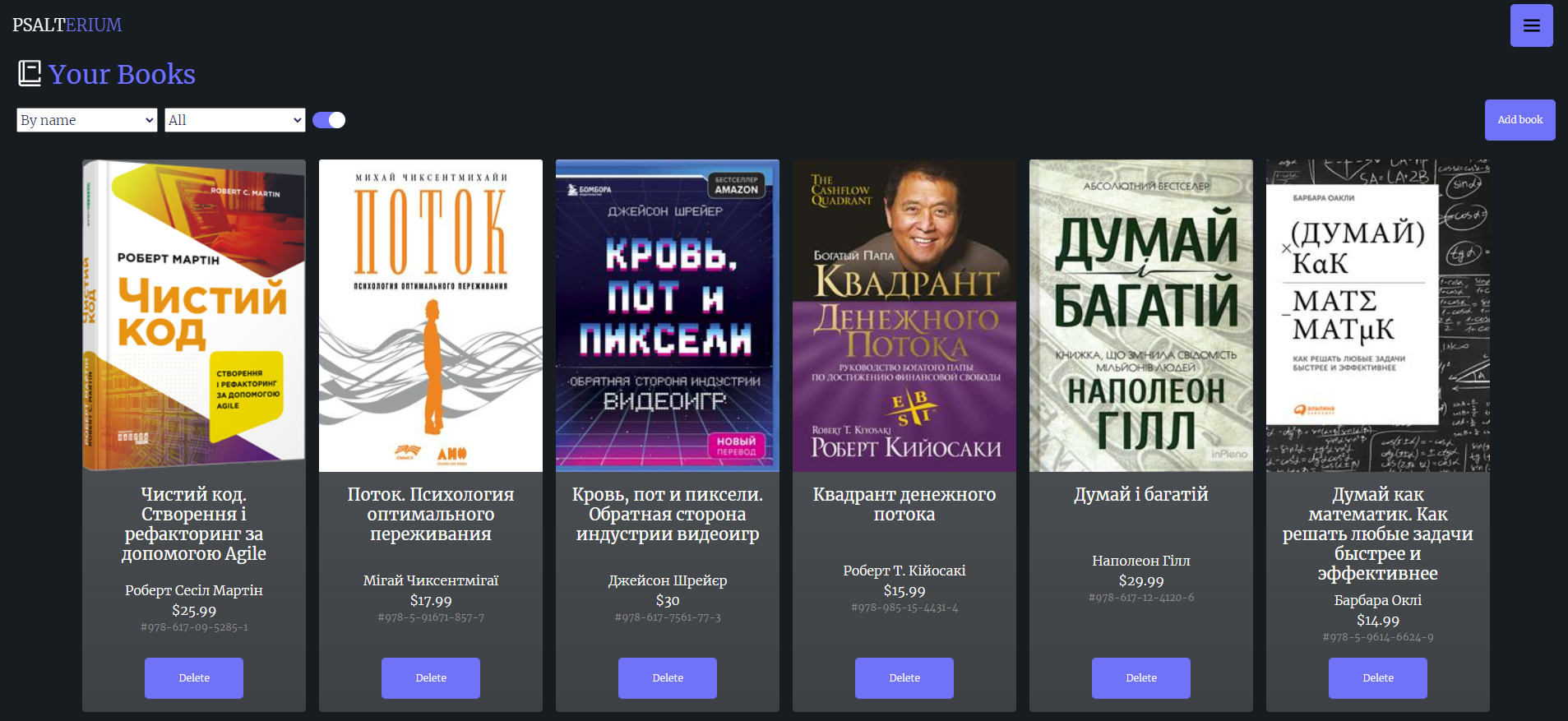


Рис 3.7 Сторінка книг виставлених на продаж користувачем

На даній сторінці у верхньому правому куті розміщена кнопка, яка відкриває форму для книги(Рис. 3.8). Коли користувач заповнить цю форму, книга відобразиться на сторінці та додасться до магазину, де інший користувач зможе її замовити.

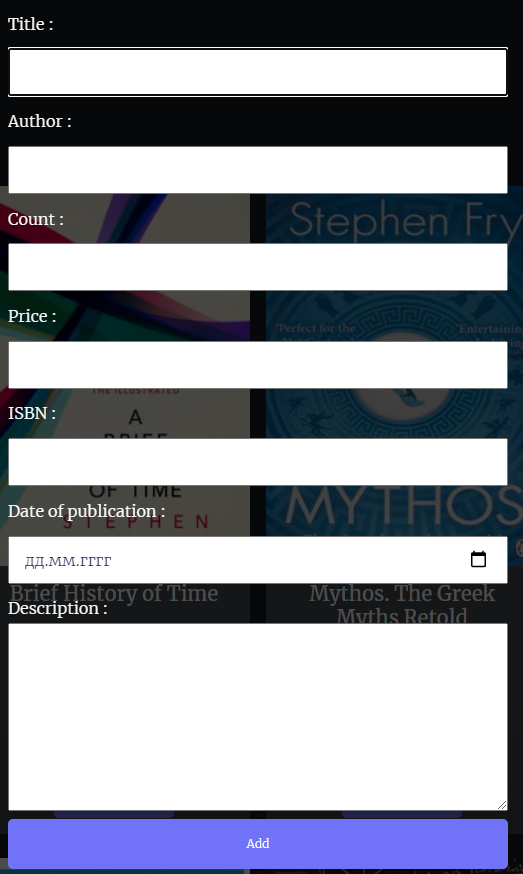


Рис 3.7 Форма для виставлення книг на продаж

Після виставлення книги, її користувач може перейти на неї і побачити додатковий функціонал, а саме можливість зміни ланих про книгу та задати їй жанри.

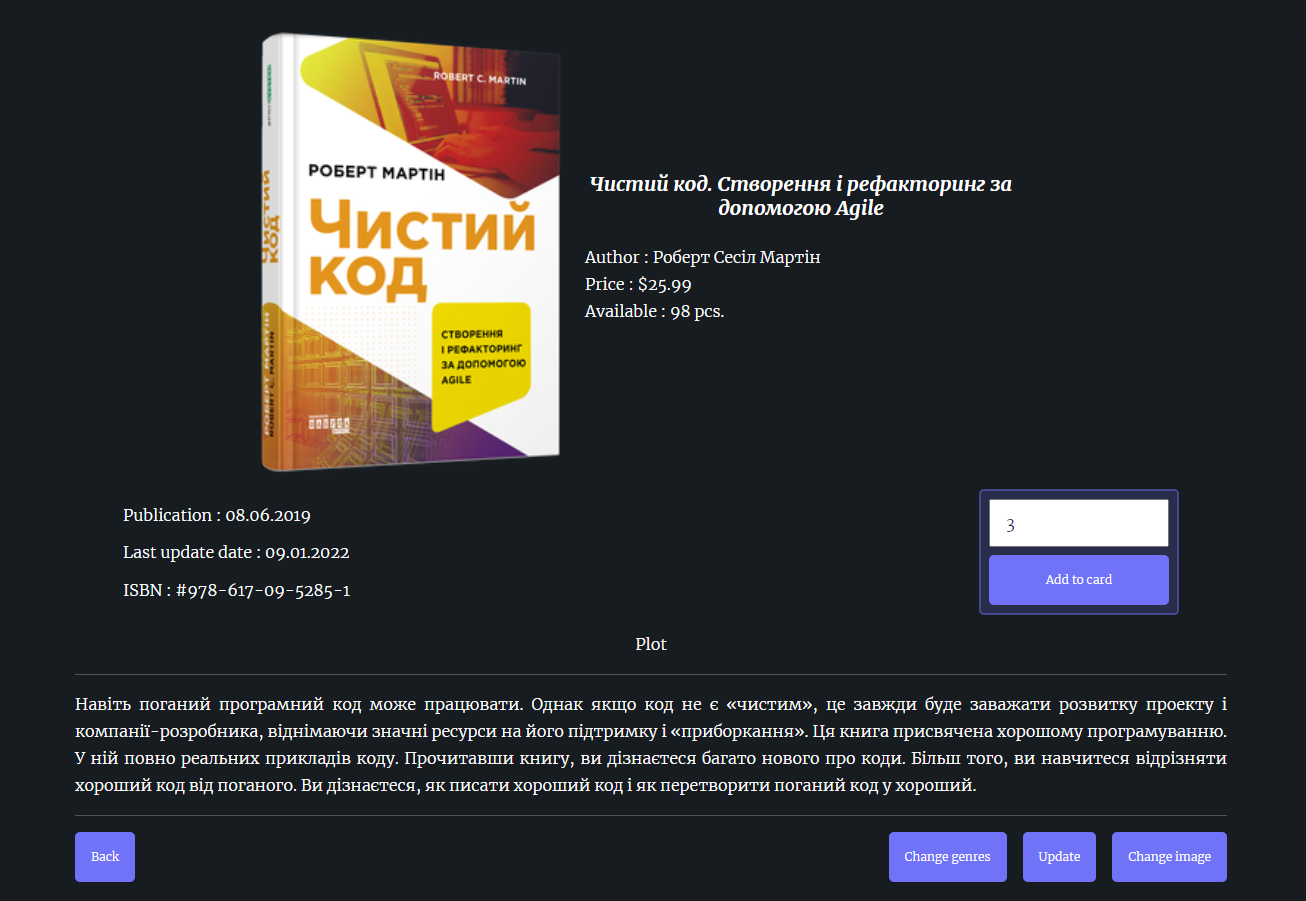


Рис 3.8 Сторінка книги з розширеним функціоналом

Користувач може перейти на власну сторінку(Рисю 3.9) і змінити про себе дані та пароль.

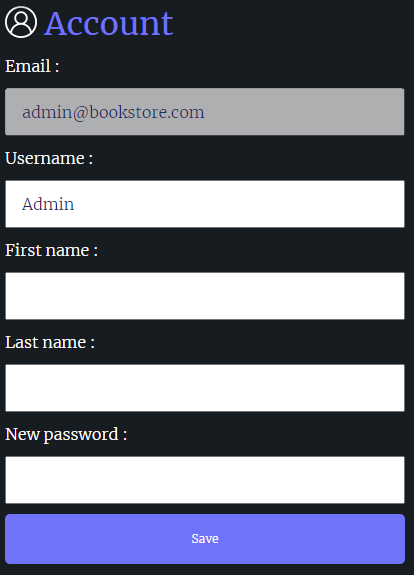


Рис 3.9 Форма для зміни інформації про користувача

Також не можна не розглянути функціонал який доступний тільки для адмінів, а саме сторінку адміністрації(Рис. 3.10).

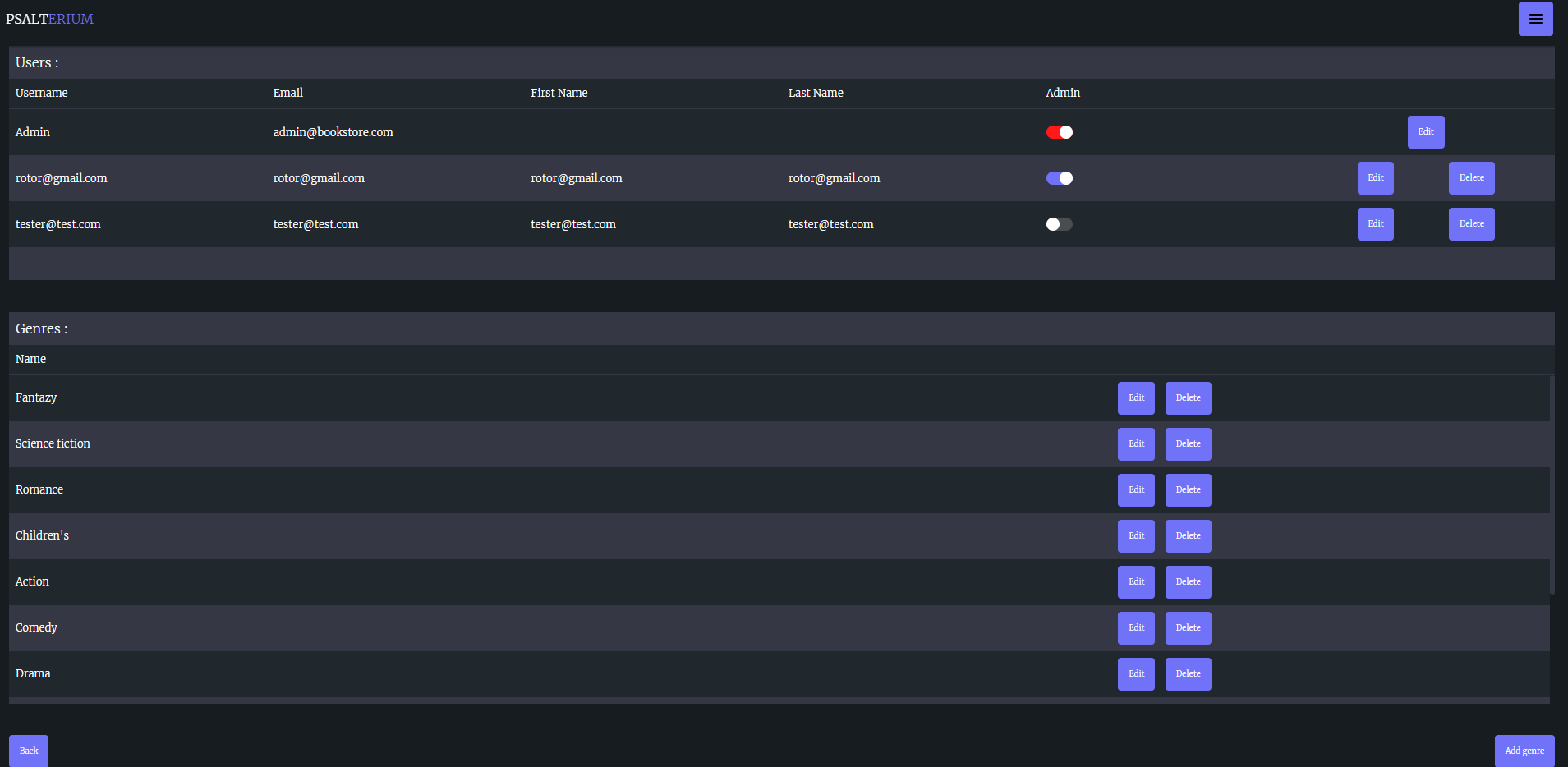


Рис 3.10 Сторінка адміністрації

На сторінку адміністрації мають доступ тільки адміністратори, як не дивно. На даній сторінці адміністратор може переглядати та змінювати дані про користувачів та жанри. Також має можливість видаляти і користувачів і жанри. Користувачі створюються при реєстрації, а для створення жанрів зліва знизу розміщена кнопка для відкриття форми створення жанру(Рис 3.11). Також адмін має можливість надати права адміна іншому користувачу.

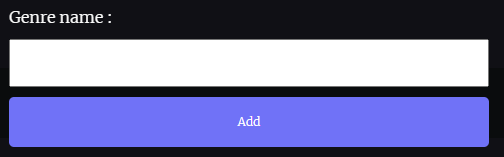


Рис 3.10 Форма створення жанру

Після кожної успішної дії користувач отримує повідомлення про успішність виконання. Розглянемо на прикладі створення нового жанру.

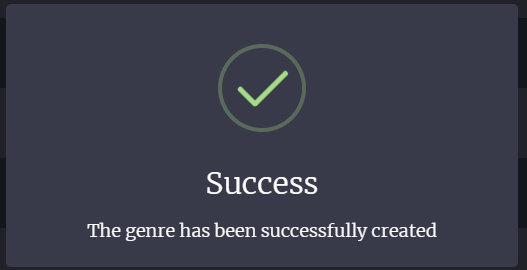


Рис 3.11 Повідомлення про успішне створення жанру

## **Тестування веб-додатку**

Веб-додаток відображає попередження про помилку при при введенні даних для зручного користування користувачем. Наприклад, при спробі створення “пустого” користувача відображаються попередження про обов’язковість полів(Рис 3.12).

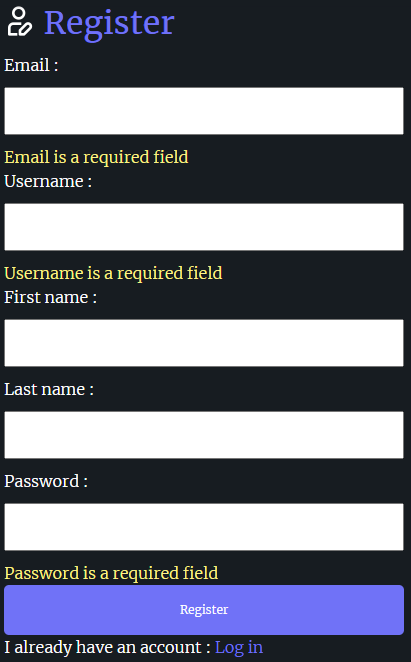


Рис 3.12 Повідомлення про необхідність заповнення

При вводі некоректного значення, з’являється попередження(Рис 3.13)

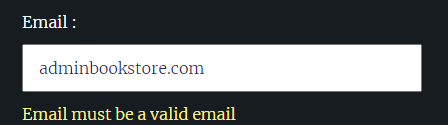


Рис 3.13 Повідомлення про некоректність вводу

При вводі даних які вже існують і не можуть дублюватися, з’являється попередження про існування(Рис 3.14)

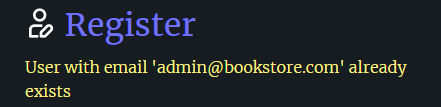


Рис 3.14 Повідомлення про існування користувача

При відправці форми з некоректними даними аутентифікації, з’являється попередження некоректність(Рис 3.15)

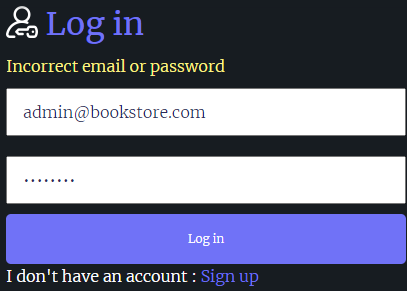


Рис 3.15 Повідомлення некоректність email або пароля

Подібна валідація реалізована на всіх формах

## **Висновки до 3 розділу :**

Було детально описано роботу програмного додатку, користувацький інтефейс, очікувані дії користувача та їх результати. Також було розписано моменти уникнення помилок завдяки допоміжним перевіркам при введенні даних або при роботі з зображеннями. Завдяки цим перевіркам вдалося уникнути частих та серйозних помилок. Наведено приклади обробки помилок.

# **ВИСНОВКИ**

Під час написання даної курсової роботи було отримано навички роботи з розробки веб-додатку.

В першому розділі курсової роботи, було проаналізовано знайдені інтернет-магазини зі схожим функціоналом та інтерфейсом, та було сформовано уявлення про інтернет-магазин, що розроблюється. Було виявлено основні потреби для написання даного веб-додатку.

В другому розділі курсової роботи, було проведено розробку функціональних і загальних частин веб-додатку. При розробці функціонального алгоритму, розробили уявлення про методи розробки програмного коду. На заключному етапі, розроблено його. В результаті, отримано готовий програмний продукт, який виконуває всі заплановані функції і задачі.

В третьому розділі курсової роботи було описана методи користування створеним інтернет-магазином, були пояснені і продемонстровані всі створені елементи інтерфейсу. Також було проведено тестування, тобто інтернет-магазин було запущено в реальних умовах її користування. В ході тестування проблем не виникало.

В результаті виконання курсової роботи отримано інтернет-магазин книжок, який повністю відповідає запланованому функціоналу, умовам та вимогам, поставленим на початку проектування та привищує його. Інтернет-магазин є зрозумілим та зручним в користуванні, адаптивним дял різних пристроїв та швидкодіючим.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Фрэйн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрэйн ; [перевод с английского В. Черник]. - Санкт-Петербург [и др.] : Питер, 2014. - 298 с.
2. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. – СПб.: Питер, 2012. – 304 с
3. Флэнаган Д. JavaScript. Подробное руководство, 6-е издание – Пер. с англ. – СПб: Символ-Плюс, 2012. – 1080 с.
4. А.Зандстра, Ф.Матт. PHP Objects, Patterns, and Practice, 2013. Режим доступу: https://www.apress.com/la/book/9781430260325
5. Дмитрий Кирсанов. “Веб-дизайн” [електронний ресурс] - 2006. Режим доступу: <http://web-diz.com.ua/skachat/veb-dizayn--dmitriy-kirsanov-skachat-besplatno/>
6. Асинхронное программирование . [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/tutorial/13.3.php>
7. Тим Кедлек. “Адаптивный дизайн. Делаем сайты для любых устройств” [електронний ресурс] - 2013. Режим доступу: <https://www.ozon.ru/context/detail/id/21098804/>
8. Плюси і мінуси обєктно-орієнтованого програмування [електронний ресурс] - 2015. Режим доступу: <http://damp.biz/plyusi-i-minusi-obyektno-oriyentovanogoprogramuvannya/>
9. SQL [електронний ресурс] - 2016. Режим доступу: <https://uk.wikibooks.org/wiki/SQL>
10. Создание SQL запросов [електронний ресурс] - 2016. Режим доступу: <https://www.politerm.com/zuludoc/spatial_query_sql.htm#_sql_overview>
11. Документація Fast API[електронний ресурс]. Режим доступу: <https://fastapi.tiangolo.com/>
12. Документація Vue js [електорнний ресурс]. Режим доступу: https://v3.vuejs.org/

# **ДОДАТКИ**

# Додаток А

Схема системи інтернет-магазину книг:

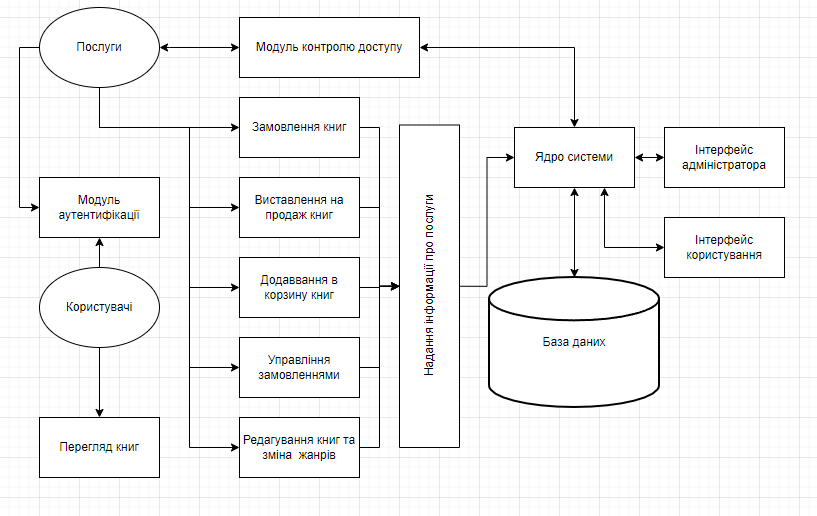
****

Рис А. Структура системи інтернет-магазину книг