



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗАТВЕРДЖУЮ**

Зав. кафедри \_\_\_\_\_ Є. О. Давиденко

« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи бакалавра**

Видано студенту групи 409 факультету комп'ютерних наук

Валюшку Богдану Ігоровичу

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Вебзастосунок інтернет-книгарні

Затверджена наказом по ЧНУ від «17» березня 2023 р. № 60

2. Строк представлення кваліфікаційної роботи « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є вебзастосунок інтернет-книгарні

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до програмного забезпечення;
- визначення архітектури для проектування програмного забезпечення;
- моделювання та проектування програмного забезпечення;
- розробка програмного забезпечення;
- здійснення тестування роботи програмного забезпечення;
- проведення аналізу результатів розробки;

5. Перелік графічних матеріалів

Презентація

---

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій

---

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
А. О. Алексєєва	Кафедра екологія	Спеціальна частина з охорони праці

Керівник роботи PhD, ст. викл. Кафедри ІІЗ Антіпова Катерина Олександрівна

---

(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Завдання прийнято до виконання

Валюшок Богдан Ігорович

---

(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 2023р.

## КАЛЕНДАРНИЙ ПЛАН

### виконання кваліфікаційної роботи

Тема: Вебзастосунок інтернет-книгарні

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	29.11.2022	29.11.2022	виконано
2.	Огляд літератури за темою роботи	29.11.2022	02.12.2022	виконано
3.	Складання календарного плану КРБ	01.04.2023	02.04.2023	виконано
4.	Аналіз предметної області	02.04.2023	03.04.2023	виконано
5.	Розробка проєктних рішень	03.04.2023	05.11.2022	виконано
6.	Моделювання та конструювання ПЗ	05.04.2023	10.04.2023	виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	11.04.2023	25.05.2023	виконано
8.	Розробка спеціальної частини з охорони праці	26.05.2023	27.05.2023	виконано
9.	Оформлення КРБ та презентації	27.05.2023	05.06.2023	виконано
10.	Відгук керівника КРБ	05.06.2023	06.06.2023	виконано
11.	Попередній захист	07.06.2023	07.06.2023	виконано
12.	Завершення оформлення КРБ та презентації	07.06.2023	14.06.2023	виконано
13.	Рецензування	14.06.2023	17.06.2023	виконано
14.	Захист кваліфікаційної роботи	27.06.2023	27.06.2023	виконано

Розробив студент Валюшок Богдан Ігорович

(прізвище, ім'я, по батькові)

(підпис)

«\_\_» \_\_\_\_\_ 2023 р.

Керівник роботи PhD, ст. викладач Антіпова Катерина Олександрівна

(посада, прізвище, ім'я, по батькові)

(підпис)

«\_\_» \_\_\_\_\_ 2023 р.

# АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебзастосунок інтернет-книгарні»

Студента 409 групи: Валюшка Богдана Ігоровича

Керівник: PhD, ст. викладач Антіпова К. О.

Актуальність теми зумовлену тим що виникає все більше необхідності використовувати ефективні технології та інструменти для отримання доступу до книг та знань що містяться в них .

Дана робота присвячена розробці вебсайту для інтернет-книгарні, що має на меті забезпечити користувачів широким доступом до книжкових новинок та класики світової літератури. Сайт буде містити базу даних книжок різних жанрів та напрямків, яку можна буде легко переглянути та знайти потрібну книгу за різними критеріями.

Об'єктом роботи є процес розробки вебзастосунку для інтернет-книгарні, що включає в себе аналіз технічної сторони розробки сайту, його функціональності, зручності інтерфейсу для користувача.

Предметом роботи є технології створення вебзастосунку для підвищення доступності книг та літератури для більш широкої аудиторії, забезпечення якості сервісу та зручності користування.

Метою кваліфікаційної роботи є підвищення обізнаності більш широкої аудиторії про літературну культуру та забезпечення зручного доступу до творів літератури, шляхом створення вебзастосунку інтернет-книгарні, яка пропонує широкий вибір книг та можливість замовити їх онлайн.

Кваліфікаційна робота складається з вступу, 4 розділів, висновків та переліку джерел посилань.

У вступі визначається актуальність теми, мета, предмет та об'єкт дослідження.

У першому розділі проведено аналіз існуючих вебзастосунків-аналогів, визначення функціоналу, переваг, недоліків, технології за допомогою яких було створено вебзастосунок. Формування та опис специфікації вимог вебзастосунку, що розробляється.

У другому розділі наведено моделі та основні етапи моделювання вебзастосунку інтернет-книгарні.

У третьому розділі описується проєктування вебзастосунку, огляд мов, технологій та бібліотек, що використовуються для розробки, та сам процес розробки програмного забезпечення.

У четвертому розділі демонструється результат виконання роботи та тестування.

У висновках проводиться аналіз виконаних робіт та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 63 сторінок, містить 4 розділи, 29 ілюстрацій, 7 таблиць, 20 джерел в переліку посилань.

Ключові слова: *інтернет-книгарня, проєктування архітектури, адаптивний інтерфейс, MySql Database, Laravel.*

## ABSTRACT

to the bachelor's qualification work

«Web application of an online bookstore»

by a student of 409 group: Valiushok Bohdan

Supervisor: PhD, senior lecturer Kateryna Antipova

The relevance of the topic is due to the fact that there is an increasing need to use effective technologies and tools to gain access to books and the knowledge they contain.

This work is devoted to the development of a website for an online bookstore, which aims to provide users with wide access to new books and classics of world literature. The site will contain a database of books of various genres and directions, which can be easily browsed and found by various criteria.

The object of the study is the process of developing a web application for an online bookstore, which includes an analysis of the technical side of the site development, its functionality, and user interface.

The subject of the research is the technology of creating a web application to increase the availability of books and literature for a wider audience, ensuring the quality of service and user friendliness.

The purpose of the thesis is to raise awareness of literary culture among a wider audience and provide convenient access to works of literature by creating a web application for an online bookstore that offers a wide selection of books and the ability to order them online. The qualification work consists of an introduction, n chapters, conclusions and a list of references.

The introduction defines the relevance of the topic, the purpose, subject and object of the study.

The first chapter analyzes the existing analog web applications, identifies the functionality, advantages, disadvantages, and technologies used to create the web application. Formation and description of the requirements specification of the web application under development.

The second section describes the models and main stages of modeling an online bookstore web application.

The third section describes the design of the web application, an overview of the languages, technologies, and libraries used for development, and the software development process itself.

The fourth section demonstrates the results of the work and testing.

The conclusion analyzes the work performed and the results obtained.

The bachelor's qualification work is set out on 63 pages, contains 4 chapters, 29 illustrations, 7 tables, 20 sources in the list of references.

Keywords: *online bookstore, architecture design, adaptive interface, MySQL Database, Laravel.*



## ЗМІСТ

<b>ПЕРЕЛІК СКОРОЧЕНЬ</b> .....	4
<b>ВСТУП</b> .....	5
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ</b> .....	7
1.1 Огляд застосунків-аналогів інтернет-книгарні.....	7
1.2 Аналіз системи, що розробляється.....	10
1.3 Специфікація вимог .....	11
Висновки до розділу 1.....	16
<b>2 МОДЕЛЮВАННЯ ІНТЕРНЕТ-КНИГАРНІ</b> .....	17
2.1 Етапи реалізації проекту .....	17
2.2 Створення сценаріїв.....	18
2.3 Створення use cases діаграм.....	20
2.4 Побудова діаграми станів .....	21
2.5 Побудова діаграми взаємодії .....	23
2.6 Діаграма розгортання .....	27
Висновки до розділу 2.....	28
<b>3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ</b> .....	29
3.1 Розробка UML-діаграм.....	29
3.1.1 Діаграма класів .....	30
3.1.2 Діаграма компонентів .....	31
3.1.3 Діаграма розгортання.....	34
3.1.4 Діаграма пакетів.....	35
3.2 Огляд технологій .....	37
3.2.1 Мови програмування.....	38
3.2.2 Технології frond-end розробки .....	39
3.2.3 Технології back-end розробки.....	42
Висновки до розділу 3.....	44
<b>4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ</b> .....	46
4.1 Дизайн вебзастосунку.....	46

	3
4.2 Реєстрація та авторизація .....	50
4.3 Додавання книг та жанрів .....	53
4.4 Тестування .....	57
Висновки до розділу 4.....	60
<b>ВИСНОВКИ.....</b>	<b>61</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....</b>	<b>62</b>

**ПЕРЕЛІК СКОРОЧЕНЬ**

БД	–	База даних
ПЗ	–	Програмне забезпечення
CRUD	–	Create, Read, Update, Delete
CSS	–	Cascading Style Sheet
JS	–	JavaScript
PHP	–	Hypertext Preprocessor
DOM	–	Document Object Model
MVC	–	Model-View-Controller

## ВСТУП

За останні роки збереження культурної спадщини та доступ до книжок стали одними з найважливіших тем для дискусії в усьому світі. Розвиток інтернет-книгарень та електронних бібліотек є актуальним завданням для багатьох компаній та науково-дослідних установ. Зокрема, програми підтримки книговидавництва та доступу до книжок стають все більш популярними

**Актуальність теми** зумовлену тим що виникає все більше необхідності використовувати ефективні технології та інструменти для отримання доступу до книг та знань що містяться в них.

Однією з важливих переваг інтернет-книгарень є їх доступність. Вони пропонують широкий вибір книжкових ресурсів, включаючи класичну літературу, нові видання, наукові праці та багато іншого. Користувачам немає потреби шукати книги у фізичних магазинах або бібліотеках, адже вони можуть знайти та придбати бажані книги просто за декілька дій.

Завдяки інтернет-книгарням, читачі можуть легко знайти й придбати книги різних жанрів, навчальні посібники, наукові статті та інші видання. Такі книгарні також забезпечують можливість читати огляди, рецензії та рекомендації інших читачів, що допомагає зробити вибір на основі думок інших людей.

Таким чином, інтернет-книгарні є важливим інструментом для збереження культурної спадщини та забезпечення доступу до неї для всіх бажаючих.

**Об'єктом роботи** є процес розробки вебзастосунку для інтернет книгарні, що включає в себе аналіз технічної сторони розробки сайту, його функціональності, зручності інтерфейсу для користувача.

**Предмет роботи** – технології створення вебзастосунку для підвищення доступності книг та літератури для більш широкої аудиторії, забезпечення якості сервісу та зручності користування

**Метою кваліфікаційної роботи** є підвищення обізнаності більш широкої аудиторії про літературну культуру та забезпечення зручного доступу до творів

літератури, шляхом створення вебзастосунку інтернет книгарні, яка пропонує широкий вибір книг та можливість замовити їх онлайн.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- аналіз предметної сфери.
- обговорення деталей проєкту.
- моделювання тоскір.
- розробка backend-частини.
- розробка дизайну за тоскір.
- тестування вебзастосунку.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Огляд застосунків-аналогів інтернет-книгарні

Для розроблення застосунку було розглянуто його аналоги, для того щоб виявити якого функціоналу найбільше потребує користувач.

### **NashaIdea**

Застосунок присвячений проблемам книговидавництва та читання. На сайті можна знайти інформацію про книги та авторів, а також замовити книгу онлайн. Сайт має зручний та зрозумілий інтерфейс. Крім того, на сайті регулярно додаються рецензії та рекомендації від книголюбів та експертів. Таким чином, сайт користувачам знаходить та отримує доступ до якісних книг, що сприяють розвитку читацької культури та підвищенню відомості в галузі літератури.

Таблиця 1.1 – Опис вебзастосунку **NashaIdea**

<b>Назва</b>	<b>NashaIdea</b>
<b>Архітектура</b>	Web application
<b>Виробник</b>	NashaIdea
<b>Мова реалізації</b>	Мова програмування JS та фреймворк JQuery. CSS
<b>Функції</b>	<ul style="list-style-type: none"> <li>– перегляд галереї книг</li> <li>– перегляд опису товару</li> <li>– оформлення замовлення</li> <li>– контактна інформація</li> <li>– актуальна інформація про проєкт</li> </ul>
<b>Переваги</b>	<ul style="list-style-type: none"> <li>– зручність використання</li> <li>– галерея з поступовим завантаженням</li> <li>– плавність анімації</li> </ul>

Кінець таблиці 1.1

<b>Недоліки</b>	– Відсутність реєстрації – Неможливість переглянути статус замовлення на сайті
<b>Посилання</b>	<a href="https://nashaidea.com">https://nashaidea.com</a>

### Книгарня «Є»

Застосунок, про який йдеться, має на меті сприяти розвитку культури читання та забезпечити доступ користувачів до якісних книг та інформації про авторів. На сайті можна знайти великий каталог книг, який охоплює різні жанри та тематики. Інтерфейс сайту простий та зрозумілий, що дозволяє легко зорієнтуватися в каталозі та знайти потрібну книгу. Також на сайті регулярно публікуються рецензії та рекомендації від книголюбів та експертів, що дозволяє користувачам знайти нові цікаві книги та розширити свій кругозір. Загалом, застосунок забезпечує зручний та швидкий доступ до якісної літератури та сприяє розвитку культури читання.

Таблиця 1.2 – Опис вебзастосунку **Книгарня «Є»**

<b>Назва</b>	<b>Книгарня «Є»</b>
<b>Архітектура</b>	Web application
<b>Виробник</b>	ECEM Media GmbH
<b>Мова реалізації</b>	Мова програмування JS та фреймворк JQuery. CSS
<b>Функції</b>	– перегляд галереї книг – перегляд опису товару – оформлення замовлення – контактна інформація – актуальна інформація про проект

Кінець таблиці 1.2

<b>Переваги</b>	<ul style="list-style-type: none"> <li>– зручність використання</li> <li>– доступність для користувача</li> <li>– швидкість роботи</li> <li>– плавність анімації</li> </ul>
<b>Недоліки</b>	<ul style="list-style-type: none"> <li>– помилки пошукової системи</li> <li>– недоступність нових видань</li> </ul>
<b>Посилання</b>	<a href="https://book-ye.com.ua">https://book-ye.com.ua</a>

### Yakaboo

Застосунок, про який йдеться, має на меті сприяти розвитку культури читання та забезпечити доступ користувачів до якісних книг та інформації про авторів. На сайті можна знайти великий каталог книг, який охоплює різні жанри та тематики. Інтерфейс сайту простий та зрозумілий, що дозволяє легко зорієнтуватися в каталозі та знайти потрібну книгу. Також на сайті регулярно публікуються рецензії та рекомендації від книголюбів та експертів, що дозволяє користувачам знайти нові цікаві книги та розширити свій кругозір. Загалом, застосунок забезпечує зручний та швидкий доступ до якісної літератури та сприяє розвитку культури читання.

Таблиця 1.3 – Опис вебзастосунку **Yakaboo**

<b>Назва</b>	<b>Yakaboo</b>
<b>Архітектура</b>	Web application
<b>Виробник</b>	Yakaboo
<b>Мова реалізації</b>	Мова програмування JSt та фреймворк JQuery. Node.js. CSS



Кінець таблиці 1.3

<b>Функції</b>	<ul style="list-style-type: none"> <li>– перегляд галереї книг</li> <li>– перегляд опису товару</li> <li>– оформлення замовлення</li> <li>– контактна інформація</li> <li>– актуальна інформація про проект</li> </ul>
<b>Переваги</b>	<ul style="list-style-type: none"> <li>– зручність використання</li> <li>– книги доступні на декількох мовах</li> <li>– швидкість роботи</li> <li>– зручна та влучна пошукова система</li> </ul>
<b>Недоліки</b>	<ul style="list-style-type: none"> <li>– перевантаженість інтерфейсу</li> <li>– проблема з відображенням статусу замовлення</li> </ul>
<b>Посилання</b>	<a href="https://www.yakaboo.ua">https://www.yakaboo.ua</a>

## 1.2 Аналіз системи, що розробляється

Розробка вебзастосунку для інтернет-книгарні, має на меті забезпечити користувачів широким доступом до книжкових новинок та класики світової літератури. Сайт буде містити базу даних книжок різних жанрів та напрямків, яку можна буде легко переглянути та знайти потрібну книгу за різними критеріями.

Функції:

1. додавання книг
2. редагування короткого змісту
3. видалення книг
4. додавання фотографій у кількості 1 або більше
5. видалення фотографій
6. додавання учасників
7. видалення учасників
8. редагування учасників

9. доступність для англomовних користувачів
10. доступність для україномовних користувачів
11. попередній перегляд фотографій, без переходу на іншу сторінку
12. авторизація для адміністратора
13. адаптивність інтерфейсу для пристроїв з різним розміром екрану

Для даного ПЗ достатньо двох ролей: гість та адміністратор, що наповнює сторінку інформацією

Сценарії:

1. Пошук та вибір книг: Користувач може шукати книги за назвою, автором, жанром або ISBN номером. Після знаходження книги користувач може детально ознайомитися з описом книги, переглянути її обкладинку та ознайомитися зі змістом.
2. Додавання товару до кошика: Після вибору книг користувач може додати їх до кошика і продовжити покупки.
3. Оформлення замовлення: Користувач може переглянути список товарів у кошику, вказати адресу доставки та спосіб оплати.

### **1.3 Специфікація вимог**

#### **ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ**

**Призначення системи (застосунку), для якої розробляється програмне забезпечення**

Призначенням застосунку є забезпечити користувачів широким доступом до книжкових новинок та класики світової літератури.

#### **Погодження, що ухвалені в програмній документації**

Було погоджено, що для створення загального ПЗ та його коректної роботи будуть використовуватися фреймворк – Laravel.

#### **Межі проєкту ПЗ**

Крайня дата завершення роботи над ПЗ – 19.05.2023р.

#### **ЗАГАЛЬНИЙ ОПИС**

#### **Сфера застосування**

Дане ПЗ призначена для продажу книг в Інтернеті та забезпечення користувачам зручного та швидкого способу знаходження та покупки книг.

### **Характеристики користувачів**

Основні характеристики користувачів: наявність ПК та доступу до мережі Інтернет.

### **Загальна структура і склад системи:**

Система складається з таких модулів:

– Модуль реєстрації та авторизації користувачів - цей модуль відповідає за реєстрацію нових користувачів, аутентифікацію та авторизацію вже зареєстрованих користувачів.

– Модуль каталогу книг - цей модуль дозволяє користувачам переглядати список доступних книг та здійснювати пошук книг за різними параметрами.

– Модуль детальної інформації про книги - цей модуль надає користувачам детальну інформацію про книгу, включаючи її опис, автора, ISBN, ціну та наявність. Модуль кошика покупок - цей модуль дозволяє користувачам додавати книги до свого кошика покупок та здійснювати оплату за книги.

– Модуль історії покупок та профілю користувача - цей модуль дозволяє користувачам переглядати свою історію покупок та змінювати свій профіль, включаючи контактну інформацію та адресу доставки.

– Модуль адміністрування - цей модуль дозволяє адміністраторам додавати нові книги до системи, оновлювати інформацію про книги та видаляти книги з системи. Адміністратор також може переглядати замовлення користувачів та виконувати інші адміністративні функції.

Загальна структура системи має такий вигляд:

1. Модуль реєстрації та авторизації користувачів
  - форма реєстрації нового користувача
  - форма входу для зареєстрованих користувачів
  - форма відновлення пароля

- сесії користувачів
- модуль каталогу книг
- список доступних книг
- фільтри та пошук за різними параметрами
- сортування списку книг
- 2. Модуль детальної інформації про книги
  - інформація про книгу
  - автор книги
  - ISBN
  - ціна книги
  - наявність книги
- 3. Модуль кошика покупок
  - Додавання книг до кошика
  - Видалення книг з кошика
  - Зміна кількості книг у кошику
  - Оплата за книги
- 4. Модуль історії покупок та профілю користувача
  - історія покупок
  - зміна профілю користувача
  - контактна інформація
  - адреса доставки

## **ФУНКЦІЇ СИСТЕМИ ІНТЕРНЕТ-КНИГАРНІ**

Функція замовлення книги

### **Опис функції**

Функція замовлення книги допомагає користувачу отримати доступ до цікавлячих його знань.

### **Вхідна і вихідна інформація**

Вхідна інформація – назва книги, короткий опис, ціна;

Вихідна інформація – дата доставки, знання.

**Функціональні вимоги**

Бази даних з інформацією про книги та доступ до мережі Інтернет.

Функція пошуку з фільтрацією

**Опис функції**

Функція пошуку з фільтрацією допомагає користувачу знайти потрібні матеріали за обраними критеріями.

**Вхідна і вихідна інформація**

Вхідна інформація – назва, критерії, ціна;

Вихідна інформація – книги що задовольняють критерії пошуку.

**Функціональні вимоги**

Бази даних з інформацією про книги та доступ до мережі Інтернет.

**Функціональні вимоги:**

- Користувачі повинні мати змогу зареєструватися в системі та увійти до свого облікового запису.
- Користувачі повинні мати змогу переглядати список доступних книг.
- Користувачі повинні мати змогу знайти книгу за назвою, автором, жанром або ISBN.
- Користувачі повинні мати змогу переглянути детальну інформацію про книгу, таку як опис, автор, ISBN, ціна та наявність.
- Користувачі повинні мати змогу додати книгу до свого кошика покупок та здійснити покупку.
- Користувачі повинні мати змогу переглянути свій історію покупок та змінити свій профіль.
- Адміністратори повинні мати змогу додавати нові книги до системи, оновлювати інформацію про книги та видаляти книги з системи.
- Адміністратори повинні мати змогу переглядати історію покупок та профілі користувачів.

**ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ****Джерела і зміст вхідної інформації (даних)**

В даному ПЗ основним джерелом вхідної інформації є користувач. Який зі свого боку має задавати вручну необхідні дані, у нашому випадку це назва, критерії(автор, жанр, ціна).

### **Вимоги до способів організації, збереження та ведення інформації**

Обмін даними відбувається через CRUD-операції Laravel, БД – MySQL.

### **ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ**

Для оптимальної роботи сайту необхідно щоб ПК з якого користувач буде переглядати сайт відповідав мінімальним вимогам до ОС Windows 10.

### **ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

#### **Архітектура програмної системи**

Архітектура складається з клієнтської частини, серверної частини та БД.

#### **Системне програмне забезпечення**

Застосунок має бути реалізовано з допомогою фреймворку Laravel. Процес обміну даними має відбуватися з використанням CRUD-операцій Laravel. В якості бази даних було обрано MySQL.

#### **Програмне забезпечення ведення інформаційної бази**

Через MySQL виконуються CRUD-операції з базою застосунку.

#### **Мова і технологія розробки ПЗ**

Мови розробки – PHP та JavaScript, та фреймворк Laravel.

### **ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ**

#### **Інтерфейс користувача**

Інтерфейс має бути простим та зручним для того щоб користувачу було легко здійснювати навігацію по застосунку.

#### **Апаратний інтерфейс**

Апаратним інтерфейсом є ПК, що буде використовуватися для роботи з вебзастосунком.

#### **Програмний інтерфейс**

Laravel - це високорівневий вебфреймворк з відкритим кодом, написаний на мові програмування PHP. Він дозволяє швидко створювати вебдодатки з

використанням сучасних рішень, таких як архітектура MVC, міграції баз даних, автоматична генерація коду.

### **Переносимість**

Програмне забезпечення може працювати на ОС Windows 10 та вище.

### **Продуктивність**

Час виконання запитів не має перевищувати терміну в 1 секунду.

### **Безпека**

Під час реєстрації та використання програмного забезпечення, користувачі повинні бути задоволені в приватності своїх особистих даних. Програмне забезпечення повинно забезпечити відсутність можливості доступу до персональних даних користувача з боку інших користувачів. Для того щоб користувач міг отримати доступ до своїх даних, йому необхідно авторизуватися в системі.

## **Висновки до розділу 1**

У першому розділі кваліфікаційної роботи бакалавра було проведено аналіз існуючих інтернет-книгарень та розглянуто їх функціонал та інтерфейс. Набуто досвіду пошуку та вирішення проблем програмного забезпечення, а також встановлено вимоги до розроблюваного програмного забезпечення. Також проведено аналіз системи, що розробляється, визначено користувачів, базовий функціонал та сценарії роботи системи.

Розділ містить специфікацію вимог до програмного забезпечення, що розробляється для інтернет-книгарні. У даній частині розділу представлено повний опис функціональності програмного забезпечення. Опис включає в себе множину функціональних та нефункціональних вимог, що описують взаємодію користувачів з застосунком.

## 2 МОДЕЛЮВАННЯ ІНТЕРНЕТ-КНИГАРНІ

### 2.1 Етапи реалізації проєкту

Планування розробки програмного забезпечення є важливим етапом у процесі розробки. Це дозволяє розробникам та командам зрозуміти, що потрібно зробити, як це робити, які ресурси та інструменти потрібні для завершення проєкту, а також уникнути можливих проблем, що можуть виникнути під час розробки. Планування також допомагає забезпечити, що проєкт буде завершено вчасно та в рамках бюджету, а результат буде відповідати вимогам та очікуванням замовника.

Для створення якісного продукту треба створити план дій та список функціоналу який буде реалізований у проєкті.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- аналіз предметної сфери.
- обговорення деталей проєкту.
- моделювання mockup.
- розробка backend-частини.
- розробка дизайну за mockup.
- розробка адмін-панелі.
- тестування вебзастосунку.
- завантаження готового продукту на сервер.

Планування проєкту буде містити в собі такі етапи:

- створення сценарії;
- побудова діаграми взаємодії;
- побудова діаграми станів;
- побудова діаграми компонентів;
- побудова діаграми розгортання;



## 2.2 Створення сценаріїв

Сценарій користування - це послідовність дій, які користувач повинен виконати в системі, щоб виконати певну функцію в залежності від можливостей програмного забезпечення. Сценарії описують те, що система повинна зробити, а не як саме це буде реалізовано [1].

### **Короткий сценарій.**

Адміністратор вебзастосунку заходить на сайт. Виконує вхід у аккаунт. Після чого відкривається адмін панель додає книгу та всю інформацію про неї. Переходить на головну сторінку, для перевірки коректного відображення. Потім знову повертається до адмін панелі та завантажує зображення. Все коректно додано. Переходить на головну сторінку для перевірки коректності. Все коректно відображається.

### **Середній сценарій.**

*Головний сценарій (успішний):*

Адміністратор вебзастосунку заходить на сайт. Виконує вхід у аккаунт. Після чого відкривається адмін. Панель додає книгу та всю інформацію про неї. Переходить на головну сторінку, для перевірки коректного відображення. Потім знову повертається до адмін панелі та завантажує зображення. Все коректно додано. Переходить на головну сторінку для перевірки коректності. Все коректно відображається.

*Альтернативний сценарій:*

1. Не коректний формат зображення, тому книгу не додано.
2. Некоректно збережено дані про книгу, тому їх необхідно редагувати знову.
3. Перевищений розмір для додавання фото.
4. Збій системи на сервері, через що вебзастосунок відмовляється працювати.
5. Адміністратор ввів не правильні дані користувача.

### **Повний сценарій**

Таблиця 2.1 – Повний сценарій

<b>Use Case Name</b>	Додавання нової книги
<b>Scope</b>	Вебзастосунок інтернет-книгарні
<b>Level</b>	Мета адміністратора
<b>Primary Actor</b>	Адміністратор
<b>Stakeholders and interests</b>	<p>Користувач: Зацікавлений у перегляді та подальшому замовленні товару.</p> <p>Адміністратор: Зацікавлений у перегляді коректності відображення інформації.</p> <p>Керівник проєкту: Зацікавлений у прибутку.</p>
<b>Preconditions</b>	Адміністратор ввів коректні данні про книгу
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. після входу адміністратор опинився у адмін. панелі</li> <li>2. перейшов на вкладнику керуванням пропозиціями</li> <li>3. переходить на сторінку додавання книги</li> <li>4. вводить необхідну інформацію, та редагує вміст товару</li> <li>5. зберігає книгу</li> <li>6. переходить на вкладнику керуванням зображенням</li> <li>7. переходить у режим додавання зображень</li> <li>8. додає зображення та вводить його опис</li> <li>9. зберігає введену інформацію</li> <li>10. виходить з адмін панелі</li> </ol>

Кінець таблиці 2.1

<b>Extensions</b>	<ul style="list-style-type: none"> <li>– не коректний формат зображення, тому книгу не додано.</li> <li>– некоректно збережено дані про книгу, тому їх необхідно редагувати знову.</li> <li>– перевищений розмір для додавання фото.</li> <li>– збій системи на сервері, через що вебзастосунок відмовляється працювати.</li> <li>– адміністратор ввів не правильні дані користувача.</li> </ul>
<b>Special Requirements</b>	– система має виконувати запити не довше 1 секунди
<b>Technology and Data Variations List:</b>	<ul style="list-style-type: none"> <li>– для успішного виконання усіх кроків, необхідно мати стабільне інтернет-з'єднання</li> <li>– редактор книг, реалізовано пропозицію саме так як виглядає у редакторі.</li> </ul>
<b>Frequency of Occurrence</b>	– система працює майже безперервно, але це залежить від електропостачання серверу та інтернету
<b>Miscellaneous</b>	<ul style="list-style-type: none"> <li>– чи виникають помилки при виконанні завантаження зображення</li> <li>– редагування зображення при додавання книги</li> </ul>

### 2.3 Створення use cases діаграм

Use Case - це опис взаємодії між користувачем та системою, який описує, як користувач взаємодіє з системою, щоб виконати певну задачу або досягти мети. Це дозволяє проаналізувати поведінку користувача та системи під час взаємодії, а також допомагає зрозуміти, як система повинна поводитися у різних ситуаціях [2].

На діаграмі (рис. 2.1) зображено 2 актори (відвідувач та адміністратор). Відвідувач може: переглядати книги, читати короткий опис, оформляти замовлення. Так як перегляд і оформлення замовлення є частиною пошуку книги то вони з'єднані стрілкою включення. Тому відвідувач з'єднані асоціацією. Адміністратор також може виконувати всі дії, що притаманні відвідувачу, тому відвідувач та адміністратор поєднані стрілкою узагальнення [3].

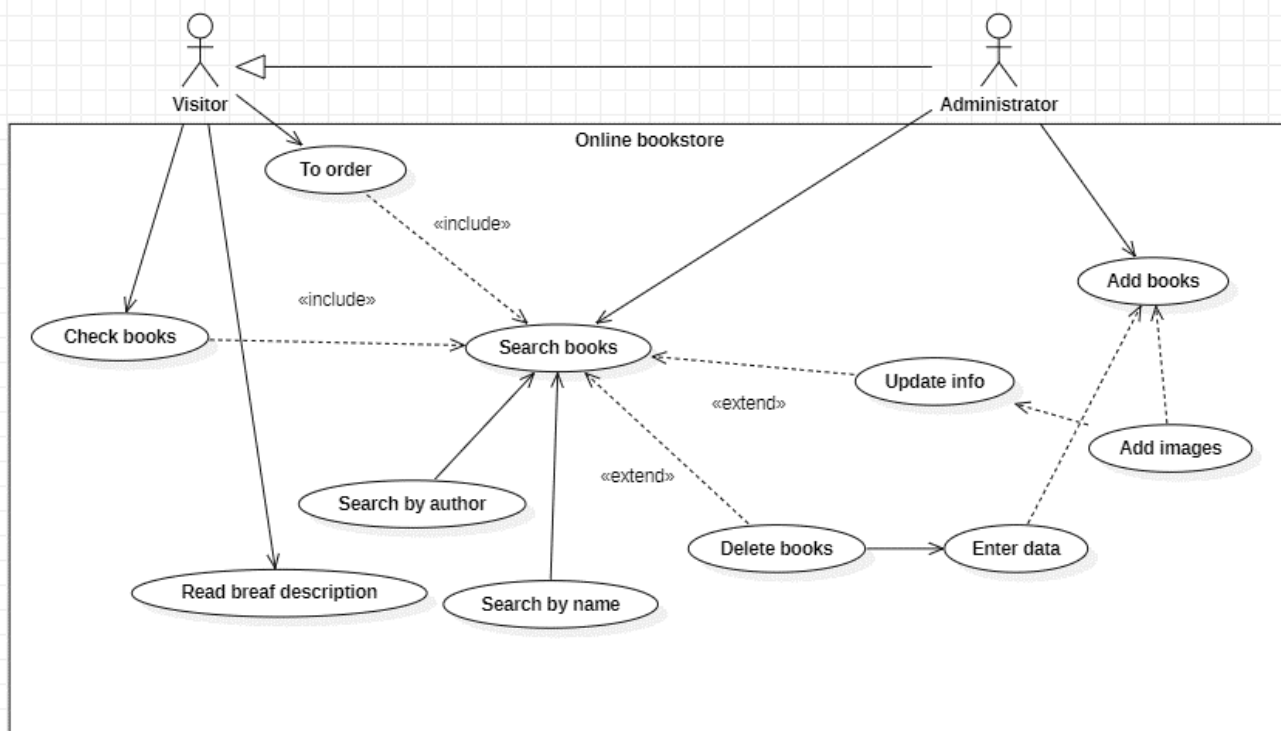


Рисунок 2.1 – Діаграма використання

Також адміністратор за допомогою панелі адміністрування може маніпулювати даними, тобто видаляти, додавати, редагувати. Так як щоб видалити та редагувати дані, потрібно їх знайти, тому «кейси» видалення та редагування поєднанні з пошуком за допомогою стрілки розширення.

## 2.4 Побудова діаграми станів

Було створено діаграму станів сторінки редагування книги (рис. 2.2). Об'єктом є книга. Точкою входу є те що книги не існує і її потрібно створити, потім вона створюється та переходить у стан «Книга була створена», після цього книга може бути у станах «Книга була видалена», «Книга редагується», «Книга в стані валідації», після ж валідації книга у стані збереження у БД, а з БД може бути видалення або у процесі оновлення. Точкою виходу є стан «Книга видалена» та стан «Збереження книги» [4].

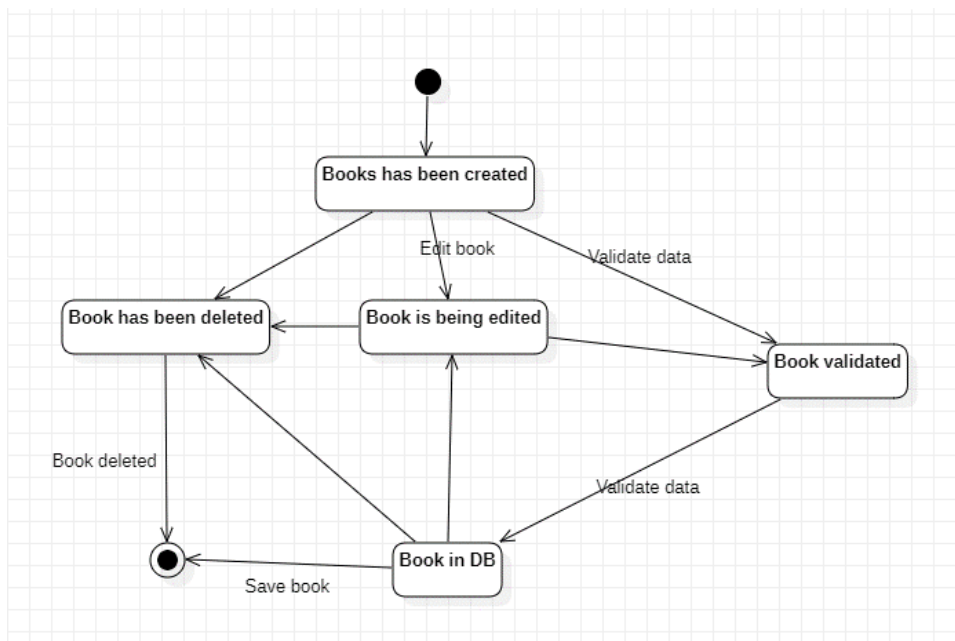


Рисунок 2.2 – Діаграма станів сторінки редагування книги

Потім було створено діаграму станів оформлення замовлення(рисунок 2.3). Об'єктом є корзина, точкою входу є те що корзина не існує. Потім йде створення корзини та додавання в неї книги, після цього можна перейти в стан продовжити покупки та перевіряти книги в корзині. Після цього можна видалити окрему книгу з корзини, видалити всі книги або перейти до оплати.

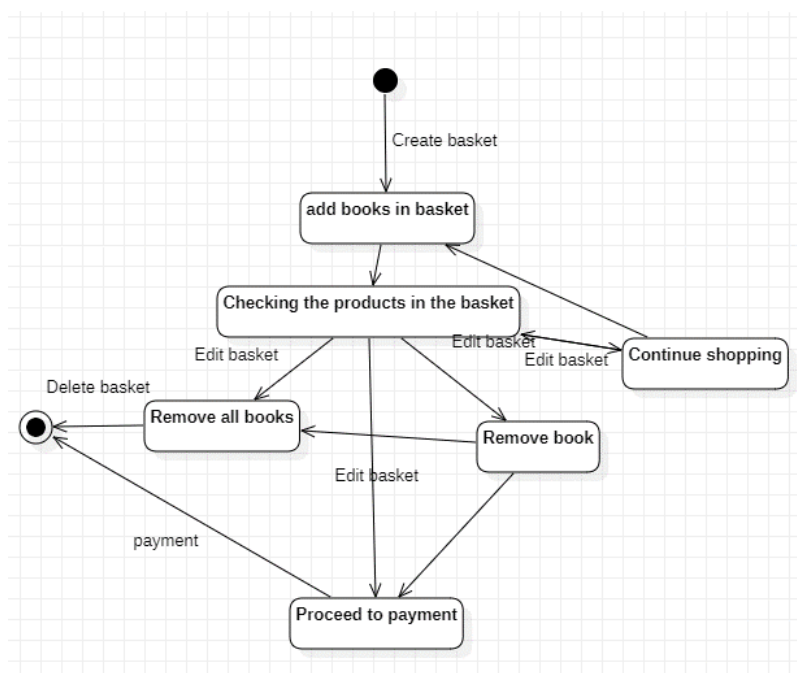


Рисунок 2.3 –Діаграма станів оформлення замовлення

## 2.5 Побудова діаграми взаємодії

Діаграма взаємодії є важливим інструментом в аналізі та проектуванні програмного забезпечення. Вона допомагає описати взаємодію між об'єктами та компонентами системи, яка є ключовою для її правильного функціонування. Діаграма взаємодії дозволяє відображати поведінку системи на рівні об'єктів та компонентів, що дозволяє легко розуміти, як вони взаємодіють між собою та які операції вони виконують [5].

Була побудована діаграма послідовності при переході на сторінку з книгами (рис. 2.4). Виконується запит на генерацію сторінки (1), запит до БД через модель (2), що повертає масив з доступними книгами (3), масив передається у View (4), що у свою чергу генерує сторінку (5), сторінка повертається (6) та відображається користувачу (7).

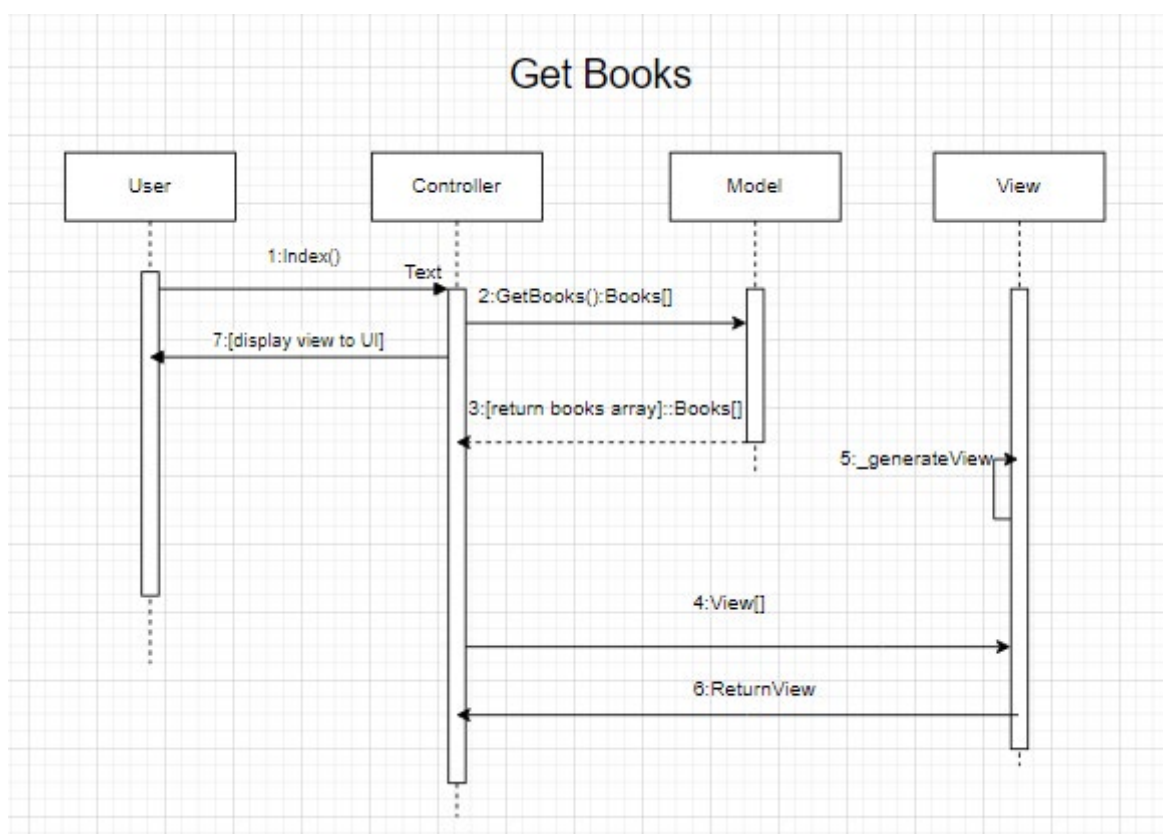


Рисунок 2.4 – Перегляд списку книг

Була побудована діаграма послідовності при перегляді конкретної книги (рис. 2.5). Виконується запит на генерацію сторінки (1), запит по ідентифікатору до БД (2), що повертає об'єкт (3), об'єкт передається у View (4), що у свою чергу генерує сторінку (5), сторінка повертається до контролеру (6) та відображається користувачу (7).

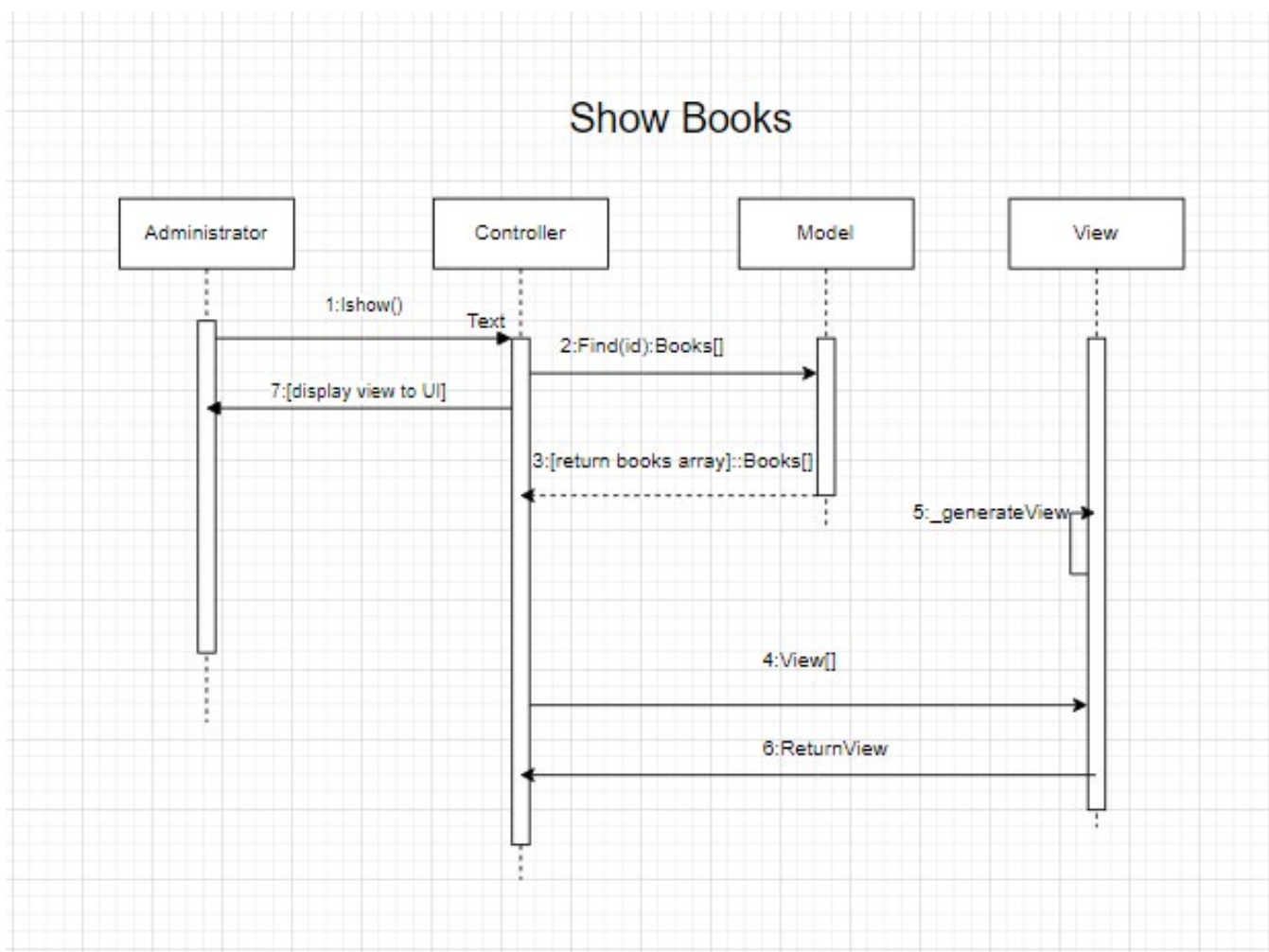


Рисунок 2.5 –Перегляд інформації про книгу

Далі було створено діаграму оновлення книги (рис. 2.6), виконується запит даних (1), збереження даних після пошуку за індексом (2), повертається оновлений масив(3), запит на оновлення сторінки (4), генерація сторінки (5), повернення сторінки (6), відображення користувачу (7).

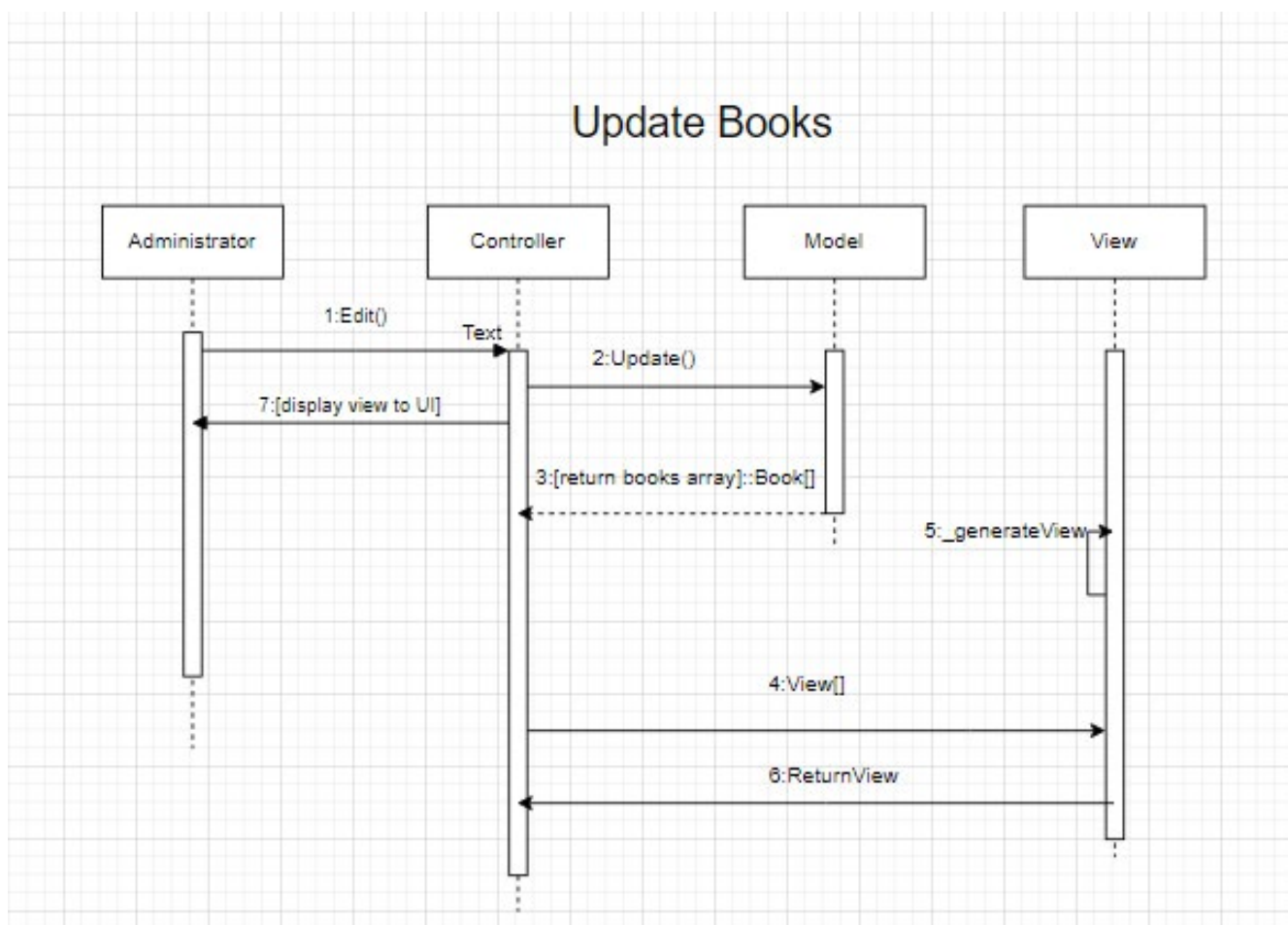


Рисунок 2.6 – Редагування книги



Далі було створено діаграму створення книги (рис. 2.7), відправлення даних(1), валідація даних(2), збереження даних (3), повертається оновлений масив(4), запит на оновлення сторінки (5), генерація сторінки (6), повернення сторінки (7), відображення користувачу (8).

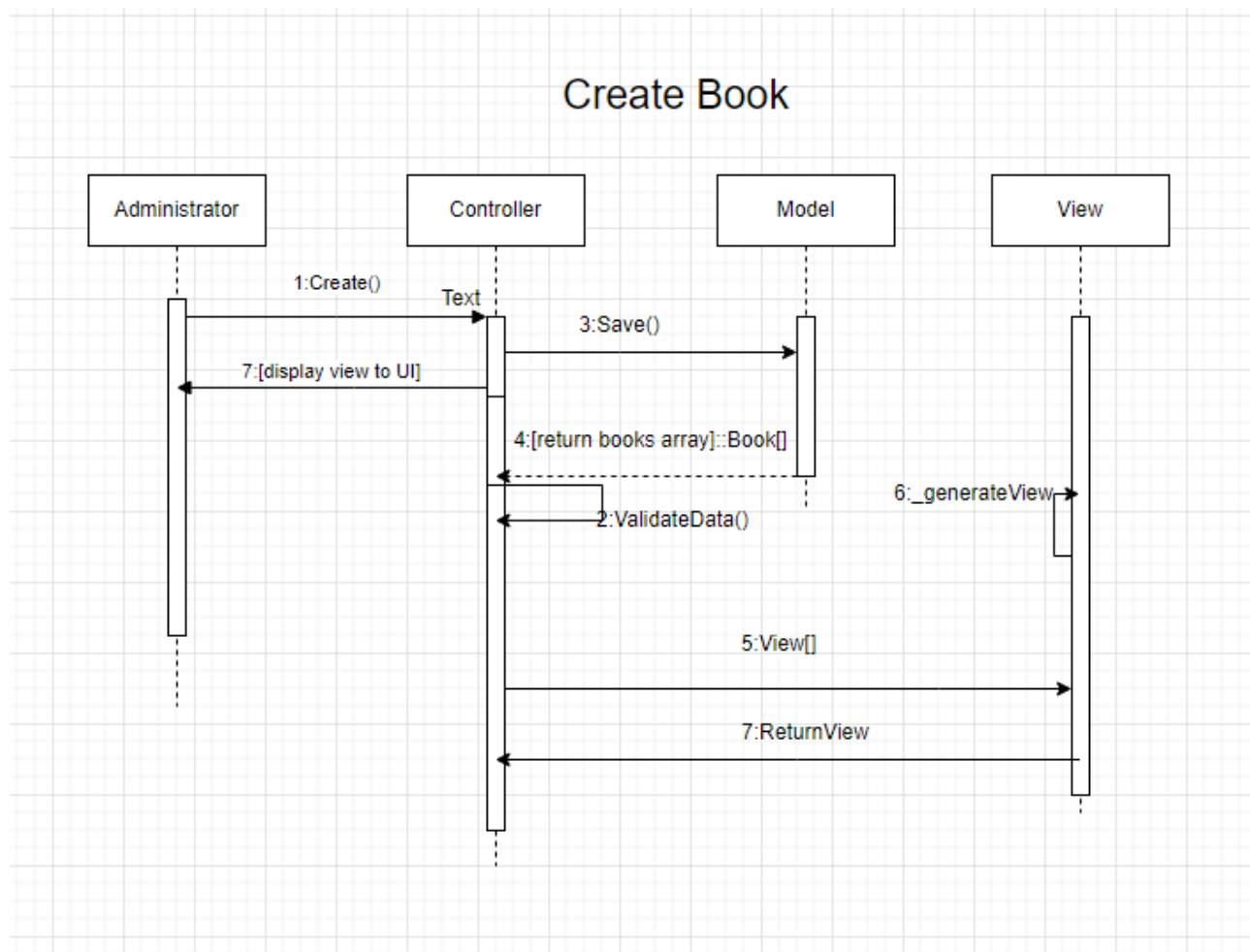


Рисунок 2.7 –Створення книги

Діаграма входу в систему (рис. 2.8) на якій користувач виконує вхід з введеними даними (1), виконується валідація даних (2), перевірка даних з даними на сервері (3), потім повертається об'єкт (4), виконується запит на генерацію сторінки в залежності від повернутих даних (5), генерується сторінка (6), повертається (7) для перегляду результату користувачем (8).

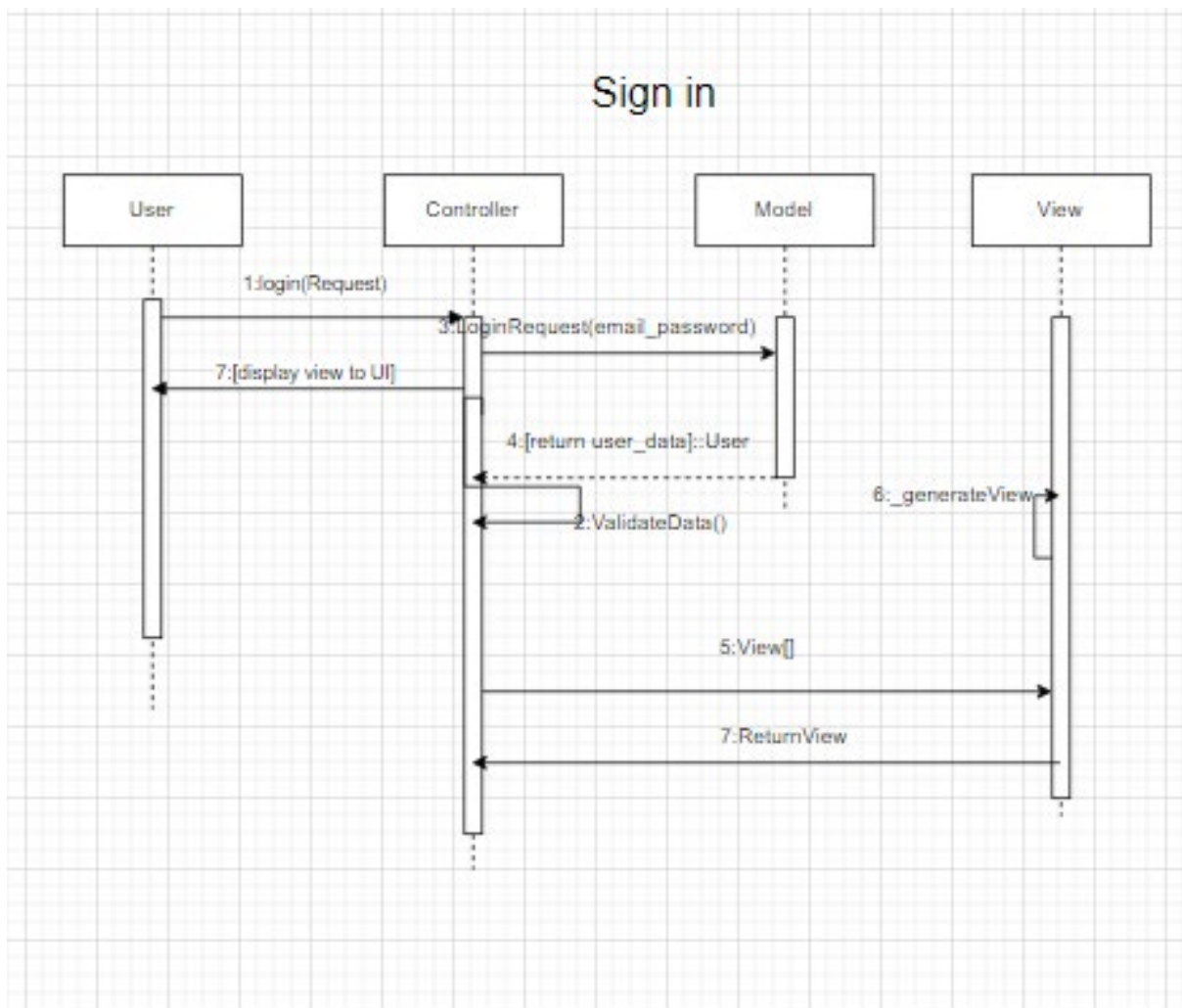


Рисунок 2.8 –Вхід в акаунт

## 2.6 Діаграма розгортання

Діаграма розгортання використовується для відображення фізичного розміщення компонентів системи, та їх залежності.

В діаграмі(рис 2.9) присутні компоненти системи, це пристрій користувача з якого він переглядає сторінку, сервер застосунку та сервер з базою даних.

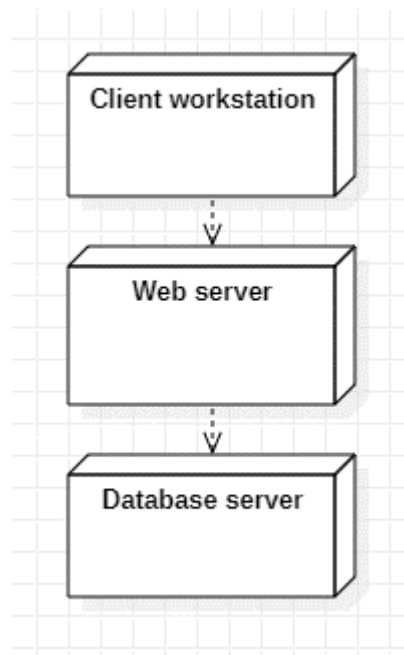


Рисунок 2.9 – Діаграма розгортання

Діаграма розгортання є важливим інструментом в архітектурному моделюванні програмного забезпечення. Вона надає візуальне представлення фізичної архітектури системи, описуючи розташування компонентів системи та способи їх взаємодії. Основною перевагою діаграми розгортання є зручність в сприйнятті та розумінні архітектурних рішень, які стосуються фізичної інфраструктури системи [6].

### **Висновки до розділу 2**

У другому розділі було описано основні етапи розробки інтернет-книгарні та вирішення поставлених завдань. При цьому, були застосовані навички у побудові та використанні діаграм взаємодії, діяльності та розгортання. З метою детального опису взаємодії користувачів з програмним забезпеченням, було створено сценарії використання на основі діаграми використання, а після розробки короткого та поверхневого сценаріїв, було складено повний usecase, що включає всі деталі використання програмного забезпечення.

### 3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ

#### 3.1 Розробка UML-діаграм

Для проєктування програмного забезпечення) використовуються різноманітні діаграми, серед яких особливе місце займають UML-діаграми. UML є стандартною мовою моделювання, яка надає зручний набір діаграм для візуалізації та опису різних аспектів системи [7].

Один з головних приводів використання UML-діаграм в процесі проєктування ПЗ полягає в їх здатності описати бізнес-процеси. Це допомагає розробникам зрозуміти потреби та вимоги користувачів та відобразити їх у структурі ПЗ.

Крім того, UML-діаграми забезпечують можливість візуалізувати системне проєктування. Наприклад, діаграми класів відображають структуру системи, показуючи класи, їх атрибути та методи. Це спрощує розуміння взаємозв'язків між компонентами системи та організацію її структури.

Також, UML-діаграми дозволяють відобразити різні організаційні структури.

Ось список різновидів UML-діаграм та їх короткий опис:

- Діаграма класів (Class Diagram): Відображає структуру системи, показуючи класи, їх атрибути, методи та взаємозв'язки між ними.
- Діаграма послідовності (Sequence Diagram): Показує взаємодію об'єктів у часовій послідовності. Демонструє порядок виконання дій між об'єктами та відображає взаємодію в рамках певного сценарію.
- Діаграма активностей (Activity Diagram): Використовується для моделювання бізнес-процесів, алгоритмів та послідовностей дій. Відображає стани, дії та розгалуження у системі, допомагаючи відстежувати логіку виконання.
- Діаграма станів (State Diagram): Відображає різні стани, у яких може перебувати об'єкт або система, а також переходи між цими станами.
- Діаграма компонентів (Component Diagram): Показує фізичну або логічну структуру системи, включаючи компоненти, інтерфейси, залежності та зв'язки між ними. Використовується для моделювання архітектури системи.

– Діаграма пакетів (Package Diagram): Відображає організацію елементів моделі в пакети. Пакети групують класи та інші елементи моделі, спрощуючи структуру та організацію моделі.

– Діаграма композиції (Composite Structure Diagram): Показує структуру об'єктів, які складаються з інших об'єктів та компонентів.

Ці діаграми UML дозволяють розробникам візуалізувати різні аспекти системи, моделювати їх структуру та поведінку, а також сприяють зрозумінню та спілкуванню між різними учасниками проєкту. Кожен тип діаграми має свою специфіку та використовується для конкретних цілей у розробці ПЗ.

### **3.1.1 Діаграма класів**

Діаграма класів(рис. 3.1) є незамінним інструментом при моделюванні та проєктуванні об'єктно-орієнтованих систем. Вона дозволяє графічно представити структуру системи, включаючи класи, атрибути та взаємозв'язки між ними. Це дає можливість розробникам краще розуміти організацію системи та взаємодії між її компонентами.

Діаграма класів забезпечує зручний спосіб моделювання структури системи. Вона допомагає ідентифікувати класи, їх атрибути та методи, а також встановлює залежності та асоціації між класами. Це дає можливість аналізувати взаємозв'язки та взаємодії між класами, що є важливим для розуміння поведінки системи та виявлення потенційних проблем [8].

Діаграма класів також використовується для проєктування системи. Вона допомагає організувати структуру системи, включаючи класи та їх взаємозв'язки. Це дозволяє розробникам відобразити, які компоненти системи взаємодіють між собою та працюють задля правильного функціонування системи. Діаграма класів сприяє правильному проєктуванню системи та забезпечує її ефективну архітектуру.

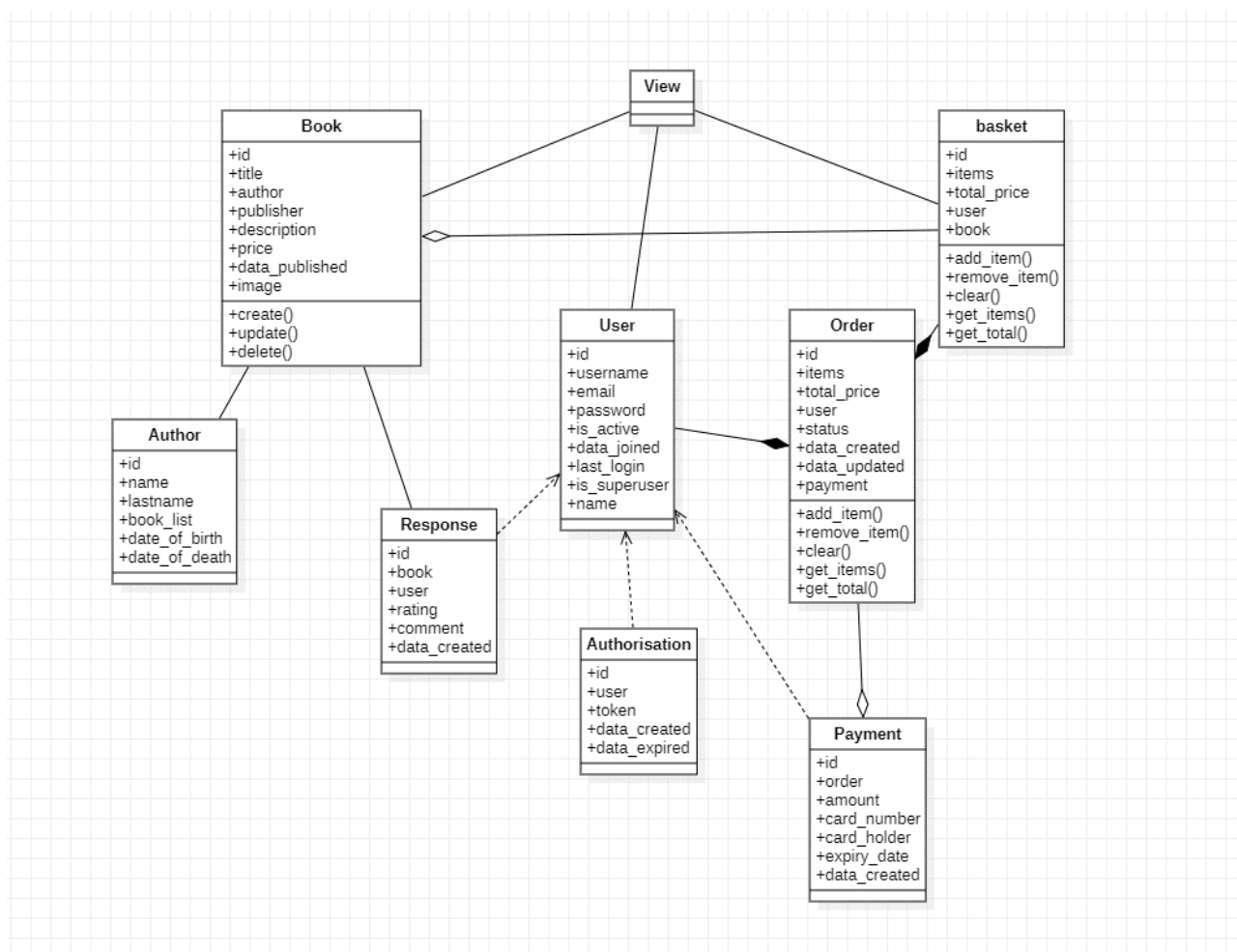


Рисунок 3.1 – Діаграма класів інтернет-книгарні

Крім того, діаграма класів сприяє комунікації та спілкуванню між учасниками проекту. Вона створює спільне розуміння структури системи та дозволяє легко обговорювати та узгоджувати рішення щодо проектування та розробки. Використання діаграм класів спрощує комунікацію між розробниками, аналітиками та іншими учасниками проекту, що сприяє успішному виконанню проекту та досягненню його цілей.

### 3.1.2 Діаграма компонентів

Діаграма компонентів є корисним інструментом для моделювання архітектури системи. Вона дозволяє графічно представити компоненти системи та їх взаємозв'язки, такі як модулі, бібліотеки, пакети та фреймворки. Це допомагає розробникам краще розуміти структуру системи та взаємодії між її компонентами.

Одна з основних переваг використання діаграми компонентів полягає в тому, що вона дозволяє виявити залежності між компонентами. Це допомагає розробникам усвідомити, які компоненти взаємодіють між собою та які залежності можуть виникати між ними. Також діаграма компонентів дозволяє відображати залежності від зовнішніх систем або сторонніх модулів, що допомагає усвідомити зовнішні фактори, які можуть впливати на систему[9].

Іншою важливою функцією діаграми компонентів є комунікація та спілкування між учасниками проєкту. Вона дозволяє створити спільне розуміння архітектури системи та легко обговорювати рішення щодо розробки та інтеграції компонентів. Така візуалізація сприяє ефективній комунікації між розробниками, архітекторами та іншими учасниками проєкту, що покращує розуміння та співпрацю в процесі розробки системи.

Отже, діаграма компонентів є важливим інструментом для моделювання архітектури системи, виявлення залежностей та взаємодій між компонентами та покращення комунікації між учасниками проєкту. Вона допомагає розробникам краще розуміти систему та забезпечує ефективне проєктування та розробку програмного забезпечення.

Структура діаграми компонентів(рис. 3.2), що складається з таких пунктів: `WebApplicationServer` - компонент, який відповідає за роботу сервера вебдодатка, тобто обробку запитів та надсилання відповідей на запити від користувачів.

`UserInterface` - компонент, який відповідає за роботу інтерфейсу користувача. Цей компонент може містити різні елементи інтерфейсу, такі як кнопки, текстові поля, таблиці, зображення тощо.

`Application` - компонент, який відповідає за логіку додатку. Він містить методи та функції, які виконують різні завдання, такі як обробка запитів користувачів, робота з базою даних тощо.

`Payment` - компонент, який відповідає за обробку оплати. Цей компонент може включати різні методи та функції, які дозволяють проводити оплату, перевіряти статус оплати, отримувати повідомлення про успішну або неуспішну оплату тощо.

API - компонент, який надає доступ до функціоналу додатку через відкритий інтерфейс програмування застосунків (API). Цей компонент може містити різні методи та функції, які дозволяють взаємодіяти з додатком за допомогою зовнішніх систем або послуг.

Database - компонент, який відповідає за зберігання даних. Цей компонент може включати різні елементи, такі як бази даних, таблиці, записи тощо. Даний компонент може забезпечувати доступ до даних з інших компонентів додатку, наприклад, з компоненту Application.

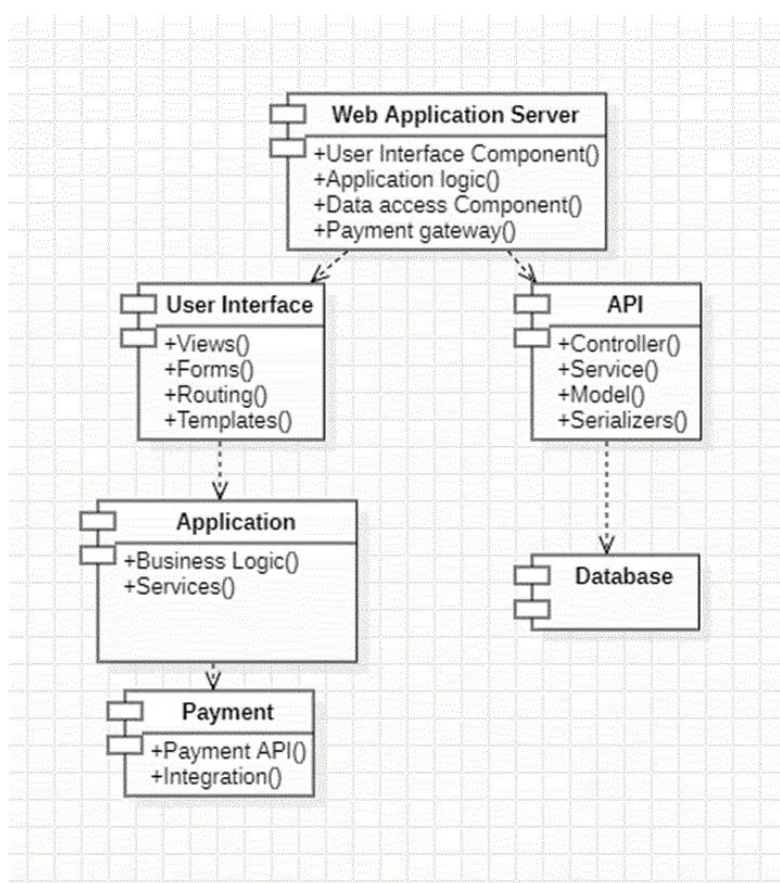


Рисунок 3.2 – Діаграма компонентів

Побудова діаграми компонентів інтернет-книгарні була проведена з високою якістю та відіграла важливу роль у процесі розробки. Ця діаграма допомогла уявити та описати всі основні компоненти системи, їх взаємозв'язки та функціональність.

Шляхом ретельного аналізу та деталізації компонентів, було визначено функціональні вимоги кожного компонента, а також їхню взаємодію з іншими



частинами системи. Це стало основою для ефективної розробки та імплементації інтернет-книгарні.

### 3.1.3 Діаграма розгортання

Діаграма розгортання є важливим інструментом в області проектування програмного забезпечення. Вона використовується для візуалізації фізичного розташування компонентів системи та їх взаємозв'язків з апаратними пристроями або середовищами, в яких система функціонує.

Основна мета діаграми розгортання - показати, як компоненти програмного забезпечення розташовані на апаратному рівні, як вони взаємодіють між собою та з зовнішніми середовищами. Вона відображає фізичну структуру системи та розподілення компонентів на різних серверах або вузлах, а також з'єднання між ними. Це дозволяє архітекторам та розробникам краще розуміти, як система буде функціонувати в реальному середовищі[9].

Діаграма розгортання має декілька важливих переваг. Вона допомагає виявити можливі проблеми та конфлікти при розгортанні системи, такі як недостатня потужність серверів або неправильне розташування компонентів. Вона також сприяє плануванню масштабування системи, дозволяючи зрозуміти, як можна розподілити компоненти на різні сервери або кластери.

Крім того, діаграма розгортання допомагає комунікації між учасниками проекту. Вона створює спільне розуміння структури системи та дозволяє ефективно обговорювати рішення щодо розгортання та інфраструктури. Використання діаграми розгортання спрощує спілкування між розробниками, архітекторами та іншими учасниками проекту, що покращує процес розробки та розгортання системи.

В діаграмі(рис 3.3) присутні компоненти системи, це пристрій користувача з якого він переглядає сторінку, сервер застосунку та сервер з базою даних.

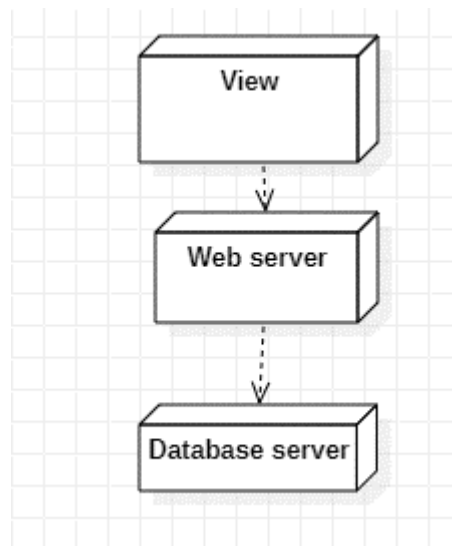


Рисунок 3.3 –Діаграма розгортання

Отже, діаграма розгортання є потужним інструментом для моделювання фізичної структури системи, виявлення проблем розгортання та планування масштабування. Вона сприяє кращому розумінню системи та полегшує комунікацію між учасниками проєкту.

### 3.1.4 Діаграма пакетів

Діаграма пакетів є інструментом який використовується для групування та організації елементів програмної системи в логічні пакети або модулі. Вона надає візуальний засіб для відображення взаємозв'язків та структури між різними компонентами системи.

Одна з основних цілей діаграми пакетів - забезпечити логічну організацію системи шляхом групування класів, інтерфейсів та інших елементів у логічні пакети. Це дозволяє зрозуміти, які елементи системи пов'язані між собою і як вони організовані на рівні модулів або компонентів. Діаграма пакетів допомагає визначити логічну структуру системи та легко розуміти, як компоненти взаємодіють один з одним [9].

Ще одна важлива функція діаграми пакетів - відображення залежностей та інтерфейсів між пакетами. Вона показує, як пакети взаємодіють між собою, які пакети використовують інші пакети та як вони спільно працюють для досягнення

функціональності системи. Це допомагає розробникам краще розуміти залежності між компонентами та впливати на архітектуру системи.

Діаграма пакетів також сприяє організації та структуруванню великих проєктів. Вона дозволяє розбити систему на логічні блоки, що полегшує розробку, тестування та управління проєктом. Використання діаграми пакетів спрощує комунікацію між учасниками проєкту та покращує загальне розуміння структури системи.

На даній діаграмі пакетів(рис. 3.4) представлено те як пов'язані модулі в проєктованій системі інтернет-книгарні.

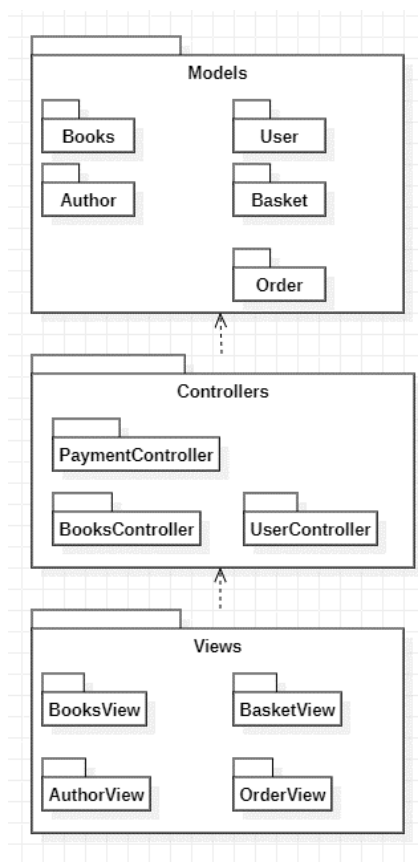


Рисунок 3.4 –діаграма пакетів

Отже, діаграма пакетів є потужним інструментом для логічної організації та візуалізації структури програмної системи. Вона допомагає зрозуміти взаємозв'язки між компонентами, виявити залежності та структурувати проєкти для більш ефективного розробки та управління.

## 3.2 Огляд технологій

Для розробки вебзастосунку інтернет-книгарні використовувались такі технології:

- HTML
- JS
- CSS
- PHP
- Laravel
- Bootstrap
- jQuery
- MySQL

Вибір правильного стеку технологій для розробки сайту є ключовим етапом, оскільки він визначає рівень функціональності, продуктивність, масштабованість і тривалість розробки даного вебпроєкту. Цей вибір надзвичайно важливий так як відштовхуючись від нього буде проводитись планування функцій, час розробки проєкту.

Правильний стек технологій забезпечує необхідні інструменти та ресурси для ефективного вирішення завдань, які перед нами стоять. Він повинен відповідати конкретним потребам проєкту та його функціональності. Вибір правильного стеку дозволяє реалізувати всі необхідні функції та взаємодію з користувачами з комфортом [10].

Правильний стек технологій також впливає на продуктивність сайту. Різні технології мають свої переваги та обмеження, і важливо зрозуміти, як вони впливатимуть на продуктивність вебпроєкту.

Крім того, вибір правильного стеку технологій має важливе значення для масштабованості нашого вебпроєкту. Якщо ви плануєте збільшувати обсяги даних або користувачів, важливо вибрати технології, які забезпечать гнучкість та масштабованість системи. Наприклад, використання розподіленої архітектури, такої

як мікросервісна архітектура, дозволить легко додавати нові сервіси та компоненти, що допоможе проєкту зростати без проблем.

Нарешті, вибір правильного стеку технологій впливає на тривалість розробки проєкту. Деякі технології мають широку спільноту розробників і велику кількість готових бібліотек і модулів, що спрощує процес розробки. Це може пришвидшити розробку проєкту і дозволити швидко й зручно використовувати готові рішення замість того, щоб розробляти певний функціонал з нуля.

### **3.2.1 Мови програмування**

PHP (Hypertext Preprocessor) є популярною мовою програмування для розробки вебдодатків з численними перевагами, які роблять її привабливою для розробників. Одна з головних переваг PHP полягає в її простоті та легкості освоєння. Вона має зрозумілий і лаконічний синтаксис, що дозволяє швидко оволодіти мовою навіть початківцям. PHP також має велику спільноту розробників, яка надає підтримку, документацію та велику кількість розширень та фреймворків. Іншою перевагою PHP є його широка підтримка та наявність на багатьох серверах та хостинг-провайдерах. Тому є можливість легко знайти підтримку для PHP на різних платформах та розгорнути проєкт без зайвих зусиль [11].

Крім того, PHP має потужні засоби для взаємодії з базами даних. Вона підтримує багато реляційних баз даних, таких як MySQL, що дозволяє ефективно працювати з даними та забезпечувати потрібну функціональність.

PHP також відома своєю гнучкістю і розширюваністю. Під час розробки та подальшої підтримки можна використовувати різні фреймворки, бібліотеки та розширення, щоб прискорити розробку і забезпечити більше можливостей для системи.

Загалом, вибір PHP для розробки вебпроєкту може бути поясненим завдяки її простоті, широкій підтримці, гнучкості та розширюваності. Вона є однією з основних мов програмування для веброзробки, і її використання може забезпечити потрібні інструменти та можливості для реалізації проєкту.

JavaScript є однією з найпопулярніших мов програмування, особливо для веброзробки, і має декілька суттєвих переваг, які роблять її привабливою для розробників.

Одна з ключових переваг JavaScript полягає у його широкому застосуванні. JavaScript використовується для створення інтерактивних вебсторінок, вебдодатків, мобільних додатків та навіть настільних програм. Він підтримується всіма сучасними веббраузерами, що робить його універсальним і доступним інструментом для веброзробки.

JavaScript також має широку спільноту розробників, що забезпечує доступ до великої кількості ресурсів, бібліотек та фреймворків. Це сприяє швидкому розробленню, оновленню та підтримці проєктів, а також дозволяє використовувати готові рішення для вирішення різних функціональних потреб.

JavaScript є мовою з високим рівнем взаємодії з користувачем. Він забезпечує можливість маніпулювання елементами сторінки, обробки подій, валідації даних та багато іншого. Це дозволяє створювати багатофункціональні та динамічні вебдодатки, які забезпечують відмінне користувацьке досвід.

Загалом, вибір JavaScript для розробки має ряд переваг: широке застосування, можливість працювати на клієнтській та серверній стороні, велика спільнота розробників та здатність створювати високоінтерактивні вебдодатки. Ці фактори роблять JavaScript потужним і ефективним інструментом для розробки вебпроєктів.

### **3.2.2 Технології frond-end розробки**

Розробка з використанням HTML та CSS має кілька суттєвих переваг, які роблять її привабливою для розробників.

По-перше, HTML та CSS є основними мовами для створення вебсайтів та вебдодатків. HTML відповідає за структуру та контент сторінки, тоді як CSS займається її візуальним оформленням. Використання цих мов дозволяє розробникам точно контролювати вигляд та розташування елементів на сторінці.[12]

По-друге, HTML та CSS мають простий і зрозумілий синтаксис, що дозволяє швидко освоїти їх навіть початківцям. Розробка з використанням HTML та CSS не вимагає глибоких знань програмування, що дозволяє швидше розпочати роботу та швидко бачити результати.

По-третє, HTML та CSS є стандартами веброботки, що означає, що їх підтримують усі сучасні веббраузери. Це забезпечує відмінну переносимість та сумісність вебпроєкту з різними пристроями та платформами.

По-четверте, HTML та CSS підтримують розширення та фреймворки, які розширюють їх можливості та дозволяють розробникам прискорити процес розробки. Наприклад, фреймворки, такі як Bootstrap, надають готові компоненти та стилі для швидкого побудови сторінок.

Загалом, розробка з використанням HTML та CSS має переваги простоти, доступності, стандартизації та розширюваності, що робить її привабливим вибором для створення вебсайтів та вебдодатків.

Розробка з використанням вище згаданого фреймворка Bootstrap має кілька переваг, які роблять її привабливою для розробників.

Одна з основних переваг Bootstrap полягає в його готових компонентах та стилях. Bootstrap надає широкий набір готових елементів, таких як кнопки, форми, навігаційні панелі, каруселі та багато іншого. Це дозволяє розробникам швидко створювати структуру та інтерфейс вебсайту без необхідності писати код з нуля. Ще одна перевага Bootstrap полягає в його адаптивності. Фреймворк включає в себе гнучку сітку, яка автоматично адаптується до різних розмірів екранів, що дозволяє створювати респонсивні вебзастосунки.

Bootstrap також забезпечує єдиний та стандартизований стиль для сайту. Він має добре продуману систему класів і стилів, що допомагає зберігати єдність дизайну на всій сторінці. Це полегшує розробку та підтримку проєкту, а також забезпечує професійний вигляд.

Крім того, Bootstrap активно підтримується та оновлюється спільнотою розробників. Згідно статистики це найпопулярніший фреймворк для фронт-

розробки(рис. 3.5). Він має велику кількість документації, прикладів та плагінів, що дозволяє розробникам швидко знайти необхідну інформацію та рішення для своїх потреб.

Загалом, розробка з використанням Bootstrap має переваги готових компонентів, адаптивності, єдності стилю та підтримки спільноти, що дозволяє розробникам прискорити процес створення вебсайтів та забезпечити їх професійний вигляд.

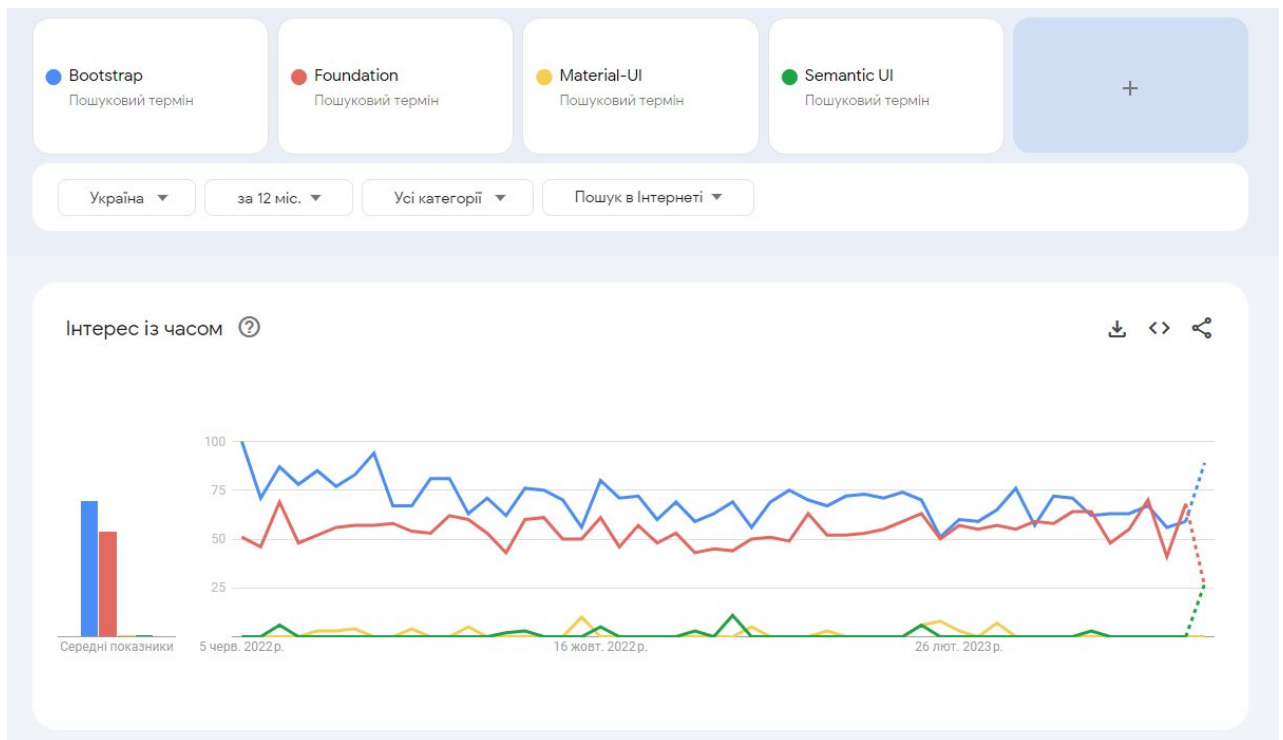


Рисунок 3.5 – Порівняння популярності фреймворків

Наступним фреймворком що використовувався для розробки є jQuery. jQuery є однією з найпопулярніших JavaScript-бібліотек для веброзробки. Протягом багатьох років, починаючи зі свого випуску у 2006 році, jQuery набув широкої популярності серед розробників і став неофіційним стандартом для роботи з JavaScript на вебсторінках.

Одна з основних переваг jQuery полягає в його простоті та лаконічності. jQuery надає простий та зрозумілий синтаксис, який значно спрощує взаємодію з DOM та обробку подій. Він дозволяє з легкістю маніпулювати елементами сторінки, змінювати їх стилі, анімацію та взаємодіяти з подіями користувача.



Ще одна перевага jQuery - його універсальність і сумісність з усіма сучасними браузерами. Він абстрагує веброзробників від різниці в реалізації функціональності браузерів і забезпечує єдиний інтерфейс для роботи з DOM. Це робить розробку більш уніфікованою та зменшує ризик виникнення помилок.

jQuery також має велику спільноту розробників, що забезпечує багато документації, рішень для поширених завдань та плагінів. Це дозволяє розробникам швидко знаходити відповіді на питання та ефективно вирішувати завдання без необхідності писати власний код з нуля.

Окрім того, jQuery забезпечує підтримку AJAX, що дозволяє взаємодіяти з сервером асинхронно без перезавантаження сторінки. Це робить розробку вебдодатків більш динамічною та ресурсозберігаючою.

Загалом, розробка з використанням jQuery має переваги простоти, універсальності, підтримки спільноти та можливості взаємодії з сервером, що дозволяє розробникам ефективно та зручно працювати з DOM та подіями на вебсторінках.

### 3.2.3 Технології back-end розробки

Для розробки інтернет-книгарні був використаний фреймворк Laravel, який є одним з найпопулярніших фреймворків PHP(рис. 3.6) для веброзробки. Laravel має широку спільноту розробників і завдяки цьому має велику кількість навчальних матеріалів, книг та туторіалів що дозволяють активно застосовувати його та швидко вирішувати поставлені завдання.[12]

Однією з основних особливостей Laravel є його зручний та зрозумілий синтаксис, що сприяє швидкому та читабельному написанню коду. Він надає розробникам потужні інструменти для реалізації різноманітних функціональностей, таких як маршрутизація, кешування, сесії, автентифікація та багато іншого.

Laravel також надає розробникам можливість використовувати популярну архітектурну модель MVC, що спрощує структурування та організацію коду. Це

дозволяє розробникам зосередитись на логіці додатку, розділивши його на моделі, представлення та контролери.

Ще одною перевагою Laravel є наявність великої кількості розширень та пакетів, які спрощують розробку. Laravel використовує менеджер залежностей Composer, що дозволяє швидко та зручно встановлювати сторонні пакети та бібліотеки.

Усі ці фактори роблять Laravel привабливим вибором для розробки інтернет-книгарні, дозволяючи забезпечити потужну та функціональну платформу для продажу та взаємодії з книжками в онлайн-середовищі.

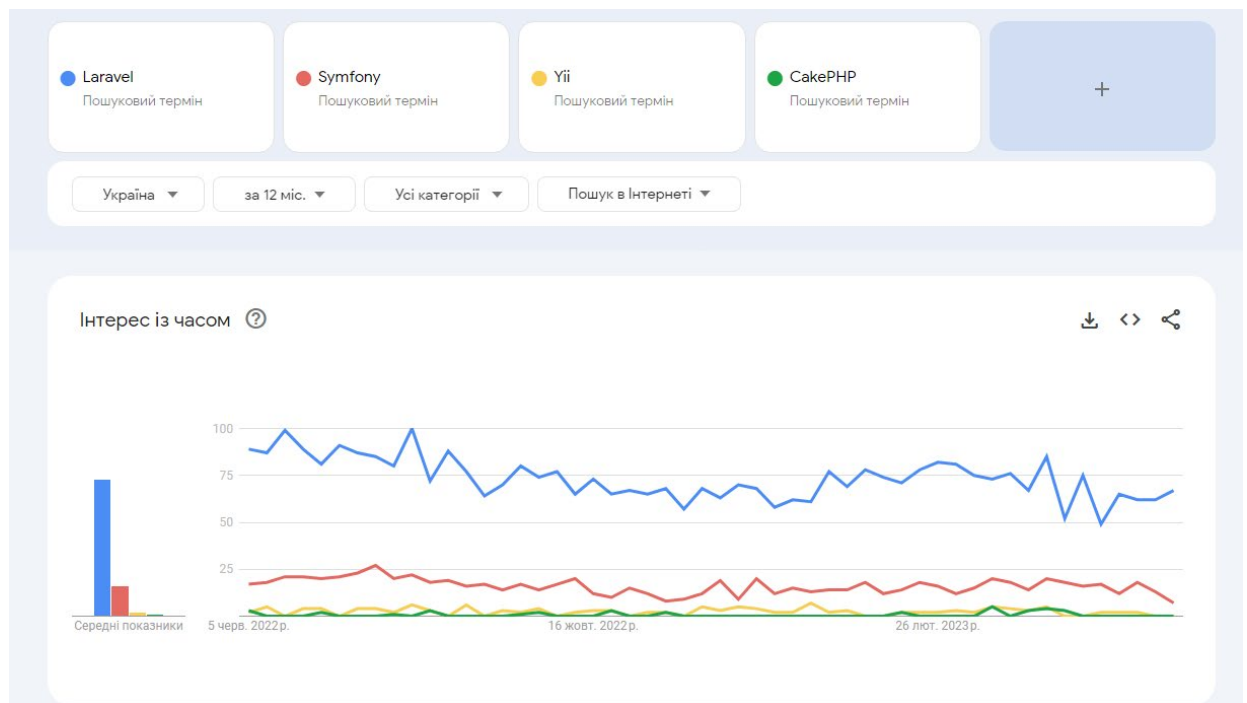


Рисунок 3.6 – Порівняння популярності фреймворків для PHP

Існує кілька типів баз даних, але для розробки інтернет-книгарні було обрано реляційну базу даних MySQL. Реляційні бази даних використовують таблиці для організації та збереження даних. Коли розробляється вебдодаток, особливо такий, як інтернет-книгарня, реляційна база даних була чудовим вибором.

Ось декілька причин, чому саме MySQL було обрано:

- Популярність: MySQL є однією з найпопулярніших реляційних баз даних. Вона має велику спільноту розробників, багато документації та підтримку, що робить її впевненим та надійним вибором для проєктів різного розміру.
- Простота використання: MySQL має зрозумілий та легкий у використанні синтаксис SQL, що спрощує розробку та підтримку бази даних. Це дозволяє команді розробників ефективно працювати з даними та виконувати потрібні запити.
- Широкий функціонал: MySQL має багатий набір функцій та можливостей, які дозволяють ефективно працювати з даними. Вона підтримує транзакції, індексацію, зовнішні ключі та багато інших функцій, які роблять її потужним інструментом для роботи з базами даних.
- Масштабованість: MySQL може масштабуватися від невеликих проєктів до великих підприємств. Вона здатна обробляти великі обсяги даних та запитів із забезпеченням швидкості та ефективності.
- Надійність та безпека: MySQL має довгу історію використання та перевірену надійність. Вона пропонує вбудовані механізми для забезпечення безпеки даних, включаючи автентифікацію, резервне копіювання та шифрування [12].

Крім того, MySQL підтримує реплікацію, що дозволяє створювати копії бази даних та забезпечувати високу доступність та надійність. Це важливо для інтернет-книгарні, оскільки дозволяє підтримувати постійний доступ до даних та уникнути втрати інформації у разі виникнення неполадок [13].

Враховуючи всі ці особливості та переваги, MySQL є відмінним вибором для розробки інтернет-книгарні, надаючи надійне та ефективне зберігання та управління даними.

### **Висновки до розділу 3**

У третьому розділі було здійснено проектування вебзастосунку інтернет-книгарні за допомогою UML-діаграм, таких як: діаграми класів, компонентів, розгортання та пакетів. Ці діаграми надають цінну інформацію та візуалізують різні аспекти системи, що допомагає розробникам краще розуміти та управляти їхнім

проєктом. Створення діаграм класів, компонентів, розгортання та пакетів є важливим етапом у розробці програмного продукту, оскільки вони надають цінну інформацію, полегшують розуміння системи та сприяють кращому управлінню проєктом. Вони допомагають розробникам зосередитися на ключових аспектах системи, виявити потенційні проблеми та зробити процес розробки більш структурованим та ефективним [14].

У процесі розробки інтернет-книгарні було проведено порівняння різних мов програмування, аналізуючи їх переваги та особливості. Це було зроблено з метою вибрати найкращий інструмент для реалізації проєкту. Кожна мова має свою унікальну набір функціональності, бібліотек та фреймворків, які сприяють швидкому розвитку та ефективній реалізації проєкту. Проведення аналізу стеку технологій допомагає виявити переваги обраних мов, зрозуміти їхні можливості та обмеження. Це дозволяє зосередитися на виборі оптимального інструменту, що підходить для виконання певних завдань та задоволення потреб проєкту. Таке порівняння допомагає забезпечити ефективну розробку, підвищити продуктивність та забезпечити якість проєкту.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

Вебзастосунок інтернет-книгарні містить 6 сторінок, таких як головна, сторінка книги, сторінка кошику, для користувача. У вебзастосунку наявні два типи користувачів: адміністратор та користувач. Ролі клієнта та адміністратора мають схожі функціональні можливості, оскільки обидва користувачі взаємодіють з системою. Однак, основною роллю є роль клієнта, оскільки вони є головними користувачами системи.

Враховуючи вимоги та функціонал системи, клієнт має можливість здійснювати CRUD-операції щодо книг. Вони можуть переглядати книги, додавати їх у кошик, читати відгуки інших клієнтів. Таким чином, функціонал системи спрямований на забезпечення комфортного та ефективного взаємодії клієнта з платформою.

### 4.1 Дизайн вебзастосунку

Для створення дизайну інтернет-книгарні було використано процес створення mock-ups. Mock-ups (макети) - це візуальні представлення інтерфейсу або дизайну, які демонструють вигляд та структуру елементів вебсторінок без функціональності. Вони є важливою складовою процесу дизайну, оскільки дозволяють розробникам та клієнтам отримати уявлення про те, як буде виглядати кінцевий продукт [15].

Для всіх користувачів доступна головна сторінка застосунку(рис. 4.1), з якого є доступ до переліку книг(рис. 4.2), вибір книг по жанру(рис. 4.3), вкладки авторизації та реєстрації, сторінки кошику.

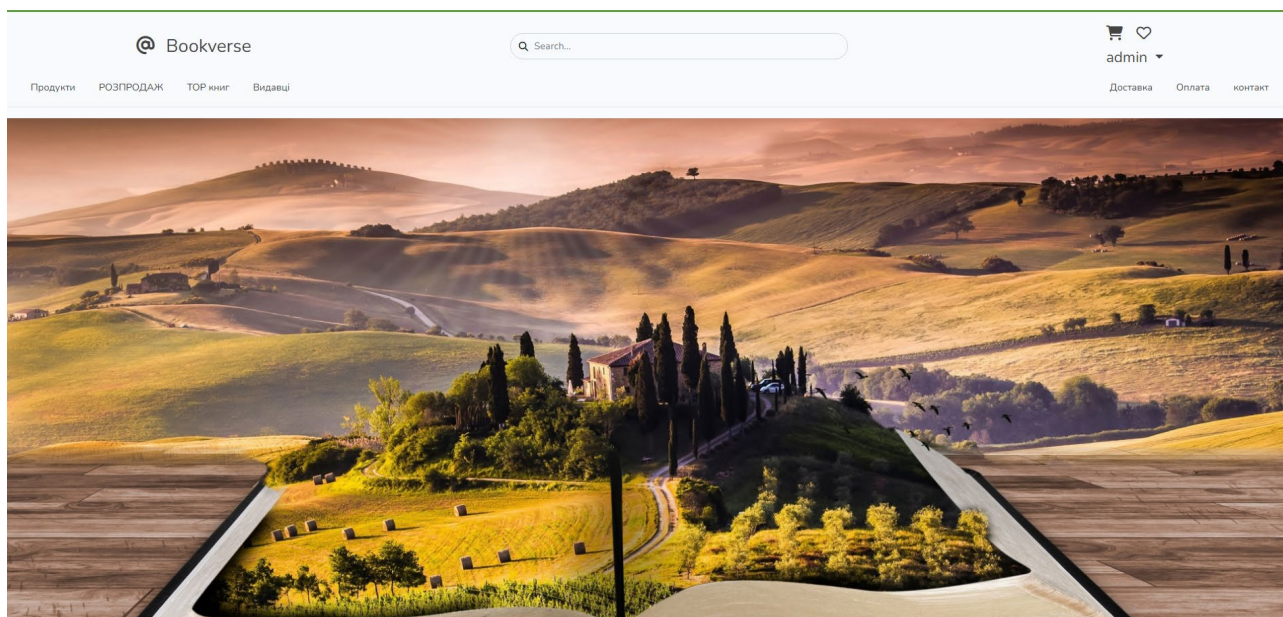


Рисунок 4.1 – Головна сторінка книгарні

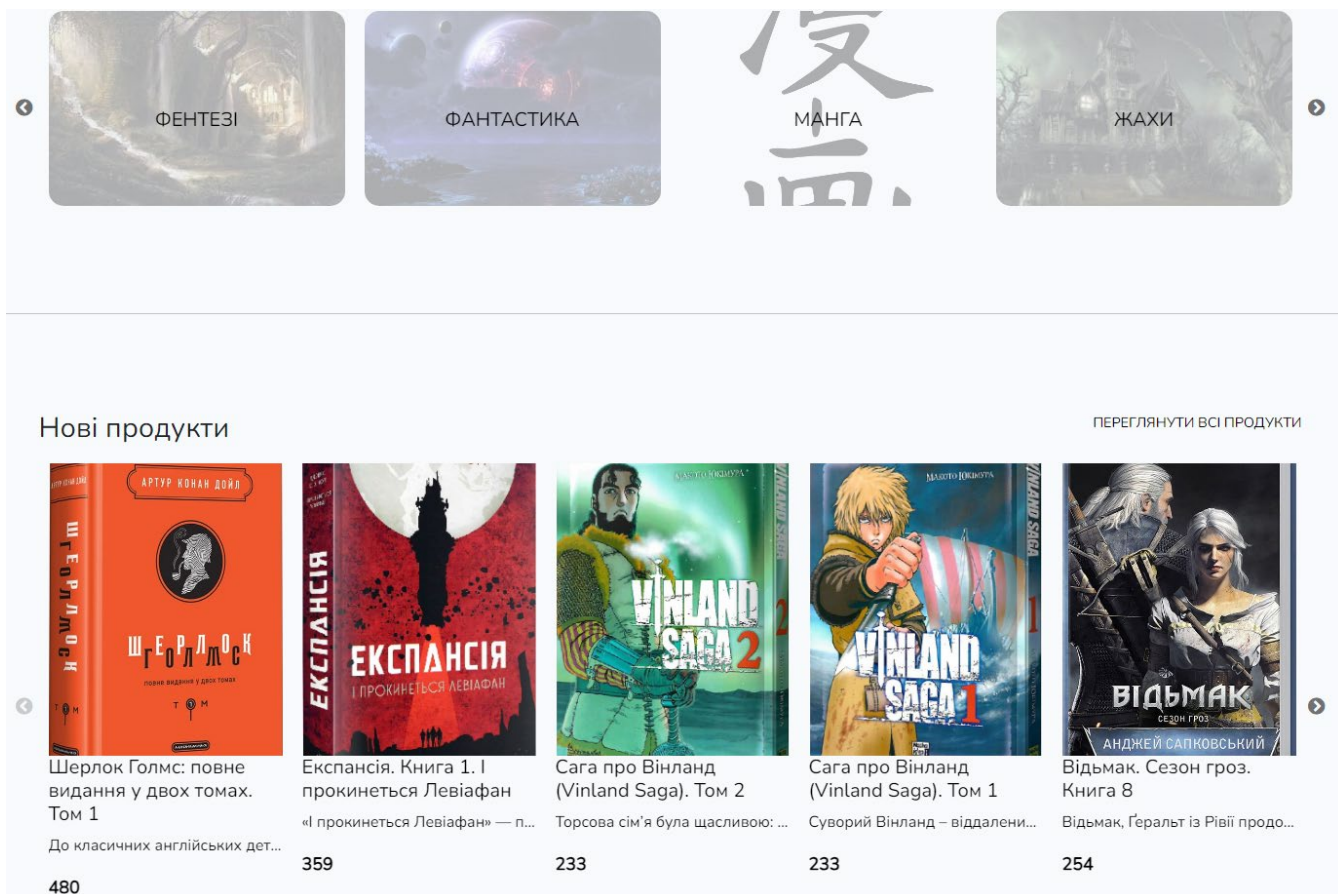


Рисунок 4.2 – Продовження основної сторінки

## Фентезі



Відьмак. Останнє бажання.  
Книга 1

«Відьмак. Останнє бажання» – це пер...

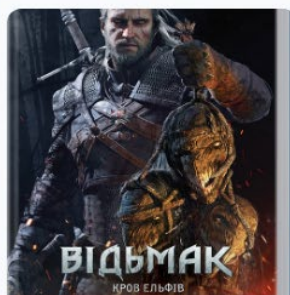
254



Відьмак. Меч призначення.  
Книга 2

Від подорожі до подорожі, від пригод...

254



Відьмак. Кров ельфів. Книга 3

Героїчна сага Анджея Сапковського п...

254



Відьмак. Час погорди. Книга 4

Героїчна сага Анджея Сапковського п...

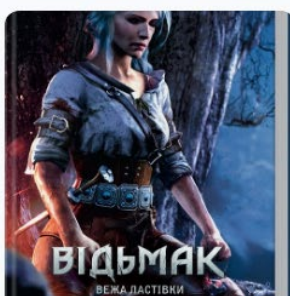
254



Відьмак. Хрещення вогнем.  
Книга 5

Поки світ магів оговтується після ката...

254



Відьмак. Вежа Ластівки. Книга 6

Диво врятувало від смерті Цірі, коли ...

254



Відьмак. Володарка Озера.  
Книга 7

Зеленоока Цірі продовжує пошуки Ге...

254



Відьмак. Сезон гроз. Книга 8

Відьмак, Геральт із Рівії продовжує св...

254

Рисунок 4.3 – Відбір книг по жанру

Також було реалізовано адаптивний інтерфейс(рис. 4.4). Адаптивний інтерфейс - це підхід до розробки вебдизайну, який забезпечує оптимальне відображення інтерфейсу на різних пристроях та розмірах екранів. Його метою є створення користувальницького досвіду, який буде зручним і доступним незалежно від пристрою, на якому відбувається перегляд вебсайту чи додатку.

Реалізація адаптивного інтерфейсу є важливою з кількох причин. По-перше, зростаюча різноманітність пристроїв та їх розмірів вимагає, щоб застосунки були пристосовані до різних екранів. Адаптивний дизайн дозволяє автоматично адаптувати розміщення, розміри та стилі елементів інтерфейсу для оптимального відображення на будь-якому пристрої(рис. 4.4), (рис. 4.5), (рис. 4.6).

По-друге, адаптивний інтерфейс покращує користувацький досвід, дозволяючи зручно взаємодіяти з вебсайтом чи додатком без необхідності масштабування чи горизонтального прокручування. Він забезпечує зручне розташування та читабельність контенту, а також інтуїтивно зрозумілу навігацію, що покращує задоволення від використання продукту.

Крім того, адаптивний дизайн є важливим з точки зору пошукової оптимізації. Оскільки пошукові системи враховують мобільну дружність сайту при визначенні рейтингу в пошукових результатах, наявність адаптивного інтерфейсу сприяє поліпшенню позицій в пошукових системах.

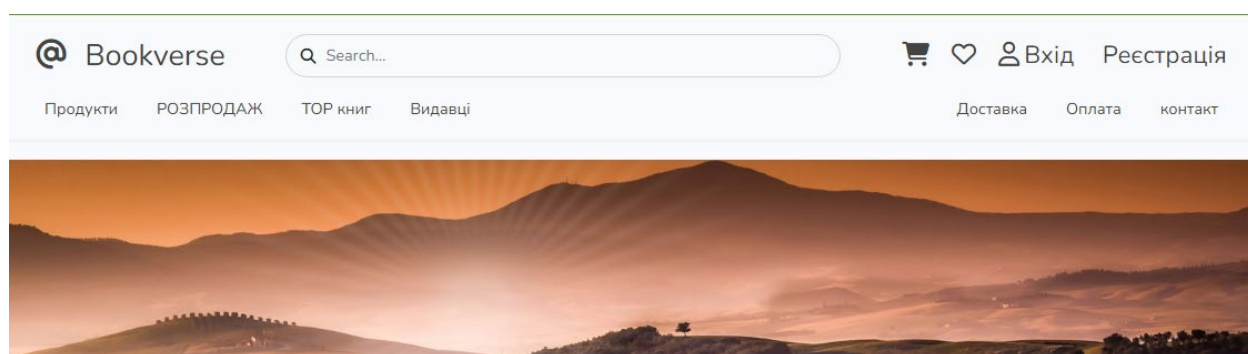


Рисунок 4.4 – Стандартний інтерфейс

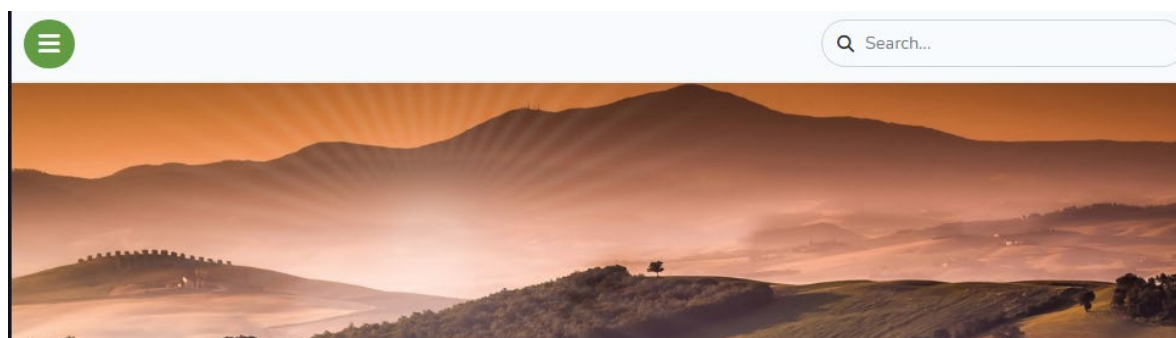


Рисунок 4.5 – Інтерфейс при зменшенні розширення



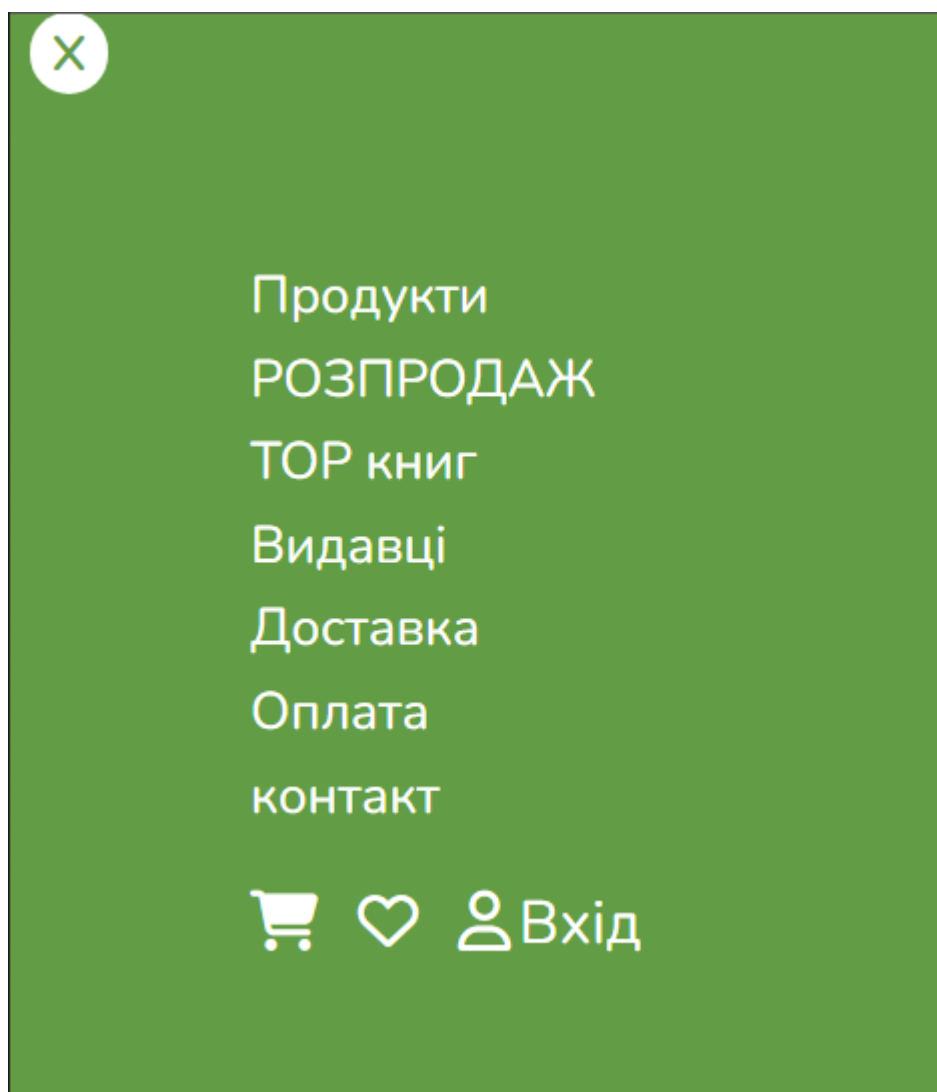


Рисунок 4.6 – Інтерфейс при зменшенні розширення

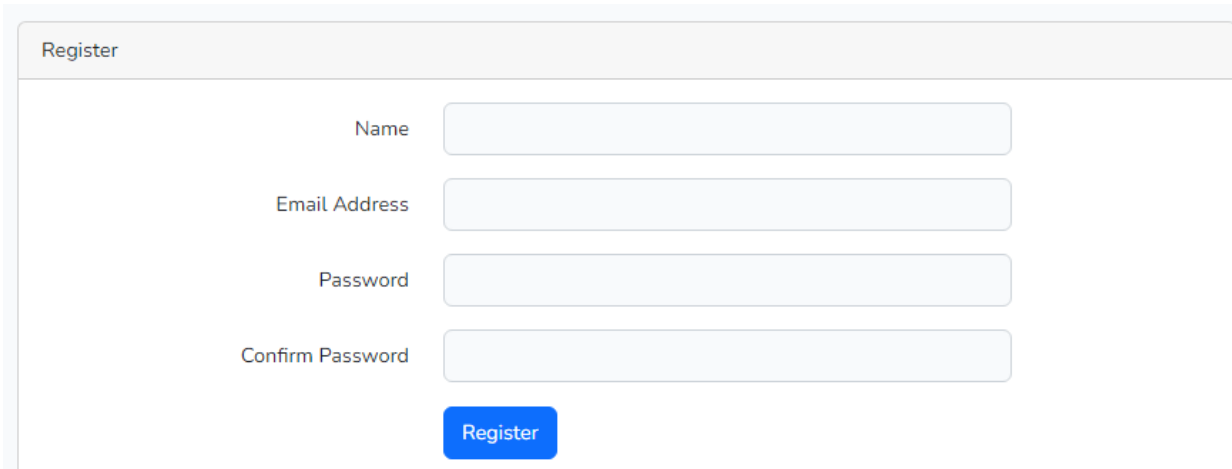
Отже, реалізація адаптивного інтерфейсу є необхідною для покращення користувальницького досвіду, забезпечення доступності на різних пристроях не зважаючи на роздільну здатність та розмір екрану користувача.

#### **4.2 Реєстрація та авторизація**

Реалізація реєстрації на вебзастосунку має велику важливість для різних сторін, як користувачів, так і власників сайту. Перш за все, реєстрація дозволяє сайту ідентифікувати та розрізняти користувачів. Це дозволяє сайту персоналізувати досвід користувача, зберігати його налаштування та вподобання.

Крім того, реєстрація відкриває можливості для взаємодії між користувачами та власниками сайту. Зареєстровані користувачі можуть залишати коментарі, відгуки, взаємодіяти з іншими користувачами або звертатися до підтримки [17].

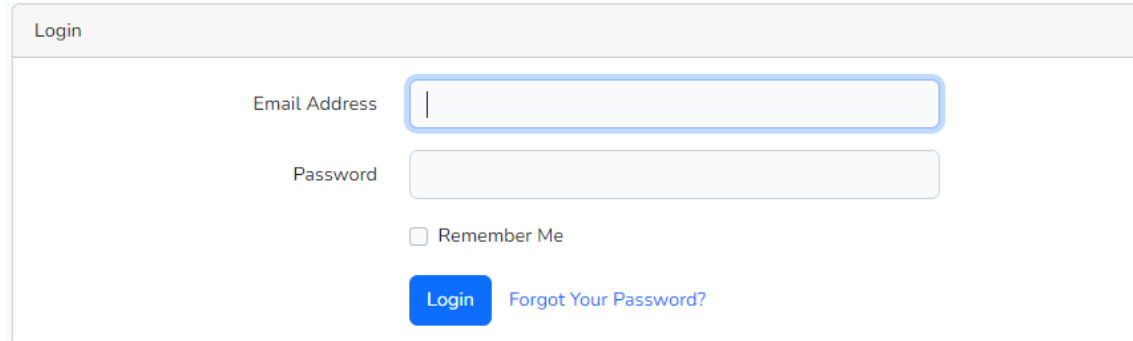
В цілому, реалізація реєстрації(рис. 4.7-4.8) на вебсайті є важливим елементом для забезпечення індивідуалізації, безпеки, контролю доступу та взаємодії з користувачами. Це дозволяє створити більш персоналізований та взаємодійний досвід для користувачів та забезпечити успішну роботу вебсайту для його власників.



The image shows a registration form with the following fields and elements:

- Title: Register
- Form fields: Name, Email Address, Password, Confirm Password
- Submit button: Register

Рисунок 4.7 – Форма реєстрації користувача



The image shows a login form with the following fields and elements:

- Title: Login
- Form fields: Email Address, Password
- Checkbox: Remember Me
- Submit button: Login
- Link: Forgot Your Password?

Рисунок 4.8 – Форма автентифікації користувача

За відображення полів реєстрації відповідає файл `register.blade.php`(рис. 4.9). Цей файл містить HTML-розмітку та Laravel Blade-директиви для відображення форми реєстрації.

```

<div class="row mb-3">
  <label for="name" class="col-md-4 col-form-label text-md-end">{{ __('Name') }}</label>

  <div class="col-md-6">
    <input id="name" type="text" class="form-control @error('name') is-invalid @enderror" name="name" value="{{ old('name') }}" required autocomplete="name" autofocus>

    @error('name')
    <span class="invalid-feedback" role="alert">
      <strong>{{ $message }}</strong>
    </span>
    @enderror
  </div>
</div>

<div class="row mb-3">
  <label for="email" class="col-md-4 col-form-label text-md-end">{{ __('Email Address') }}</label>

  <div class="col-md-6">
    <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email">

    @error('email')
    <span class="invalid-feedback" role="alert">
      <strong>{{ $message }}</strong>
    </span>
    @enderror
  </div>
</div>

```

Рисунок 4.9 – Фрагмент коду register.blade.php

За логіку реалізації відповідає файл RegistersUsers.php(рис. 4.10).

```

public function register(Request $request)
{
    $this->validator($request->all())->validate();

    event(new Registered($user = $this->create($request->all())));

    $this->guard()->login($user);

    if ($response = $this->registered($request, $user)) {
        return $response;
    }

    return $request->wantsJson()
        ? new JsonResponse([], 201)
        : redirect($this->redirectPath());
}

```

Рисунок 4.10 – Фрагмент коду RegistersUsers.php

Крім того, реєстрація дозволяє встановлювати контроль доступу до певного контенту або функцій на сайті. Можна обмежити певні можливості тільки для зареєстрованих користувачів або надати різні рівні доступу для різних типів користувачів. Це допомагає забезпечити безпеку та конфіденційність інформації, а також керувати взаємодією адміністраторів з відвідувачами.

### 4.3 Додавання книг та жанрів

Додавання книг та жанри є одною з основних функціональностей для інтернет-книгарні.

При створенні книги в базі даних MySQL, для збереження відповідної інформації, можна створити таблицю, яка міститиме наступні стовпці(рис 4.11):

- Назва (Title): Назва книги, яка зберігається у текстовому форматі.
- Автор (Author): Ім'я або ім'я автора книги, яке також зберігається у текстовому форматі.
- Ціна (Price): Вартість книги, яка може бути числовим значенням або десятковим числом.
- ISBN: Унікальний номер книги, який використовується для її ідентифікації та відстеження. Зберігається у текстовому форматі.
- Наявність (Availability): Поле, яке вказує, чи є книга в наявності. Може бути логічним значенням, наприклад, TRUE або FALSE.
- Жанр (Genre): Інформація про жанр книги, яка зазвичай зберігається у текстовому форматі або може мати відповідну ID-посилання на таблицю жанрів.
- Фото (Photo): Зображення обкладинки книги, яке може зберігатися у форматі URL або як шлях до файлу.
- Опис (Description): Текстовий опис книги, який дає коротку інформацію про неї.

Кожен запис (рядок) у таблиці буде містити значення для кожного з цих стовпців, що дозволяє зберігати і організувати дані про книги у структурованому форматі. Такі дані можуть бути використані для відображення інформації про книги на сторінках магазину, пошуку, фільтрації та інших операцій, пов'язаних з каталогом книг[18].

## Додати книгу

messages.back


Назва	<input type="text"/>
Автор	<input type="text"/>
Ціна	<input type="text"/>
ISBN	<input type="text" value="111-222-333-444-5"/>
В наявності	<input type="checkbox"/>
Жанр	<input type="text" value="Фантастика"/>
Категорія	<input type="text" value="Роман"/>
Видавець	<input type="text" value="Книжковий клуб «Клуб Сімейного Дозвілля»"/>
Зображення	<div style="border: 1px solid #ccc; width: 100px; height: 100px; background-color: #e0ffe0; display: flex; align-items: center; justify-content: center;"></div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>Вибрати файли</span><span>Файл не вибрано</span></div>
Опис	<input type="text"/>

Рисунок 4.11 – Відображення можливих дій з книгою після створення

Це включає в себе CRUD операції, що означає, що ми можемо створювати (Create), читати (Read), оновлювати (Update) та видаляти (Delete) дані пов'язані з книгами та жанрами(рис 4.12).



Рисунок 4.12 – Відображення можливих дій з книгою після створення

Після додавання даних в MySQL базу даних, вони зберігаються у вигляді таблиць. Для книг може бути створена окрема таблиця, де кожна книга буде мати свій рядок або запис. Кожен стовпець у таблиці буде відповідати певному атрибуту книги, наприклад, ID, назва, автор, рік видання тощо. Аналогічно, для жанрів може бути створена окрема таблиця, де кожен рядок представлятиме певний жанр [19].

Додавання жанрів відбувається схожим чином(рис. 4.13), та це є важливою частиною застосунку так як саме по ній відбувається основний розподіл книг та виведення книг до користувачів(рис. 4.14).

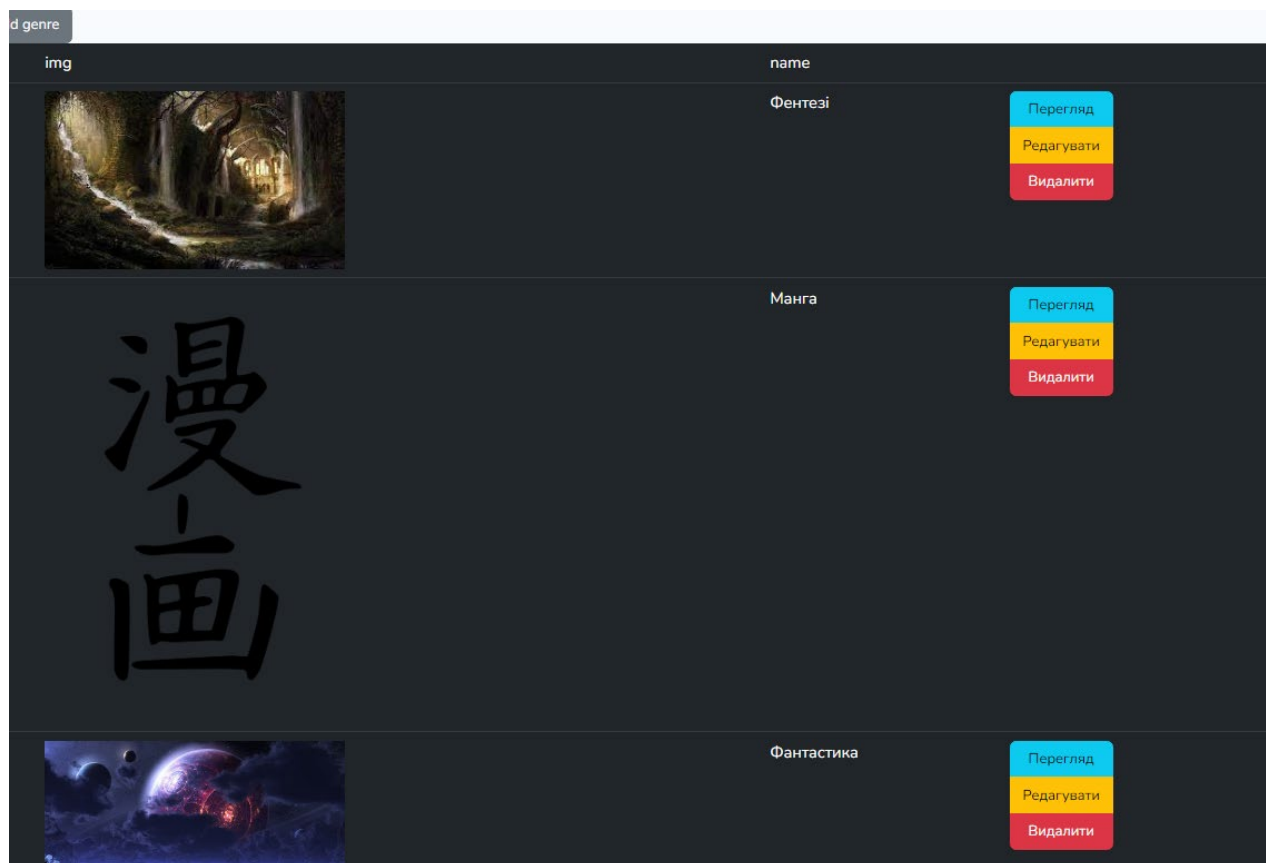


Рисунок 4.13 – Панель керування жанрами

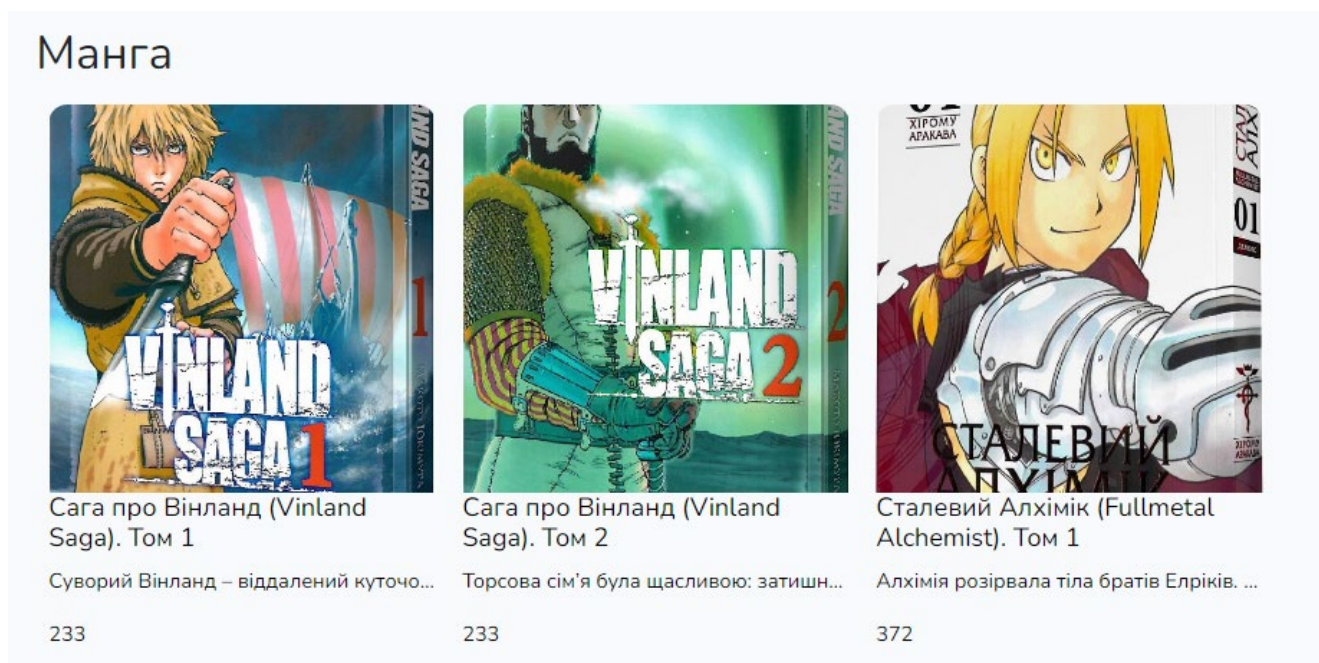


Рисунок 4.14 – Фільтрація книг по жанру

Після додавання нової книги до бази даних, дані про цю книгу будуть вставлені в відповідну таблицю. Кожен стовпець у рядку буде заповнений відповідною інформацією про книгу, яку було додано. Потім ці дані можуть бути використані для відображення книги на сторінці продукту в інтернет-магазині та для реалізації інших операцій, таких як пошук, фільтрація тощо.

#### **4.4 Тестування**

Тестування вебзастосунку має важливе значення для його успішної розробки та функціонування. Основна мета тестування полягає у виявленні помилок, дефектів та проблем, а також забезпеченні високої якості продукту. Існує кілька ключових причин, чому тестування є необхідною функціональною частиною процесу розробки вебзастосунків.

По-перше, тестування допомагає виявити помилки та проблеми в роботі вебзастосунку. Це можуть бути помилки у функціональності, взаємодії з користувачем, продуктивності або інші проблеми, які можуть вплинути на якість та надійність продукту. Виявлення цих проблем на під розробки дозволяє виправити їх перед залученням продукту до використання. Тестування сприяє покращенню продуктивності вебзастосунку. Воно допомагає виявити проблеми швидкодії, зависання або необхідність збільшити здатність системи до масштабування. Це дозволяє розробникам знайти причини таких проблем та скорегувати відповідні покращення для оптимізації роботи вебзастосунку [20]. Під час тестувань було проведено тестування основних та найбільш важливих елементів системи, таких як: реєстрація, авторизація та оформлення замовлення. Дані щодо проведення тестувань будуть наведені в наступних таблицях.



Таблиця 4.1 – Процес реєстрація у вебзастосунку

Діючі особи	Незареєстрований користувач
Мета	Створити обліковий запис
Передумова	Користувач не зареєстрований у системі
Успішні сценарії:	
<ul style="list-style-type: none"> <li>– Користувач переходить до форми реєстрації;</li> <li>– Користувач заповнює необхідні дані (Логін, електрона пошта, пароль);</li> <li>– Користувач надсилає запит до системи і бази даних;</li> <li>– Система оброблює обліковий запис;</li> <li>– Система зберігає обліковий запис;</li> <li>– Система переводить користувача на головну сторінку;</li> </ul>	
Сценарій успішний. Збережено новий обліковий запис.	
Розширенні сценарії:	
<ul style="list-style-type: none"> <li>– Користувач водить не всі необхідні дані. Результат: система не створює акаунт;</li> <li>– Користувач водить не дійсні дані електронної пошти. Результат: система не створює акаунт;</li> <li>– Користувач водить дані що вже задіяні в системі. Результат: система не створює акаунт;</li> </ul>	
Усі сценарії розширення успішно виконані.	

Таблиця 4.2 – Процес авторизації у вебзастосунку

Діючі особи	Користувач
Мета	Війти в свій обліковий запис
Передумова	Користувач не авторизований у системі
Успішні сценарії:	
<ul style="list-style-type: none"> <li>– Користувач переходить до форми авторизації;</li> <li>– Користувач заповнює необхідні дані (електрона пошта, пароль);</li> </ul>	

## Кінець таблиці 4.2

<ul style="list-style-type: none"> <li>– Користувач надсилає запит до системи і бази даних;</li> <li>– Система оброблює дані;</li> </ul> <p>Система переводить користувача на головну сторінку</p>
Сценарій успішний. Авторизація успішна.
<p>Розширенні сценарії:</p> <ul style="list-style-type: none"> <li>– Користувач водить не всі необхідні дані. Результат: Авторизація не відбулась;</li> <li>– Користувач водить не дійсні дані електронної пошти. Результат: Авторизація не відбулась;</li> </ul>
Усі сценарії розширення успішно виконані.

Таблиця 4.3 – здійснення замовлення

Діючі особи	Користувач
Мета	Оформити замовлення
Передумова	Користувач зареєстрований у системі
<p>Успішні сценарії:</p> <ul style="list-style-type: none"> <li>– Користувач обирає потрібну книгу та додає її до кошика;</li> <li>– Користувач переходить до кошику;</li> <li>– Користувач водить необхідні дані(номер телефону, адресу доставки) та обирає спосіб оплати;</li> <li>– Система оброблює замовлення;</li> <li>– Система переводить користувача на головну сторінку;</li> </ul>	
Сценарій успішний. Замовлення оформлено.	
<p>Розширенні сценарії:</p> <ul style="list-style-type: none"> <li>– Користувач водить не всі необхідні дані. Результат: система не дозволяє оформити замовлення;</li> <li>– Користувач не обирає спосіб оплати або доставки. Результат: система не дозволяє оформити замовлення;</li> </ul>	

## Кінець таблиці 4.3

– Користувач не проводить оплату замовлення. Результат: система не дозволяє оформити замовлення;
Усі сценарії розширення успішно виконані.

Тестування інтернет-книгарні дозволило перевірити функціональність та ефективність різних аспектів платформи. Виявлені помилки включали, наприклад, проблеми з відображенням сторінок, неправильне функціонування пошуку або труднощі з оформленням замовлень. Завдячуючи тестуванню ці помилки були виявлені та задокументовані, що дозволило ефективно їх усунути.

Також, завдяки тестуванню, було виявлено й інші аспекти, які можуть вплинути на функціонування інтернет-книгарні, такі як продуктивність, масштабованість та безпека. Це дало змогу розробникам усунути виявлені проблеми та забезпечити надійну та безперебійну роботу платформи.

#### **Висновки до розділу 4**

У цьому розділі було представлено основні елементи розробленої системи для інтернет-книгарні. Було описано найбільш важливі частини реалізованого вебзастосунку інтернет-книгарні. Також були розглянуті функціонал системи, включаючи можливості користувача з виконання CRUD-операцій, додавання нового наповнення застосунку від адміністратора, механізм реєстрації та авторизації.

Було проведено тестування вебзастосунку, де були представлені різні сценарії тестування. Тестування допомогло перевірити правильність функціонування системи, її відповідність вимогам та виявлення можливих помилок чи проблем. Завдяки проведеному та занотованому тестуванню було успішно виправлено знайдені помилки.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було реалізовано вебзастосунок інтернет-книгарні з можливістю перегляду та замовлення книг.

Для досягнення поставленої мети було виконано такі завдання:

- аналіз предметної сфери.
- обговорення деталей проєкту.
- проведення моделювання та планування застосунку.
- створення дизайну за допомогою mockup.
- розробка backend-частини.
- розробка frontend-частини.
- тестування вебзастосунку.

Було проведено детальний процес визначення вимог до застосунку, проведено моделювання та планування інтернет книгарні. Для цього було проведено аналіз аналогів та визначено функціональні вимоги до розроблюваного застосунку, а також були створені сценарії використання. Для забезпечення зрозумілості принципів роботи застосунку, якості та ефективності були побудовані наступні діаграми:

- діаграма прецедентів;
- діаграма взаємодії;
- діаграма станів;
- діаграма класів;
- діаграма компонентів;
- діаграма розгортання;
- діаграма пакетів.

Було проведено аналіз застосунків-аналогів, та визначено їх переваги та недоліки, та за результатами аналізу були сформовані специфікації вимог. Згідно специфікації вимог було реалізовано заплановану backend та frontend-частини ПЗ.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Scholz M., Seidl M. UML @ Classroom: An Introduction to Object-Oriented Modeling: 2015th Edition 2015 p. P. 25 URL: [www.academia.edu/42042403](http://www.academia.edu/42042403) (Last access 05.05.2023)
- 2) Scott W. Ambler, The Elements of UML™ 2.0 Style 2005 P. 86 UML: <https://www.cambridge.org/core/books/elements-of-uml-20-style> (Last access 05.05.2023)
- 3) Martin Fowler UML Distilled: A Brief Guide to the Standard Object Modeling Language 2013 P. 80 URL: <https://martinfowler.com/books/uml.html>. (Last access 05.05.2023)
- 4) Craig Larman Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development 2004 p. P. 52 (Last access 06.05.2023)
- 5) Alan D., Barbara H. Wixom, Tegarden D., Systems Analysis and Design with UML 4th Edition 2012 p. P. 238 URL: <http://www.seu1.org/files/level5/IT243/Book%20-%20Systems%20Analysis%20Design%20with%20UML%20v%202%20Fourth%20Edition.pdf> (Last access 06.05.2023)
- 6) Rumbaugh J. The Unified Modeling Language User Guide by Grady Booch 2005 p. P. 91 (Last access 07.05.2023)
- 7) Rumbaugh J., Jacobson I. The Unified Modeling Language Reference Manual 2004 p. P. 15 (Last access 07.05.2023)
- 8) Delligatt L. SysML Distilled: A Brief Guide to the Systems Modeling Language 2013 p. P. 43 (Last access 08.05.2023)
- 9) Russ M., Hamilton K. Learning UML 2.0: A Pragmatic Introduction to UML 2006p. URL: <https://books.google.com.ua/books?id=QhiA6vT56E4C&printsec> P. 75 (Last access 05.05.2023)
- 10) David T., Hunt A. The Pragmatic Programmer: your journey to mastery, 2nd Edition 2019 p. URL: <https://www.oreilly.com/library/view/the-pragmatic-programmer/9780135956977/> P. 63 (Last access 05.05.2023)

- 11) Lerdorf R., Tatroe K. Programming PHP Second Edition URL: P. 34 [https://books.google.com.ua/books/about/Programming\\_PHP.html?id=7OjvOmol3CcC&redir\\_esc=y](https://books.google.com.ua/books/about/Programming_PHP.html?id=7OjvOmol3CcC&redir_esc=y) (Last access 05.06.2023)
- 12) Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 2018 p. P. 312 <https://www.oreilly.com/library/view/learning-php-mysql> (Last access 05.06.2023)
- 13) Nixon R. Learning PHP, MySQL Java. Script: Step-by-Step Guide to Creating Dynamic Websites 6th Edition, 2021 p. P. 172
- 14) Beaulieu A. Learning SQL: Generate, Manipulate, and Retrieve Data 3rd Edition 2020 p. P. 80
- 15) Addy O. Learning JavaScript Design Patterns 2nd Edition, 2023 p. P. 250
- 16) Duckett J. Front-End Back-End Development with HTML, CSS, JavaScript, jQuery, PHP, and MySQL 1st Edition 2020 p. P. 183
- 17) Duckett J. PHP & MySQL: Server-side Web Development 1st Edition 2022p. P. 236
- 18) Ullman L. PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide 4th Edition 2014 p. P. 90
- 19) Glaman M., Quillen K. Drupal 10 Development Cookbook: Practical recipes to harness the power of Drupal for building digital experiences and dynamic websites, 3rd Edition 3rd ed. Edition 2023 p. P. 130
- 20) Myers J. "The Art of Software Testing" by Glenford 2011 p. P. 40 URL: [http://www.51testing.com/N\\_download/lib/TestingTechDL/ArtofSoftwareTesting.pdf](http://www.51testing.com/N_download/lib/TestingTechDL/ArtofSoftwareTesting.pdf) (Last access 05.06.2023)