

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген ДАВИДЕНКО

«__» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
«Сучасна онлайн-бібліотека з інтеграцією функцій соціальних мереж.
Front-end частина»

Спеціальність 121 Інженерія програмного забезпечення
Освітня програма «Інженерія програмного забезпечення»

Здобувач

Максим МИЧКО

«__» _____ 20__ р.

Керівник роботи

PhD, доцент (б.в.з.)

Катерина АНТІПОВА

«__» _____ 20__ р.

Миколаїв – 2025

Завдання на виконання кваліфікаційної роботи

Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступінь	Бакалавр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген ДАВИДЕНКО

« ____ » _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну бакалаврську роботу здобувача

Мичко Максим

1. Тема кваліфікаційної роботи «Сучасна онлайн-бібліотека з інтеграцією функцій соціальних мереж. Front-end частина» затверджена наказом ректора ЧНУ ім. Петра Могили № 315 від «13» листопада 2024 р.
2. Строк представлення кваліфікаційної роботи « ____ » _____ 2025 р.
3. Очікуваний результат роботи та початкові дані якщо такі потрібні.
4. Перелік питань, що підлягають розробці:
 - аналіз предметної області;
 - розробка проєктних рішень застосунку;
 - моделювання та конструювання ПЗ;
 - формування структури бази даних;

- розробка інтерфейсу користувача;
- розробка клієнтської частини вебзастосунку.

5. Перелік графічних матеріалів: презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Дата видачі завдання « ____ » _____ 20 ____ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Сучасна онлайн-бібліотека з інтеграцією функцій соціальних мереж.
Front-end частина.

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КБР	21.10.2024	13.11.2024	Виконано
2.	Огляд літератури за темою роботи	06.01.2025	10.11.2025	Виконано
3.	Складання календарного плану КБР	13.01.2025	15.01.2025	Виконано
4.	Аналіз предметної області	16.01.2025	21.01.2025	Виконано
5.	Розробка проєктних рішень	22.01.2025	27.02.2025	Виконано
6.	Моделювання та конструювання ПЗ	11.02.2025	20.05.2025	Виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	21.05.2025	23.05.2025	Виконано
8.	Відгук керівника КБР	25.05.2025	26.05.2025	Виконано
9.	Оформлення КБР та презентації	27.05.2025	27.05.2025	Виконано
10.	Попередній захист	28.05.2025	28.05.2025	Виконано
11.	Рецензування			Виконано
12.	Завершення оформлення КБР та презентації			Виконано
13.	Захист кваліфікаційної роботи	24.06.2025	24.06.2025	Виконано

Здобувач _____

Максим МИЧКО

«__» _____ 20__ р.

Керівник роботи

PhD, доцент (б.в.з.)

Катерина АНТИПОВА

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Сучасна онлайн-бібліотека з інтеграцією функцій соціальних мереж.

Front-end частина»

Здобувач 409 групи гр.: Мичко Максим

Керівник: PhD, доцент (б.в.з.), Антіпова Катерина

Актуальність обраної теми зумовлена потребою в розвитку інноваційних рішень для покращення функціональності онлайн-бібліотек, що дозволяють користувачам не лише читати книги, а й активно брати участь у літературному житті. Інтеграція соціальних функцій є важливим кроком до створення унікального користувацького досвіду та сприяє популяризації електронних бібліотек як сучасних культурних центрів. Враховуючи зростаючу роль цифрових технологій у повсякденному житті, розробка зручного та інтерактивного інтерфейсу є важливим завданням.

Об'єктом роботи є процес пошуку, використання та розповсюдження електронних літературних творів.

Предметом роботи є інструменти створення front-end частини вебзастосунку онлайн-бібліотеки з інтеграцією функцій соціальних мереж.

Метою роботи є розробка front-end частини онлайн-бібліотеки з інтеграцією функцій соціальних мереж для популяризації електронних літературних ресурсів. Особливу увагу приділено інтеграції соціальних функцій, що сприяють активній взаємодії користувачів.

Для досягнення цієї мети було поставлено такі завдання:

- провести аналіз існуючих рішень у сфері онлайн-бібліотек та соціальних платформ;
- визначити ключові функціональні вимоги до інтерфейсу користувача;
- обґрунтувати вибір технологій для реалізації front-end частини проєкту;
- розробити архітектуру клієнтської частини вебдодатку;
- створити прототип інтерфейсу, що включає інтеграцію соціальних функцій;

– оцінити ефективність запропонованого рішення та можливості його подальшого вдосконалення.

У результаті виконання кваліфікаційної роботи було проаналізовано існуючі рішення, визначено ключові вимоги до інтерфейсу, розроблено прототип та реалізовано Front-end частину онлайн-бібліотеки з інтеграцією соціальних функцій. Платформа надає користувачам можливість реєстрації, інтерактивного виконання завдань та адаптивний інтерфейс, що забезпечує комфортне використання на різних пристроях, незалежно від їх розміру та типу.

Робота складається з вступу, чотирьох розділів та висновків. У вступі визначено актуальність теми, сформульовано мету, об'єкт, предмет та завдання роботи.

У першому розділі здійснено огляд предметної області та визначено ключові вимоги до вебплатформи. Другий розділ присвячений опису архітектури системи, обґрунтуванню вибору технологій та засобів реалізації. У третьому розділі розглянуто процес розробки користувацького інтерфейсу та серверної частини. Четвертий розділ містить результати тестування функціоналу платформи та аналіз її продуктивності. У висновках узагальнено отримані результати та окреслено можливі напрямки подальшого розвитку.

КРБ викладена на ____ сторінках, вона містить ____ розділи, ____ ілюстрацій, ____ таблиці, ____ джерел в переліку посилань.

Ключові слова: веброзробка, користувацький інтерфейс, онлайн-бібліотека, персоналізація, соціальні мережі, Front-end.

ABSTRACT

to the qualifying bachelor's thesis

«Modern online library with integration of social media features. Front-end part»

Applicant of the 409 gr: Mychko Maxim

Supervisor: PhD, Associate professor (w/o a. d.), Kateryna Antipova

The relevance of this topic is driven by the need to develop innovative solutions to improve the functionality of online libraries, allowing users not only to read books but also to actively participate in literary life. Integration of social functions is an important step towards creating a unique user experience and promoting digital libraries as modern cultural centers. Given the growing role of digital technologies in everyday life, developing a user-friendly and interactive interface is an important task.

The object of the work is the process of searching, using and distributing electronic literary works.

The subject of the work is the methods and technologies for creating the front-end part of an online library web application with the integration of social networking functions, such as recommendations, book discussions, ratings, and reviews.

The purpose of the work is to develop the front-end part of an online library with the integration of social networking functions in order to popularize electronic literary resources. Particular attention is paid to the integration of social functions that promote active user interaction.

To achieve this goal, the following tasks were set:

- analyze existing solutions in the field of online libraries and social platforms;
- identify key functional requirements for the user interface;
- justify the choice of technologies for the front-end part of the project;
- develop the architecture of the client part of the web application;
- create a prototype interface that includes the integration of social functions;
- evaluate the effectiveness of the proposed solution and the possibilities for its further improvement.

As a result of the qualification work, existing solutions were analyzed, key interface requirements were identified, a prototype was developed and the front-end part of the online library with the integration of social functions was implemented. The platform provides users with the ability to register, interactively complete tasks, and an adaptive interface that ensures comfortable use on various devices, regardless of the

The work consists of an explanatory note, which includes an introduction, four chapters and conclusions. In the introduction defines the relevance of the topic, formulates the purpose, object, subject, and objectives of the work.

The first section provides an overview of the subject area and identifies the key requirements for a web platform. The second section is devoted to the description of the system architecture, justification of the choice of technologies and implementation tools. The third section describes the process of developing the user interface and the server side. The fourth section contains the results of testing the platform's functionality and analysis of its performance. The conclusion summarizes the results and outlines possible directions for further development.

The bachelor's thesis consists of ____ pages, it contains ____ sections, ____ illustrations, ____ tables, ____ sources in the list of references.

Keywords: Front-end, online library, personalization, social networks, user interface, web development.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІТИЧНИЙ ОГЛЯД ЦИФРОВИХ ПЛАТФОРМ.....	7
1.1 Основні вимоги до цифрових платформ.....	7
1.2 Компоненти платформи літературних творів.....	10
1.3 Огляд існуючих аналогів	14
Висновки до розділу 1.....	20
2 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	21
2.1 Вибір технологій та інструментів для розробки фронтенду.....	21
2.2 Формування специфікації вимог до програмного забезпечення (SRS).....	26
Висновки до розділу 2.....	29
3 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	30
3.1 Побудова UML-моделей.....	30
3.2 Опис сценаріїв та інтерфейса користувача.....	37
Висновки до розділу 3.....	47
4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	49
4.1 Реалізація програмного забезпечення	49
4.2 Тестування програмного забезпечення	52
4.3 Інструкція користувача	56
Висновки до розділу 4.....	59
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	63

ПЕРЕЛІК СКОРОЧЕНЬ

AI	—	Artificial Intelligence
API	—	Application Programming Interface
CSS	—	Cascading Style Sheets
DOM	—	Document Object Model
HTTP	—	HyperText Transfer Protocol
Npm	—	Node Package Manager
PDF	—	Portable Document Format
SEO	—	Search Engine Optimization
SPA	—	Single Page Application
SRS	—	Software Requirements Specification
UI	—	User Interface
UX	—	User Experience
ПЗ	—	програмне забезпечення

ВСТУП

У сучасному світі інформаційні технології стали невід'ємною частиною життя кожної людини. Особливо важливу роль вони відіграють у сфері культури, освіти та взаємодії суспільства. Онлайн-бібліотеки перетворилися на потужний інструмент збереження, поширення та популяризації літературних творів у цифровому форматі. Усе більше людей надають перевагу електронним книгам та інтерактивним платформам, що дозволяють швидко отримати доступ до потрібної інформації, незалежно від географічного розташування.

Зокрема, сучасні онлайн-бібліотеки, оснащені інтеграцією функцій соціальних мереж, відкривають нові можливості для взаємодії користувачів. Ці бібліотеки дозволяють обговорювати книги, залишати рецензії, ділитися рекомендаціями та створювати літературні спільноти. Така інтеграція не лише підвищує інтерес до читання, але й сприяє культурному розвитку суспільства, адже об'єднує людей за їхніми літературними уподобаннями.

Актуальність обраної теми зумовлена потребою в розвитку інноваційних рішень для покращення функціональності онлайн-бібліотек, що дозволяють користувачам не лише читати книги, а й активно брати участь у літературному житті. Інтеграція соціальних функцій є важливим кроком до створення унікального користувацького досвіду та сприяє популяризації електронних бібліотек як сучасних культурних центрів.

Окрім того, проблема персоналізації користувацького досвіду залишається однією з головних у розробці вебзастосунків. Розробка front-end частини такої онлайн-бібліотеки забезпечує не лише зручний і естетичний інтерфейс, але й максимальну інтерактивність та доступність функцій для кожного користувача.

Метою роботи є розробка front-end частини онлайн-бібліотеки з інтеграцією функцій соціальних мереж з метою популяризації електронних літературних ресурсів.

Для досягнення цієї мети було поставлено такі завдання:

- провести аналіз предметної області, зокрема існуючих рішень у сфері онлайн-бібліотек та соціальних платформ;
- здійснити розробку проєктних рішень застосунку, з урахуванням функціональних вимог до користувацького інтерфейсу;
- виконати моделювання та конструювання програмного забезпечення, у тому числі структури клієнтської частини;
- здійснити формування структури бази даних, орієнтуючись на функціональність системи;
- здійснити розробку інтерфейсу користувача та створення його прототипу з урахуванням інтеграції соціальних функцій;
- виконати розробку клієнтської частини вебзастосунку додатку з використанням обґрунтованих front-end технологій;
- оцінити ефективність запропонованого рішення та визначити напрямки його подальшого вдосконалення.

Об'єктом кваліфікаційної роботи є процеси пошуку, використання та розповсюдження електронних літературних творів.

Предметом кваліфікаційної роботи є інструменти створення front-end частини вебзастосунку онлайн-бібліотеки з інтеграцією функцій соціальних мереж.

Обґрунтування необхідності розробки полягає в тому, що існуючі рішення не завжди забезпечують достатній рівень взаємодії між користувачами та контентом. Багато існуючих онлайн-бібліотек, хоча і надають доступ до великих баз даних книг, обмежуються лише функціоналом перегляду і завантаження, не забезпечуючи повноцінної інтерактивної взаємодії (обговорень, рецензій, рекомендацій).

Незважаючи на численні існуючі платформи, є чітке розуміння, що потреба в створенні універсальної, інтерактивної онлайн-бібліотеки залишається. Зокрема, проєкт у галузі соціалізаційних функцій онлайн-бібліотек, таких як взаємодія через рецензії та рейтинги, є важливим кроком до подолання цих недоліків.

Сучасні тенденції в онлайн-культурі, зокрема популярність платформ типу Goodreads, показують зростаючий інтерес до соціальних функцій у сфері читання.

Ведучі технологічні компанії активно впроваджують механізми рекомендацій, щоб покращити користувацький досвід. Платформи такі як Audible (Amazon), Scribd та інші також почали активно впроваджувати елементи соціальних мереж, створюючи можливості для взаємодії між користувачами.

З огляду на це, існує потреба в удосконаленні існуючих рішень шляхом розробки нових функцій і можливостей для користувачів, включаючи інтеграцію соціальних функцій у онлайн-бібліотеки для створення більш динамічного і персоналізованого досвіду.

Основні проєктні рішення передбачають використання сучасних front-end технологій, зокрема React для створення динамічного інтерфейсу та інтеграція соціальних функцій через API для обговорень, рецензій і рекомендацій. Вибір таких технологій обґрунтований їхньою популярністю, масштабованістю та здатністю забезпечити високу продуктивність і інтерактивність інтерфейсу. Також важливим напрямком дослідження є вивчення існуючих рішень в галузі онлайн-бібліотек та соціальних платформ для виявлення найкращих практик інтеграції соціальних функцій, що покращують користувацький досвід.

Сферою застосування результатів є розвиток онлайн-бібліотек, інтерактивних платформ для читання та обміну літературними творами. Запропоноване рішення може бути використано для покращення функціональності існуючих електронних бібліотек, інтеграції соціальних платформ для літературних спільнот, а також у розробці мобільних застосунків та вебсайтів, що забезпечують доступ до електронних книг з інтерактивними функціями. Окрім того, результати можуть бути корисними для дослідження та впровадження інноваційних технологій у сфері освіти та культури.

1 АНАЛІТИЧНИЙ ОГЛЯД ЦИФРОВИХ ПЛАТФОРМ

1.1 Основні вимоги до цифрових платформ

У сучасних умовах стрімкого розвитку інформаційних технологій та активного переходу людства до цифрового середовища, особливого значення набуває питання адаптації традиційних форм культурної діяльності до нових цифрових форматів. Однією з таких форм є літературна творчість – одна з найдавніших сфер людської діяльності, яка зберігає свою актуальність і в XXI столітті. Проте, змінилися не лише носії інформації, а й підходи до створення, поширення та споживання літературного контенту.

З розвитком інтернету та мобільних технологій читання перестало асоціюватися виключно з друкованими книгами. Сьогодні користувачі мають змогу читати твори будь-де: на смартфоні, планшеті, ноутбучі чи навіть смарт-годиннику. Змінилася й культура споживання тексту – читачі прагнуть до інтерактивності, соціальної взаємодії, миттєвих рекомендацій і зворотного зв'язку з авторами.

У зв'язку з цим виникає потреба в розробці нових цифрових платформ, які б дозволяли авторам публікувати свої твори, читачам – комфортно читати й оцінювати контент, а спільнотам – формувати навколо літератури активне середовище. Ідеальним рішенням для реалізації таких потреб стає вебплатформа, що поєднує функціональність електронної бібліотеки, блогу, спільноти та системи взаємодії користувачів.

Цифрові рішення в галузі літератури охоплюють такі ключові аспекти:

- зручний доступ до літературного контенту в різних форматах (текст, аудіо, PDF);
- можливість швидкої навігації, пошуку та фільтрації;
- інтеграція з особистим кабінетом користувача, його читацькою історією;
- формування рейтингових систем творів на основі оцінок читачів;
- підтримка коментарів, обговорень і створення персональних добірок;

– надання авторам зручних інструментів для публікації, редагування і просування власних робіт.

Предметна область даної кваліфікаційної роботи охоплює як програмні, так і функціональні аспекти побудови вебзастосунку, орієнтованого на літературну взаємодію. Розробка подібної системи має враховувати потреби кількох ключових категорій користувачів:

- читачів, які прагнуть знайти цікавий контент, зберігати вподобані твори та брати участь в обговореннях;
- авторів, які мають змогу створювати персональні сторінки, викладати твори, відслідковувати реакцію аудиторії;
- адміністраторів, які підтримують якість контенту, модерують спільноти, керують доступом і функціоналом платформи.

Водночас, предметна область включає і низку сучасних викликів:

- забезпечення безпеки даних та авторських прав;
- реалізація адаптивного дизайну для широкого спектра пристроїв;
- оптимізація роботи з великим обсягом даних;
- підтримка швидкодії та масштабованості.

Сьогодні подібні рішення активно впроваджуються провідними платформами, однак більшість із них або обмежені функціонально, або створені для іншої цільової аудиторії (наприклад, Wattpad – для молодіжної фанфікшн-спільноти, Goodreads – для обговорення, але без самих текстів творів тощо). Таким чином, зберігається актуальна потреба у створенні спеціалізованої системи, що б одночасно відповідала сучасним вимогам до дизайну, функціоналу, соціальної взаємодії та масштабованості.

Особливістю запропонованої розробки є те, що вона поєднує кілька векторів:

- технологічний (використання сучасного стеку: React, ASP.NET, MySQL);
- культурний (популяризація творчості незалежних авторів);
- соціальний (створення онлайн-спільноти навколо літератури);

– освітній (можливість використання у навчальних цілях, для поширення наукової та публіцистичної літератури).

Також спостерігається явище так званої самвидавної культури (англ. *self-publishing*), коли автори обходяться без традиційних видавництв, викладаючи власні твори безпосередньо в мережу. Цей підхід дозволяє значно скоротити часові та фінансові витрати на публікацію, але при цьому створює нові виклики – зокрема, потребу в просуванні, захисті авторських прав та налагодженні взаємозв'язку з читачами.

Крім того, відсутність редакторської та дизайнерської підтримки змушує авторів самостійно опановувати інструменти для форматування текстів, створення обкладинок і навіть ведення аналітики аудиторії. У такому контексті особливої ваги набувають онлайн-платформи, які можуть забезпечити авторів необхідними засобами для повноцінної презентації своїх творів та формування власного читацького кола.

Загалом, предметна область дослідження має перетин із наступними напрямками:

- цифрове книговидання;
- веброзробка та UX/UI-дизайн;
- інформаційні системи та бази даних;
- соціальні мережі та взаємодія в онлайн-спільнотах.

Таким чином, платформа для публікації й читання творів повинна поєднувати в собі функціонал класичної електронної бібліотеки, елементи соціальної мережі, а також інструменти для редагування та форматування літературного контенту. З точки зору технічної реалізації, подібні сервіси вимагають ретельного підходу до вибору архітектури, мов програмування, фреймворків, систем управління базами даних, а також рішень щодо безпеки, масштабування та SEO-оптимізації.

Варто зазначити, що більшість популярних сервісів, таких як Wattpad, Royal Road, Inkitt, Medium, орієнтовані або на вузький сегмент аудиторії, або мають значні обмеження щодо кастомізації, монетизації та локалізації. Крім того, далеко

1.2 Компоненти платформи літературних творів

У сучасному цифровому суспільстві дедалі більшої популярності набувають онлайн-платформи, які дозволяють користувачам створювати, зберігати, обмінюватися та поширювати інформаційний контент різного типу. Одним із актуальних напрямів є створення веб-платформ, орієнтованих на літературний контент, що дає змогу авторам представляти свої твори широкому загалу, а читачам – відкривати для себе нових митців, взаємодіяти з ними, залишати відгуки та брати участь у формуванні рейтингу творів. Вивчення структурних і функціональних особливостей такого роду систем є надзвичайно важливим кроком при плануванні та проєктуванні повноцінного та ефективного програмного продукту.

У рамках розробки вебплатформи для публікації та читання літературних творів необхідно детально проаналізувати її структурні та функціональні компоненти. Це дозволяє не лише систематизувати вимоги до майбутнього програмного продукту, а й спрогнозувати шляхи його подальшого масштабування й підтримки.

Платформа, що є об'єктом дослідження, має чітко визначену структурну модель, яка дозволяє забезпечити гнучку і масштабовану архітектуру. Зазвичай у подібних системах реалізується поділ на три головні компоненти: інтерфейс користувача (frontend), серверна логіка (backend) та база даних (storage). Кожен із цих компонентів виконує свою специфічну функцію та тісно взаємодіє з іншими, забезпечуючи цілісність та узгодженість процесів.

З точки зору функціонального наповнення, така система передбачає наявність широкого спектра можливостей. Серед основних функціональних блоків можна виділити: систему реєстрації та аутентифікації користувачів, можливість завантаження та збереження літературних творів, організацію бібліотеки користувача, реалізацію механізмів рейтингу, коментування, пошуку за жанрами, авторами, ключовими словами тощо. Значну увагу приділяють також питанням

З метою більш наочного уявлення про структуру та функціональність об'єкта дослідження, нижче наведено узагальнену таблицю, яка ілюструє основні структурні компоненти та їхні функції.

Таблиця 1.1 – Структурні компоненти об'єкта дослідження та їх функціональне призначення

№	Компонент системи	Опис функціонального призначення
1	Інтерфейс користувача	Забезпечує графічну взаємодію користувача з системою. Включає сторінки реєстрації, входу, перегляду творів, профілю автора, тощо.
2	Серверна логіка	Обробляє запити від користувачів, реалізує бізнес-логіку системи, зокрема додавання, редагування творів, пошук, фільтрацію.
3	База даних	Зберігає всю структуровану інформацію про користувачів, твори, рейтинги, коментарі, категорії тощо.
4	Адміністративна панель	Дозволяє адміністраторам керувати вмістом платформи, модерувати відгуки, керувати користувачами.
5	API-шар	Відповідає за обмін даними між frontend і backend, а також можливість інтеграції з мобільними застосунками або зовнішніми сервісами.

Кожен із вказаних компонентів не лише виконує певну роль, але й утворює із іншими взаємопов'язану систему, що забезпечує ефективне функціонування платформи. Наприклад, коли користувач переглядає сторінку певного літературного твору, інтерфейс надсилає запит на сервер, де відповідний обробник

отримує дані з бази даних, обробляє їх і передає назад у вигляді зручного формату (наприклад, JSON), що відображається у браузері користувача.

Окрему увагу в контексті структурного аналізу варто приділити способу організації бази даних. Як правило, використовується реляційна модель, у якій основними сутностями є: користувачі, твори, автори, жанри, рецензії, оцінки, історія переглядів, коментарі. Залежно від обраної СКБД (системи керування базами даних) – зокрема, MSSQL, PostgreSQL чи MySQL – структура БД може мати певні особливості реалізації, однак загальна модель залишається схожою.

У підсумку, об'єкт дослідження можна охарактеризувати як складну, багаторівневу програмну систему, побудовану за принципами клієнт-серверної архітектури, з чітко визначеними структурними та функціональними модулями. Така модель дозволяє легко масштабувати систему, розширювати функціонал, оптимізувати взаємодію між її частинами та забезпечити якісний досвід користувача.

У контексті дослідження об'єкта як користувацького інтерфейсу веб-платформи важливим аспектом є побудова фронтенд-частини, яка слугує головним каналом взаємодії між користувачем і системою. У сучасному розробленні інтерфейсів особлива увага приділяється не лише функціональності, а й естетичному оформленню, доступності, логічності структури й адаптивності до різних пристроїв.

Фронтенд, або клієнтська частина, як правило, створюється за допомогою таких технологій, як HTML, CSS, JavaScript, а також сучасних фреймворків і бібліотек, зокрема React, Vue.js, Angular тощо. У випадку об'єкта дослідження – веб-платформи для літературних творів – ключовим є забезпечення інтуїтивного та привабливого інтерфейсу, що враховує як потреби авторів, так і читачів.

Розроблення інтерфейсу (табл. 1.2) повинно здійснюватися з урахуванням принципів UX/UI-дизайну, які передбачають логічне групування елементів, зручну навігацію, наявність візуальних підказок, відповідність кольорів і шрифтів до тематики проєкту. Важливо забезпечити достатній рівень контрастності для тексту, видимість активних елементів, підтримку темної теми та варіанти налаштувань

інтерфейсу, включаючи зміну шрифтів, розміру тексту, кольорових схем та адаптацію для людей із порушеннями зору чи моторики що дозволяє зробити взаємодію з платформою інклюзивною та зручною для користувачів.

Невід'ємною частиною фронтенд-розробки є адаптивна верстка, яка дозволяє інтерфейсу коректно відображатися на екранах з різними роздільними здатностями – від мобільних телефонів до великих моніторів. Для цього використовуються медіазапити (media queries), відносні одиниці виміру, гнучкі сітки (grid, flexbox), а також підхід mobile-first, який передбачає спершу розроблення дизайну для мобільних пристроїв із подальшою адаптацією для десктопів.

Таблиця 1.2 – Основні елементи сучасного фронтенду для літературної платформи

№	Компонент інтерфейсу	Функціональне призначення
1	Шапка (Header)	Містить логотип, меню навігації (Головна, Книги, Автори, Блог, Вхід/Реєстрація, Кабінет), пошук.
2	Головна сторінка	Демонструє популярні твори, нові надходження, банери, відгуки, новини платформи.
3	Сторінка книги	Відображає обкладинку, назву, автора, жанр, вік читача, рейтинг, опис, кнопку «Читати» та коментарі.
4	Сторінка автора	Інформація про автора, його твори, біографія, фото, кількість переглядів, підписка на автора.
5	Адаптивне меню	Згортається у мобільному вигляді, доступне через бургер-іконку, дозволяє швидкий доступ до розділів.
6	Секція відгуків	Дає змогу залишити оцінку, написати відгук, поставити «лайк» іншим відгукам.
7	Кабінет користувача	Містить інформацію про обліковий запис, бібліотеку прочитаного, чернетки, список збереженого.

Адаптивність є важливою не тільки з технічної точки зору, але й з погляду доступності. Усе більше користувачів взаємодіють із платформами не лише з ПК, а й зі смартфонів, планшетів, телевізорів тощо. Таким чином, адаптивний інтерфейс є не розкішшю, а необхідністю. Недостатня адаптивність призводить до зменшення часу перебування на сайті, підвищення показника відмов і, зрештою, втрати довіри до ресурсу.

Крім того, інтерфейс має відповідати принципам зрозумілості та єдності стилю. Це означає, що всі елементи дизайну повинні дотримуватись спільної візуальної мови: однакові шрифти, палітра кольорів, відступи, іконки та анімації. Застосування дизайн-системи або UI-китів значно спрощує підтримку такого стилю та робить розробку швидшою.

У сучасних умовах дизайн також має враховувати тенденції мінімалізму, soft UI або neumorphism, що дозволяють створити приємний і не перевантажений інтерфейс. Особливо важливо враховувати відгук системи (система повинна реагувати на дії користувача – підсвічування кнопок, анімації, сповіщення), що покращує користувацький досвід і підвищує залученість.

Таким чином, аналіз функціональних та структурних особливостей об'єкта дослідження з точки зору фронтенду демонструє складність та багатогранність підходів до його реалізації. Грамотно побудований інтерфейс не лише полегшує користувачам взаємодію з системою, а й визначає загальне враження про якість розробки, забезпечуючи довіру та лояльність аудиторії.

1.3 Огляд існуючих аналогів

На сьогоднішній день ринок програмних рішень для публікації та обміну літературним контентом демонструє значну динаміку розвитку. Особливої популярності набувають онлайн-платформи, які дозволяють авторам безпосередньо взаємодіяти з аудиторією, минаючи класичні видавничі інституції. Такий підхід не лише спрощує процес публікації, але й трансформує саму модель функціонування літературного середовища.

Серед найвідоміших прикладів подібних платформ варто виокремити такі як Wattpad, Royal Road, Archive of Our Own (AO3), Tapas, Scribble Hub, а також деякі локальні проекти, наприклад Букнет чи Літнет. Кожна з них має свої унікальні функціональні особливості, цільову аудиторію, підхід до взаємодії між авторами та читачами, а також власну філософію розвитку контенту. Нижче наведено детальніший аналіз цих платформ.

Wattpad – один із наймасштабніших сервісів для самвидаву, створений у Канаді у 2006 році. Наразі платформа налічує понад 90 мільйонів користувачів щомісяця. Основною метою Wattpad є демократизація літературного процесу – кожен може публікувати власні твори, не маючи жодного досвіду чи зв'язків у видавничій сфері.

Інтерфейс сервісу інтуїтивно зрозумілий: автор може створювати розділи, додавати обкладинки, інтегрувати мультимедійні елементи (наприклад, зображення або GIF-файли). Читачі мають можливість не лише ставити вподобання, а й коментувати кожен абзац окремо, що створює ефект живого спілкування між автором та аудиторією.

Wattpad (рис. 1.1) також має власну видавничу ініціативу Wattpad Books, завдяки якій найпопулярніші автори можуть укладати контракти з видавцями. Крім того, існує програма Wattpad Paid Stories, яка дозволяє авторам заробляти на своїх творах. Проте участь у ній обмежена: автор має відповідати вимогам щодо активності, якості текстів та кількості читачів.



Рисунок 1.1 – Wattpad

Платформа орієнтована переважно на молодіжну аудиторію (віком 13–25 років), яка цікавиться фанфіками, романтичними історіями, драмою та підлітковою

фантастикою. Основна частина контенту – неформальна література з активним діалогом між автором та читачем.

Royal Road – платформа, заснована з фокусом на жанри фентезі, наукової фантастики, LitRPG, cultivation та isekai. На відміну від Wattpad, Royal Road орієнтована на більш серйозну та вимогливу аудиторію, яка шукає великі об'єми якісно написаного тексту з регулярними оновленнями. Особливістю платформи є чітка структура організації творів – автор може позначити статус (ongoing, hiatus, completed), налаштувати доступність глав, створити індекс або словник термінів. Читачі можуть залишати відгуки, додавати твори до бібліотеки, оцінювати історії за шкалою, а також слідкувати за оновленнями.

Платформа також пропонує авторам доступ до аналітики переглядів, показників утримання читачів, кількості унікальних відвідувачів тощо. Однак монетизація реалізована через зовнішні сервіси – автор може вказати посилання на Patreon, Ko-fi або інші платформи підтримки.

У плані дизайну Royal Road виглядає досить просто, проте функціонально. Основна аудиторія – дорослі читачі, що надають перевагу серіальним творам з глибокою міфологією та world-building.

АОЗ (Archive of Our Own) – одна з найбільш знакових платформ для фанфікшну, створена зусиллями волонтерської організації Organization for Transformative Works (OTW). АОЗ є некомерційним ресурсом, який існує виключно завдяки підтримці спільноти, а весь код системи знаходиться у відкритому доступі.

Головна цінність АОЗ полягає у відсутності обмежень: автор може публікувати будь-який вміст (за умови, що він не порушує закон), а читачі можуть фільтрувати твори за тегами, попередженнями (наприклад, «смерть головного героя», «жорстокість») та жанрами.

Функціонал коментування, підписки на авторів, створення власних колекцій, створення тегів та пошук за складними запитамі – усе це дозволяє глибоко персоналізувати взаємодію з контентом. АОЗ не містить реклами, не має платного доступу й не передбачає монетизації творів, що робить її унікальною у своєму роді.

Платформа є прикладом цифрового гуманітарного проєкту, що орієнтований на захист творчої свободи та авторських прав.

Taras – платформа, що поєднує функції публікації вебкоміксів та серіальних новел. Вона має привабливий інтерфейс, мобільний застосунок та інтегровану систему монетизації.

Taras працює за моделлю «freemium»: частина контенту доступна безкоштовно, а за доступ до наступних розділів користувачі можуть платити за допомогою монет (Taras Coins), які купуються або заробляються під час активності.

Платформа також підтримує систему донатів, що дозволяє авторам заробляти навіть без прямого продажу контенту. Дизайн сервісу орієнтований на молодіжну аудиторію, а значна частина творів створена у форматі «light novel» або манхви.

Scribble Hub – менш відомий, але популярний серед фанатів аматорської літератури ресурс, який нагадує Royal Road, але є відкритішим до неформатного контенту, романтичної літератури та жанрів з ознаками еротики. Особливістю є дуже гнучка система тегів, що дозволяє точно фільтрувати контент за найрізноманітнішими параметрами. Scribble Hub також дозволяє налаштовувати зовнішній вигляд сторінок (наприклад, змінювати кольорову схему), а автори мають змогу вести блоги та залишати повідомлення своїм підписникам.

Букнет / Літнет – це локальні україномовні (і російськомовні) сервіси, що працюють за моделлю часткового платного доступу до контенту. Автори можуть публікувати глави, а після набору певної кількості читачів — переводити твір на комерційний статус.

Букнет активно розвиває систему рейтингів, конкурсів для авторів, співпрацю з видавництвами. Інтерфейс платформи зручний для мобільних пристроїв, підтримує темну тему та можливість створювати добірки. Значну частину контенту складає романтична проза, детективи, підліткова література. Платформа має певні елементи соціальної мережі, зокрема сповіщення, лайки, стіну автора та підписки.

Ці платформи демонструють широкий спектр підходів до організації літературного контенту – від повністю відкритих і некомерційних до інтегрованих у бізнес-моделі видавництва. Різноманіття функціоналу, рівнів взаємодії та моделей монетизації дає підстави вважати, що існує запит на нові рішення, які можуть поєднувати зручність, гнучкість, сучасний дизайн та локалізований підхід до української аудиторії.

Для порівняння основних функціональних характеристик найпопулярніших платформ подано таблицю нижче.

Таблиця 1.3 – Порівняння функціональних можливостей онлайн-платформ для публікації літератури

Платформа	Коментарі до розділів	Темна тема	Монетизація	Пошук за тегами	Мобільний застосунок
Wattpad	Так	Так	Так	Так	Так
Royal Road	Частково	Ні	Ні	Так	Ні
АОЗ	Так	Частково	Ні	Так	Так (через браузер)
Букнет/Літнет	Так	Так	Так	Частково	Так
Tapas	Так	Так	Так	Так	Так

На основі аналізу вищенаведених платформ можна зробити висновок, що ключовими трендами у розвитку подібних сервісів є:

- фокус на мобільність – більшість користувачів читають зі смартфонів, тому адаптивний дизайн або окремий мобільний застосунок є обов’язковими;
- темна тема – підтримка різних варіантів відображення інтерфейсу, включаючи темний режим, стає стандартом;
- інтерактивність – можливість коментування текстів, участі в обговореннях, оцінювання творів;
- алгоритмічне просування – системи рекомендацій на основі уподобань користувача;

– монетизація – можливість авторів заробляти на своїх творах (платні глави, донати, підписки).

Проте жодна з проаналізованих платформ не є досконалою. Часто вони не дозволяють повною мірою кастомізувати зовнішній вигляд сторінок, мають застарілий інтерфейс або перенасичені рекламними блоками. З огляду на це, створення нової платформи з орієнтацією на зручність UI/UX, адаптивність, мінімалізм, а також можливість персоналізації виглядає логічним і доцільним.

Також варто зазначити, що на ринку не вистачає локалізованих рішень з підтримкою української мови, що особливо актуально в контексті культурного та інформаційного суверенітету. Більшість існуючих сервісів орієнтовані на глобальну аудиторію, в той час як українські автори часто змушені публікувати свої твори на неадаптованих для них платформах.

З точки зору технологічної реалізації, сучасні фронтенд-платформи найчастіше використовують фреймворки React, Vue.js, або Angular, що забезпечують високу продуктивність, гнучкість у реалізації інтерфейсу, а також зручну інтеграцію з backend-сервісами через REST API або GraphQL. Такі фреймворки дозволяють реалізовувати SPA (Single Page Applications), які значно покращують користувацький досвід.

Переважає більшість проєктів застосовує компонентний підхід до побудови інтерфейсів, що дозволяє легко масштабувати проєкт, підтримувати код і перерозподіляти відповідальність між членами команди. Особливу увагу також приділяють забезпеченню адаптивності, адже понад 60% користувачів споживають контент через мобільні пристрої [4].

Таким чином, з огляду на вивчений досвід, є всі передумови для розробки інноваційного інтерфейсного рішення, яке базуватиметься на сучасних вебтехнологіях, враховуватиме потреби користувачів та забезпечуватиме високий рівень доступності, адаптивності та персоналізації.

Висновки до розділу 1

У першому розділі було здійснено системний аналіз обраної предметної області, яка стосується розробки платформ для публікації та обміну літературними творами користувачів.

Було розкрито об'єкт і предмет дослідження: об'єктом виступають онлайн-платформи для самостійної публікації літературних творів, тоді як предметом — процеси їхнього проєктування, розробки та функціонування з урахуванням вимог сучасних користувачів.

Проведено опис і аналіз структурних та функціональних особливостей таких платформ, визначено їх основні компоненти: система управління контентом, механізми взаємодії авторів і читачів, сервіси монетизації, модерації та аналітики. Розглянуто, як ці компоненти реалізуються у межах відомих прикладів — Wattpad, Royal Road, Archive of Our Own (AO3), Tapas, Scribble Hub, Букнет і Літнет.

У ході огляду та аналізу сучасного стану програмних рішень у даній сфері було встановлено, що більшість популярних платформ орієнтуються на вузьку аудиторію за жанровими або віковими ознаками, мають розвинені інструменти для підтримки спільноти та стимулювання творчості, а також пропонують різні можливості монетизації контенту. Водночас спостерігаються недоліки у вигляді обмеженої свободи публікації, складних правил монетизації або високого порогу входу для нових авторів.

На основі проведеного аналізу було сформульовано постановку завдань для подальших етапів кваліфікаційної роботи, яка полягає у розробці власного веб-ресурсу, що поєднуватиме кращі функціональні рішення існуючих платформ та враховуватиме їхні недоліки.

2 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір технологій та інструментів для розробки фронтенду

Розробка сучасних вебзастосунків вимагає ретельного вибору технологій, які забезпечують зручність розробки, масштабованість, швидкодію та зручний інтерфейс для користувача. Клієнтська частина, або фронтенд, безпосередньо взаємодіє з користувачем і відповідає за відображення вмісту, навігацію по сторінках, взаємодію з формами, сповіщеннями та іншими інтерактивними елементами. Тому саме на етапі планування фронтенду потрібно обрати технології, які забезпечать зручну розробку і якісний користувацький досвід.

Перед тим, як перейти до вибору конкретних інструментів, були сформульовані основні вимоги до майбутньої системи. Вони включають можливість створення односторінкового застосунку, адаптивність до різних пристроїв, підтримку компонентного підходу до побудови інтерфейсу, високу продуктивність, наявність централізованого керування станом, а також простоту інтеграції з серверною частиною через REST API [16]. Крім того, важливим фактором є наявність широкої спільноти та документації, що спрощує підтримку та розвиток проєкту.

Для реалізації клієнтської частини було розглянуто три популярні технології: React.js, Vue.js та Angular. Кожна з них має свої переваги й недоліки (табл. 2.1). Для зручності порівняння основних характеристик цих фреймворків/бібліотек було складено таблицю.

Аналіз показав, що React.js (рис. 2.1) має найбільш гнучку структуру, високу швидкість рендерингу завдяки віртуальному DOM і потужну екосистему.

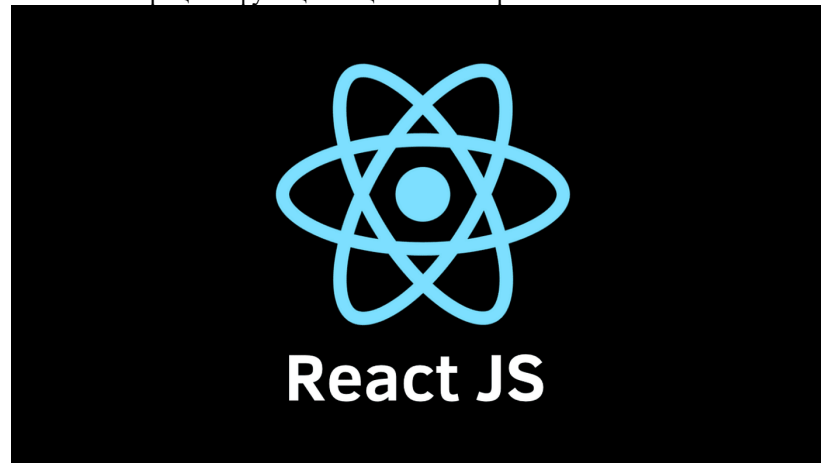


Рисунок 2.1 – React.js

Vue.js (рис. 2.2) вирізняється простотою у вивченні та має зрозумілий синтаксис, що особливо зручно для новачків.



Рисунок 2.2 – Vue.js

Angular (рис. 2.3) же надає повноцінний фреймворк із вбудованою системою залежностей, модулями та розширеною функціональністю, проте є більш складним у вивченні та використанні.



Рисунок 2.3 – Angular

Враховуючи вимоги до масштабованості, продуктивності та популярності серед розробників, було прийнято рішення використовувати React.js як основу для клієнтської частини системи.

Таблиця 2.1 – Порівняння популярних фреймворків для frontend-розробки

Критерій	React.js	Vue.js	Angular
Тип	Бібліотека	Фреймворк	Повноцінний фреймворк
Продуктивність	Висока	Висока	Висока
Крива навчання	Помірна	Низька	Висока
Компонентний підхід	Так	Так	Так
Ширина спільноти	Дуже широка	Широка	Середня
Гнучкість	Висока	Висока	Середня
Ідеально підходить для	SPA, UI-компонентів	SPA, UI	Складних корпоративних застосунків

Після детального аналізу було прийнято рішення використовувати саме React.js як основну бібліотеку для побудови клієнтської частини. Вона є універсальною, має величезну спільноту, гнучку архітектуру та дозволяє створювати як прості компоненти, так і складні застосунки. Крім того, React

підтримується компанією Meta (раніше Facebook), що забезпечує її актуальність та постійний розвиток.

Для організації глобального стану в застосунку було вирішено використати Redux Toolkit. Це сучасне рішення, яке суттєво спрощує використання Redux, дозволяє уникнути зайвого шаблонного коду і забезпечує зрозумілу логіку керування даними. Такий підхід є надзвичайно корисним для масштабних проєктів, де дані використовуються одночасно в багатьох компонентах, як-от інформація про користувача, перелік книг, улюблене, налаштування тощо.

Для стилізації інтерфейсу обрано Bootstrap (рис. 2.4) – один з найпопулярніших CSS-фреймворків для створення адаптивного та зручного інтерфейсу. Bootstrap надає велику кількість готових компонентів (кнопки, форми, навігація, модальні вікна тощо), що дозволяє швидко реалізовувати функціональний та привабливий дизайн без потреби писати стилі з нуля. Його система сітки забезпечує гнучке компонування елементів для різних розмірів екранів, що сприяє кращій адаптивності. Завдяки широкому поширенню та активній спільноті Bootstrap значно спрощує процес розробки та підтримки інтерфейсу.



Рисунок 2.4 – Bootstrap

Також важливою частиною SPA-застосунків є організація маршрутизації. Для цього буде використано бібліотеку React Router (рис. 2.5). Вона забезпечує ефективну побудову внутрішньої навігації між сторінками без перезавантаження

браузера, підтримує динамічні маршрути, вкладені сторінки та передавання параметрів через URL.

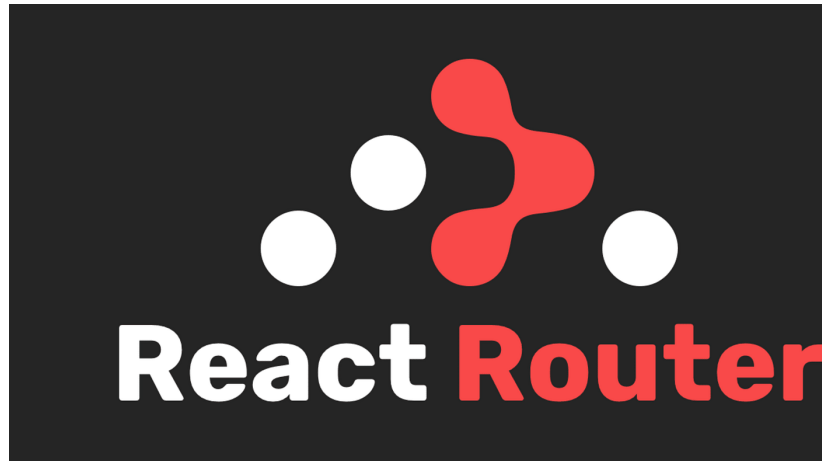


Рисунок 2.5 – React Router

Усі залежності та бібліотеки будуть керуватись за допомогою Node.js та менеджера пакетів npm. Як середовище розробки використовується Visual Studio Code – популярний редактор з великою кількістю розширень для роботи з React, форматування коду, автоматичного виправлення помилок тощо. Для керування версіями коду та співпраці над проектом використовуватиметься система контролю версій Git, а також GitHub (рис. 2.6) для збереження віддаленого репозиторію та розгортання демо-версії через GitHub Pages.



Рисунок 2.6 – GitHub

Таким чином, обраний набір технологій повністю відповідає поставленим вимогам і дозволяє створити функціональний, масштабований та естетично привабливий клієнтський інтерфейс. Вибір React.js як основної бібліотеки, у поєднанні з Redux Toolkit, Bootstrap та React Router, є оптимальним рішенням для реалізації проєкту сучасного онлайн-сервісу.

2.2 Формування специфікації вимог до програмного забезпечення (SRS)

У цьому розділі визначається повний набір вимог до програмного забезпечення Manuscript – вебплатформи для публікації, збереження, перегляду та обговорення літературних творів. Специфікація вимог (англ. Software Requirements Specification, SRS) є ключовим документом, що формалізує домовленості між замовником і виконавцем. Вона містить опис функціональності, структури, характеристик користувачів, обмежень і вимог до якості програмного продукту. SRS є основою для розробки, тестування і подальшого супроводу ПЗ.

Призначення програмного забезпечення

Платформа Manuscript створюється з метою об'єднання авторів, перекладачів та читачів у єдиному цифровому просторі. Користувачі зможуть розміщувати свої літературні твори, читати книги інших авторів, залишати рецензії, додавати твори до своєї бібліотеки та ділитися вподобаними роботами. Система підтримуватиме текстові формати творів (оповідання, романи, вірші), а також забезпечуватиме категоризацію за жанром, віковим обмеженням та мовою.

Межі проєкту

Проєкт обмежується створенням вебінтерфейсу платформи, без розробки мобільного застосунку. Платформа не підтримує монетизацію, рекламу чи платний контент. Основна увага приділяється публікації та читанню безкоштовних творів. Редактор для публікації текстів передбачає лише базове форматування (заголовки, абзаци, курсив, жирний текст). Не передбачається автоматичне вичитування, AI-коректура чи інтеграція з офісними програмами.

Характеристики користувачів

Цільова аудиторія платформи – це користувачі віком від 16 років, які мають базові навички роботи з вебзастосунками. Серед них – аматори та професійні автори, читачі художньої літератури, перекладачі, редактори, а також літературні критики. Очікується, що користувачі зможуть легко орієнтуватися у структурі сайту, здійснювати пошук за категоріями, оцінювати та коментувати твори.

Функціональні вимоги

До функціональних вимог належать такі можливості:

- реєстрація та авторизація користувачів (у тому числі через Google-акаунт);
- створення особистого профілю, завантаження аватара;
- публікація власних творів з можливістю редагування та видалення;
- читання творів інших користувачів, з можливістю перемикання на темний/світлий режим;
- додавання творів до "Улюблених", "Читати пізніше", "Моя бібліотека";
- залишення відгуків, оцінок, кількості переглядів;
- фільтрація та сортування творів за жанром, віковою категорією, популярністю;
- перегляд нових надходжень, рекомендованих та популярних книг.

Нефункціональні вимоги

До нефункціональних вимог відносяться:

- доступність: система має бути доступною 24/7 з мінімальним часом простою;
- продуктивність: середній час завантаження сторінки не повинен перевищувати 2 секунд;
- безпека: усі з'єднання мають відбуватись через захищений HTTPS-протокол; паролі зберігаються у зашифрованому вигляді;
- масштабованість: система повинна дозволяти розширення функціоналу без повної перебудови архітектури;
- адаптивність: інтерфейс має коректно відображатися на пристроях з різною роздільною здатністю.

Вимоги до інтерфейсу

Головний інтерфейс має бути інтуїтивно зрозумілим. Основні елементи:

- верхнє меню з навігацією (Головна, Книги, Автори, Моя бібліотека, Вхід/Реєстрація);
- блок популярних книг з обкладинками, назвами, авторами та рейтингом;
- фільтр категорій, жанрів, вікових обмежень;
- перегляд книг у вигляді списку або сітки;
- відображення обкладинки, рейтингу, кнопки "Читати", "Додати до бібліотеки".

Інформаційне забезпечення

Система зберігає такі типи даних:

- текстові твори (форматований текст);
- обкладинки (у форматі .jpg/.png);
- профілі користувачів (ім'я, біо, фото);
- рецензії, рейтинги, кількість переглядів;
- жанри, теги, вікові обмеження.

Уся інформація зберігається у структурованій базі даних Microsoft SQL Server (MSSQL), з можливістю резервного копіювання.

Програмне та технічне забезпечення

Платформа Manuscript розробляється із застосуванням таких технологій:

- **frontend:** React, Bootstrap;
- **backend:** ASP.NET Core Web API;
- **база даних:** MySQL;
- **контейнери:** Docker;
- **середовище розробки:** Visual Studio, GitHub.

Платформа працює у середовищі Docker-контейнерів із налаштованими мережевими та базовими безпековими параметрами.

Вимоги до зовнішніх інтерфейсів

Система використовує REST API для взаємодії між фронтом і бекендом. Підтримується аутентифікація через OAuth (Google). Зовнішні інтеграції плануються лише для авторизації. Комунікація здійснюється через HTTPS, стандарт JSON використовується для обміну даними між клієнтом і сервером.

Властивості програмного забезпечення

- **доступність:** висока (мінімальний простій, хостинг із резервуванням);
- **супроводжуваність:** весь код підтримується системою контролю версій (GitHub);
- **переносимість:** завдяки використанню контейнеризації (Docker) систему можна розгорнути на будь-якому сервері;
- **продуктивність:** швидке реагування, кешування популярних сторінок;
- **надійність:** обробка помилок, валідація введення, логування;
- **безпека:** захист від XSS, CSRF, SQL-ін'єкцій, SSL, хешування паролів.

Висновки до розділу 2

У другому розділі було здійснено ґрунтовний аналіз технічних та функціональних аспектів проєкту Manuscript. Було обґрунтовано вибір технологій для розробки клієнтської частини застосунку, зокрема використання HTML5, CSS3, JavaScript та фреймворку Bootstrap, що дозволяє створити адаптивний, сучасний та зручний інтерфейс користувача.

Також сформовано специфікацію вимог до програмного забезпечення у вигляді документа SRS (Software Requirements Specification), де деталізовано загальні характеристики системи, функціональні та нефункціональні вимоги, інтерфейси та системні обмеження.

У результаті виконаної роботи сформовано чітке технічне бачення проєкту, яке забезпечує основу для подальшого етапу – безпосередньої реалізації програмного забезпечення відповідно до визначених вимог.

3 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Побудова UML-моделей

У даному підрозділі розглянуто основні функції фронтенд-частини програмного забезпечення, що забезпечують повноцінну взаємодію користувача із системою. Інтерфейс користувача розроблено таким чином, щоб забезпечити зручний доступ до контенту, можливість персоналізації, а також ефективну навігацію по всіх основних розділах сайту. Користувач має можливість зареєструватися або увійти до системи.

Користувач має можливість створити новий акаунт або увійти до вже існуючого. Після автентифікації дані користувача зберігаються у localStorage, що дозволяє підтримувати сесію на всіх сторінках сайту без потреби повторного входу. Це забезпечує безперервність користування функціоналом платформи.

Головна сторінка та окрема сторінка "Книги" містять каталог літератури, який можна сортувати за жанрами, тегами, віковими обмеженнями. Передбачено також текстовий пошук за назвою книги. Кожна книга має окрему сторінку з розгорнутою інформацією: описом, обкладинкою, автором, рейтингом та коментарями користувачів.

Зареєстровані користувачі мають розширений функціонал: можливість залишати коментарі до книг, ставити лайки та дизлайки, додавати книги у свою бібліотеку (закладки). У профілі користувача відображаються його особисті дані, перелік збережених книг, активність, залишені коментарі та вподобані записи.

Окрім того, реалізовано стрічку новин, блогів і подій, яку можуть переглядати як гості, так і зареєстровані користувачі. Цей розділ містить інформацію про нові книги, активність авторів, оновлення сайту та тематичні статті.

Логіка взаємодії передбачає два основні режими роботи: як гість або як зареєстрований користувач. Гість має змогу переглядати каталог, новини та блоги, проте не може залишати коментарі, додавати книги до бібліотеки або взаємодіяти

зі сторінками профілю. Зареєстрований користувач, у свою чергу, має повний доступ до функціоналу: авторизація, збереження обраного контенту, коментування, персоналізація профілю, перегляд активності та керування налаштуваннями.

Основна взаємодія з інтерфейсом здійснюється через кліки по кнопках, перемикачах, фільтрах і посиланнях, а також за допомогою введення тексту у формах (реєстрація, логін, пошук, коментарі). Такий підхід дозволяє забезпечити інтуїтивно зрозумілу навігацію та ефективне використання ресурсу як для нових, так і для досвідчених користувачів.

Для візуалізації структури та взаємозв'язків компонентів вебзастосунку використано стандартні засоби об'єктно-орієнтованого моделювання на основі UML-нотації.

Система передбачає два основні типи користувачів:

- гість – неавторизований користувач;
- зареєстрований користувач – авторизований користувач із розширеними правами доступу.

Взаємодія з інтерфейсом реалізована за допомогою кліків по елементах управління (кнопки, посилання, перемикачі), а також шляхом введення тексту у відповідні форми (пошук, реєстрація, коментування).

Для формалізації структури застосунку побудовано такі діаграми:

Діаграма класів відображає основні сутності, що використовуються у програмному забезпеченні, їх атрибути, методи та зв'язки між ними. Це дозволяє описати модель предметної області на рівні логіки бізнес-об'єктів.

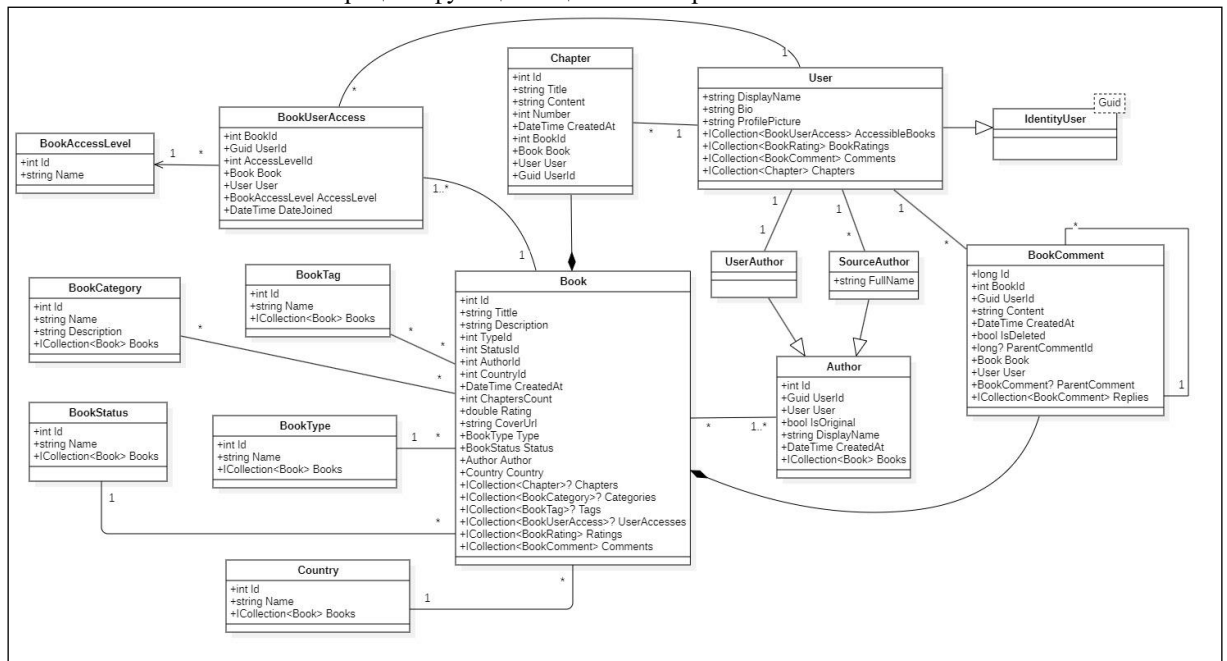


Рисунок 3.1 – Діаграма класів

На наступній діаграмі зображено процес взаємодії між користувачем та підсистемами електронної бібліотеки під час відкриття книги для читання. Учасниками даного процесу є: користувач (User), центральна система бібліотеки (LibrarySystem), каталог книг (BookCatalog) та застосунок для читання (ReaderApp). Процес починається з того, що користувач відкриває бібліотеку, ініціюючи виклик методу `OpenLibrary()`. У відповідь на це бібліотечна система звертається до каталогу книг із запитом `GetBookList()` для отримання повного списку доступних книг. Каталог повертає список, який далі передається користувачу у вигляді візуалізованого переліку книг (`ShowAvailableBooks()`). Користувач обирає потрібну книгу за допомогою виклику `SelectBook()`, після чого бібліотечна система надсилає запит до каталогу на отримання детальної інформації про книгу (`GetBookInfo()`). У відповідь каталог повертає відповідні дані, які потім демонструються користувачу (`BookInfo`). Коли користувач вирішує відкрити книгу, викликається метод `OpenBook()`, що ініціює завантаження вмісту книги в застосунок для читання (`LoadContent()`). Після передачі даних книги (`SendBookData()`), система запускає процес візуалізації книги в `ReaderApp` за допомогою методу `RenderBook()`. Таким чином, діаграма ілюструє послідовний

потік запитів і відповідей між компонентами системи, який забезпечує користувачу зручний доступ до електронних книг.

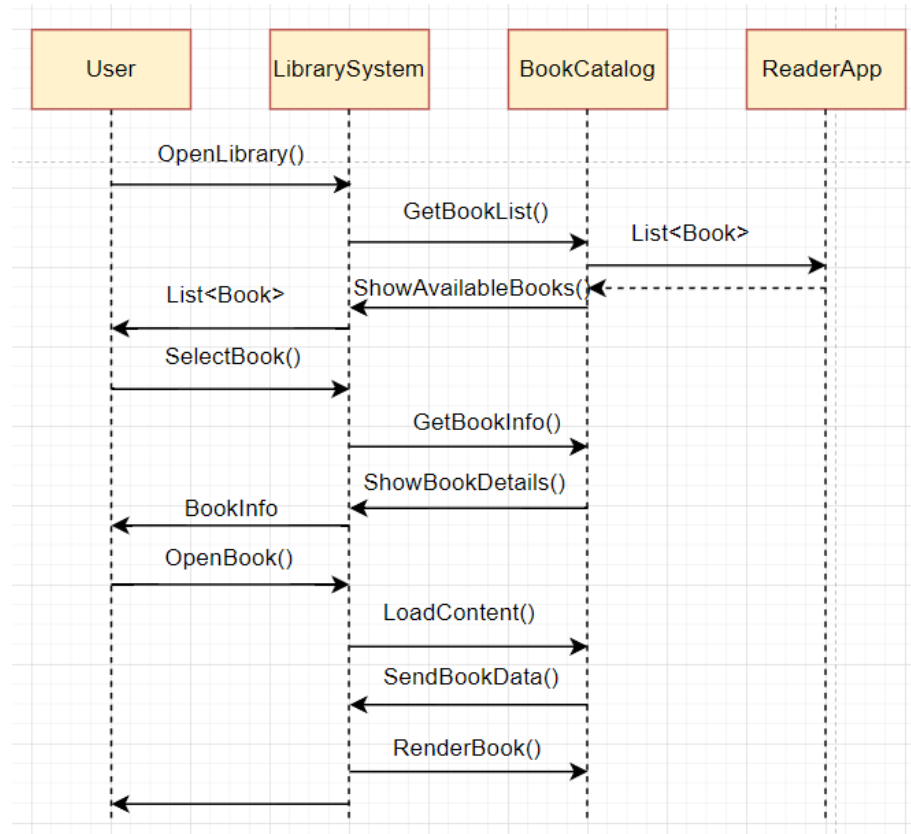


Рисунок 3.2 – Діаграма взаємодії «Читання книги онлайн»

Діаграма послідовностей демонструє процес відкриття книги користувачем у системі електронної бібліотеки. Користувач ініціює відкриття бібліотеки викликом `OpenLibrary()`, після чого система `LibrarySystem` запитує список книг у `BookCatalog` через `GetBookList()`. Отриманий список повертається користувачу. Далі користувач вибирає книгу (`SelectBook()`), і система надсилає запит на отримання інформації про книгу (`GetBookInfo()`), після чого повертає її користувачу. При відкритті книги (`OpenBook()`), система завантажує її вміст (`LoadContent()`), надсилає дані в `ReaderApp` (`SendBookData()`) та виконує візуалізацію книги (`RenderBook()`).

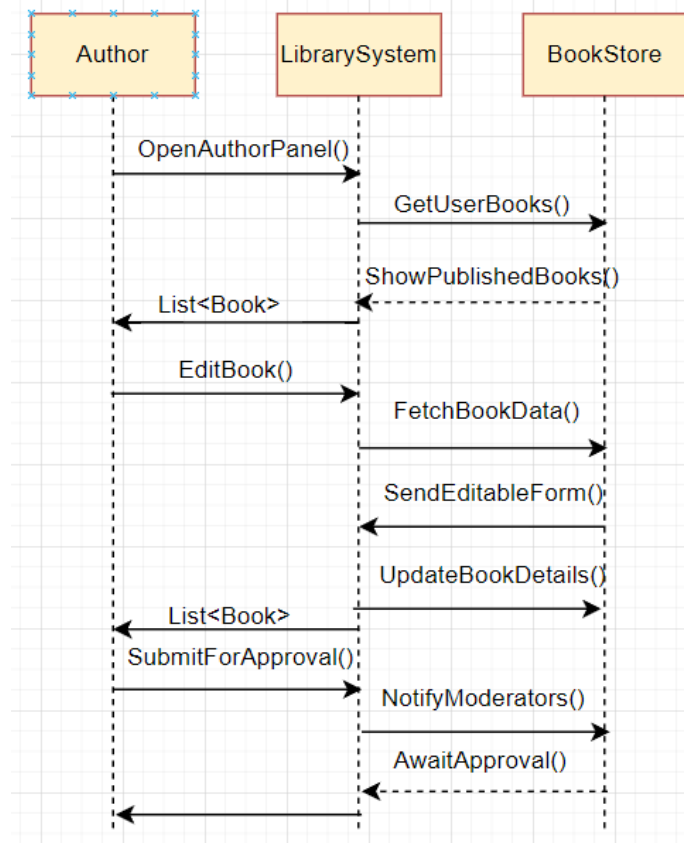


Рисунок 3.3 – Діаграма послідовностей «Управління публікаціями»

Ці діаграми надають цілісне уявлення про архітектуру вебзастосунку, визначають основні компоненти, їх призначення та взаємозв'язки, що є основою для подальшої реалізації та супроводу програмного продукту.

Для кращого розуміння логіки роботи вебсистеми було здійснено моделювання ключових процесів за допомогою діаграм. Використання діаграм послідовності дозволяє наочно відобразити взаємодію між користувачем, клієнтською частиною, сервером та зовнішніми сервісами під час реєстрації. У свою чергу, діаграма діяльності демонструє типові сценарії взаємодії користувача з інтерфейсом сайту при пошуку та перегляді книг. Таке моделювання дозволяє формалізувати алгоритми роботи системи, виявити потенційні вузькі місця та забезпечити ефективну комунікацію між розробниками.

Діаграми діяльності використовуються для представлення послідовності кроків, що відбуваються під час виконання певного сценарію в системі. Вони ілюструють дії, варіанти переходів, розгалуження та умови, що дозволяє чітко зрозуміти логіку реалізації функціоналу. У рамках даного проєкту діаграми

діяльності були створені для відображення основних сценаріїв взаємодії користувача із системою. Зокрема, діаграма діяльності «Отримання книги» (рис. 3.4) демонструє різні шляхи доступу до книжкового контенту. Після відкриття головної сторінки користувач може обрати один із варіантів: перегляд каталогу, пошук книги або ознайомлення з тематичними підбірками. Якщо вибрано каталог – користувач отримує загальний перелік книг і може скористатися функцією сортування або фільтрації. У разі пошуку – він вводить ключові слова, щоб знайти книгу за назвою чи автором. У випадку з добірками – система пропонує список рекомендованих книг. Після формування списку користувач обирає конкретну позицію, і система відкриває сторінку з повною інформацією про книгу.

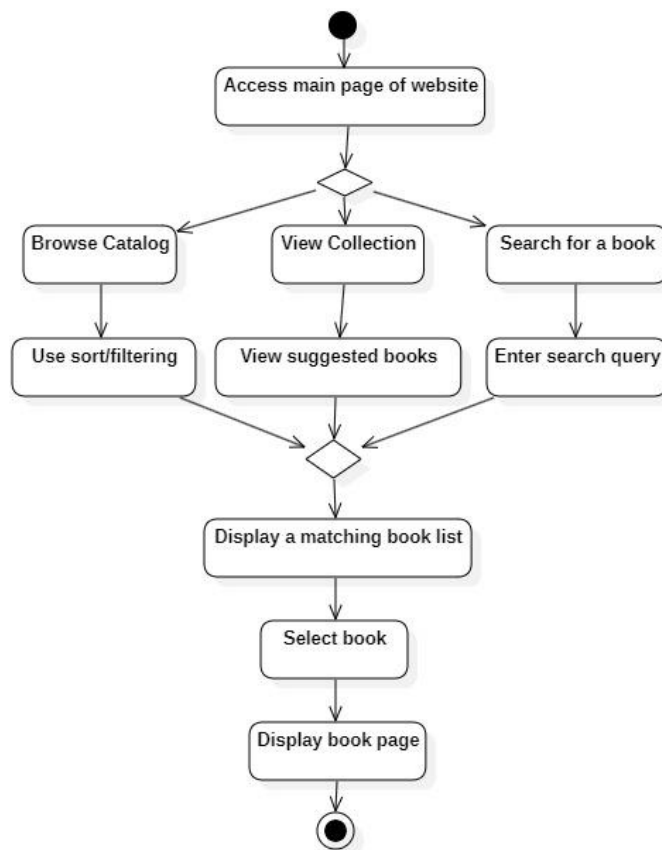


Рисунок 3.4 – Діаграма діяльності «Отримання книги»

Діаграма діяльності «Публікація книги» (рис. 3.5) ілюструє послідовність дій, які виконує користувач під час додавання нової книги до системи. На початковому етапі користувач обирає тип книги – оригінальна чи перекладена. Якщо вказано переклад, система запитує інформацію про автора оригінального твору. У випадку, якщо такого автора ще немає в базі даних, користувач має змогу додати його

вручну. Далі заповнюється основна інформація про книгу, після чого процес публікації завершується.

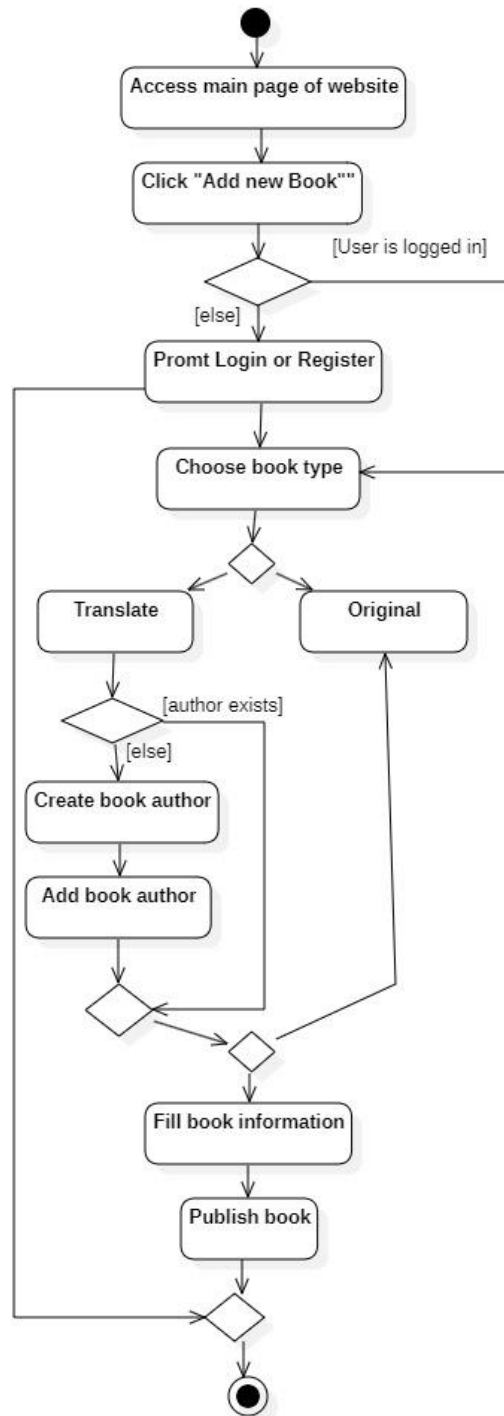


Рисунок 3.5 – Діаграма діяльності «Публікація книги»

Загалом, моделювання процесу публікації книги дозволяє чітко визначити всі необхідні кроки для додавання нового літературного твору до системи, враховуючи можливість роботи як з оригінальними, так і з перекладеними виданнями. Такий підхід забезпечує гнучкість і зручність для користувача, а також сприяє

наповненню бібліотеки актуальним та різноманітним контентом. Завершення процесу публікації книги є логічним підсумком взаємодії користувача із відповідним функціоналом, що підтверджує цілісність і завершеність розробленої моделі роботи системи.

3.2 Опис сценаріїв та інтерфейса користувача

У процесі розробки веб-системи електронної бібліотеки важливим етапом є моделювання поведінки користувачів при взаємодії із системою. Для цього використовуються діаграми варіантів використання (Use Case Diagram), що дозволяють описати основні дії, які може виконувати кожен тип користувача. Це дозволяє чітко структурувати функціональність системи та розмежувати доступ до дій залежно від ролі користувача. У даному підрозділі розглянуто основні сценарії для чотирьох типів користувачів: гість, читач, автор та адміністратор.

Гість – це користувач, який відкрив сайт без попередньої реєстрації або авторизації. Його можливості є обмеженими, але достатніми для ознайомлення з функціоналом платформи. Основна мета для гостя — отримати загальне враження про платформу, оцінити доступні книги, переглянути структуру та, за потреби, зареєструватися (рис. 3.6).

Сценарії використання включають:

- пошук літератури – гість може здійснювати пошук за ключовими словами, такими як назва книги, автор, жанр, рейтинг;
- перегляд інформації про книгу – відображення опису книги, її жанру, дати публікації, вікової категорії, кількості сторінок та середньої оцінки;
- перехід на сторінку реєстрації – для отримання повного доступу до функцій (наприклад, читання, коментування або додавання до бібліотеки), гість може зареєструватися або увійти у свій обліковий запис.

Таке обмеження функціоналу дозволяє зберігати безпеку системи, запобігаючи небажаним діям з боку невідомих користувачів, і стимулює реєстрацію.

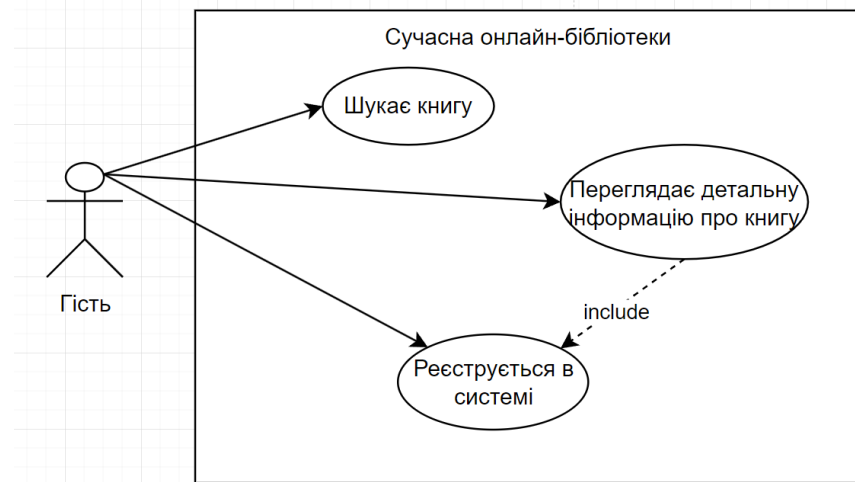


Рисунок 3.6 – Діаграма використання. Сценарій для гостя

Читач – це зареєстрований користувач, який має доступ до основного функціоналу системи. Після авторизації він може взаємодіяти з книгами на глибшому рівні, а також формувати свою персональну бібліотеку. Система зберігає дані читача, що дозволяє організувати персоналізований досвід .

Основні сценарії:

- авторизація – вхід у систему за допомогою логіна та пароля;
- каталог книг – доступ до всіх опублікованих книг, сортування та фільтрація;
- читання книг онлайн – відкриття повної версії книги для перегляду у вбудованому PDF або EPUB переглядачі;
- додавання книги до особистої бібліотеки – збереження обраних книг у розділі "Моя бібліотека" для зручного доступу пізніше;
- залишення відгуку або рейтингу – користувач може оцінити книгу, залишити коментар або рецензію;
- редагування профілю – оновлення персональних даних, зображення профілю, налаштування приватності.

Система підтримує збереження прогресу читання, дозволяючи читачу продовжити читання з того місця, на якому він зупинився (рис. 3.7).

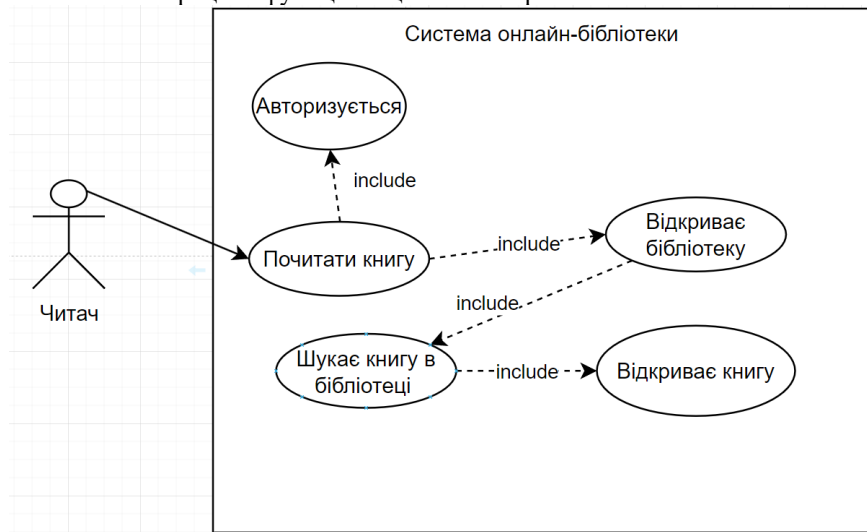


Рисунок 3.7 – Діаграма використання. Сценарій для зареєстрованого користувача

Автор – це зареєстрований користувач, який має право публікувати власні твори. Такий користувач має доступ до спеціального інтерфейсу для завантаження та редагування книг. Основна ідея – надати авторам простий та інтуїтивний інструмент для публікації своїх творів.

Основні сценарії:

- авторизація – перевірка прав доступу, адже публікація дозволена лише зареєстрованим авторам;
- завантаження книги – подання файлу (наприклад, PDF, DOCX) через спеціальну форму;
- додавання метаданих – вказання назви, короткого опису, жанру, мови, обкладинки та вікової категорії;
- автоматична перевірка книги – перевірка на наявність забороненого контенту, розмір файлу, форматування;
- публікація книги – після схвалення (автоматичного або адміністративного) книга з’являється у публічному каталозі;
- редагування або видалення опублікованої книги – можливість оновити зміст або приховати твір;

Ця роль стимулює творчу діяльність та формує спільноту авторів, які зацікавлені у поширенні своїх ідей (рис. 3.8);

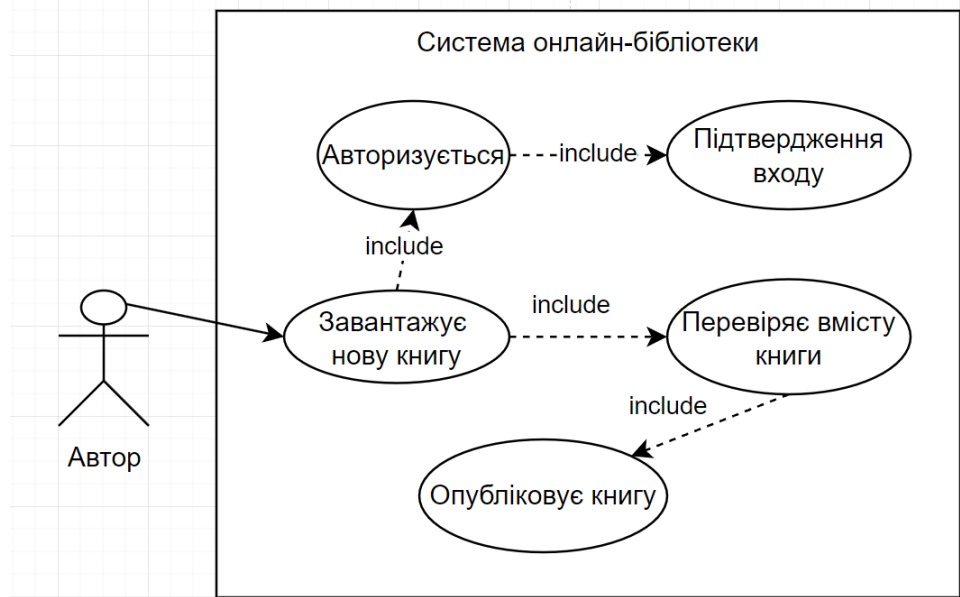


Рисунок 3.8 – Діаграма використання. Сценарій для автора

Адміністратор – це спеціальний користувач із розширеними правами, який виконує модерацію та управління платформою. Його завдання полягають у підтримці порядку, контролі контенту та обслуговуванні користувачів. Доступ адміністратора обмежено лише до довірених осіб, оскільки він може впливати на ключові аспекти роботи сайту.

Основні сценарії:

- модерація контенту – перевірка нових книг, рецензій, коментарів на наявність неприйняттого або шкідливого вмісту;
- управління користувачами – перегляд активності користувачів, блокування або розблокування акаунтів при порушенні правил;
- генерація звітів – створення звітів щодо активності сайту, кількості публікацій, найпопулярніших книг, загальної кількості зареєстрованих користувачів тощо;
- перевірка завантажених книг – у разі автоматичної модерації можливість ручного перегляду й підтвердження;
- налаштування системи – зміна налаштувань платформи, жанрів, категорій, користувацьких прав тощо;

Роль адміністратора критично важлива для забезпечення стабільної, безпечної та контрольованої роботи системи.

Проектування інтерфейсу користувача є важливою складовою процесу розробки програмного забезпечення, оскільки саме через інтерфейс відбувається основна взаємодія користувача з системою. Одним із ключових етапів цього процесу є створення макетів (mockups) – графічних зображень, що демонструють зовнішній вигляд майбутнього застосунку, розміщення основних елементів інтерфейсу та логіку навігації між сторінками.

Макети дозволяють на ранньому етапі узгодити вимоги до дизайну, структури та функціоналу, отримати зворотний зв'язок від потенційних користувачів, уникнути непорозумінь між розробниками та замовниками, а також оптимізувати архітектуру клієнтської частини системи.

У цьому підрозділі буде представлено опис користувацького інтерфейсу вебзастосунку. На зображеннях нижче продемонстровано, як саме виглядатиме сайт після реалізації дизайну та основних компонентів.

Першим елементом інтерфейсу є головна сторінка, яка відкривається при запуску застосунку (рис. 3.9) що містить верхнє меню (хедер), де розміщені логотип сайту, пошуковий рядок, навігаційне меню (Головна, Книги, Автори, Моя бібліотека) та кнопка входу/реєстрації. Нижче розташована карусель із популярними книгами, яка відображає обкладинки, назви, авторів і середній рейтинг творів. Такий підхід дозволяє користувачу одразу ознайомитися з найцікавішим контентом.

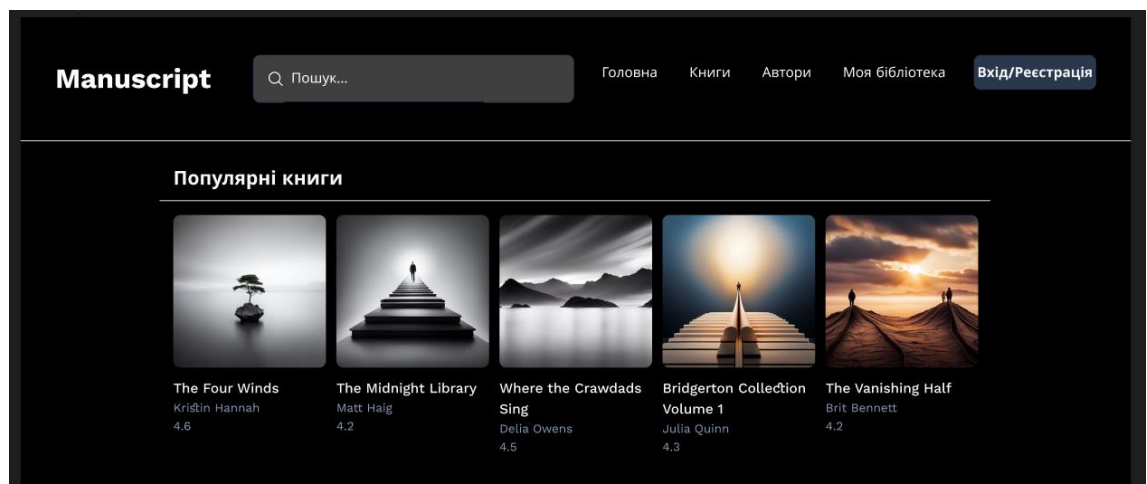


Рисунок 3.9 – Хедер та Популярні книги

Наступним елементом є сторінка входу до особистого кабінету що оформлена у мінімалістичному стилі з контрастними кольорами, що робить її зрозумілою та привабливою. У центрі екрану розміщена форма для введення електронної пошти та пароля, кнопка входу та посилання на реєстрацію. Візуальна структура сприяє швидкому доступу користувача до функцій сайту без зайвих відволікань.

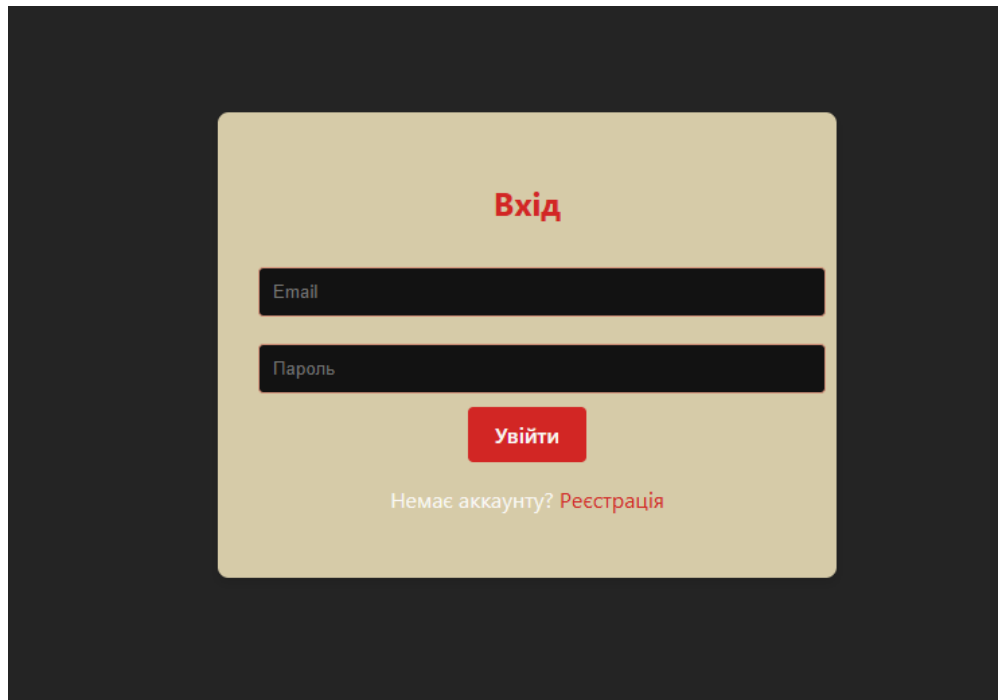


Рисунок 3.10 – Сторінка входу користувача

Ще одним важливим компонентом інтерфейсу є блок новинок, який розміщується нижче каруселі популярних книг (рис. 3.11). Побудований у вигляді сітки з картками, де кожна картка представляє окрему книгу. У картці відображено обкладинку, назву твору, автора та середню оцінку. Такий підхід дозволяє користувачеві швидко ознайомитися з останніми оновленнями бібліотеки, а також візуально оцінити представлені матеріали.

Реалізація сітки забезпечує ефективне використання простору на сторінці та забезпечує адаптивність до різних розмірів екранів. Чітке типографічне розділення елементів (назва, автор, рейтинг) покращує зчитуваність та загальне враження від інтерфейсу.

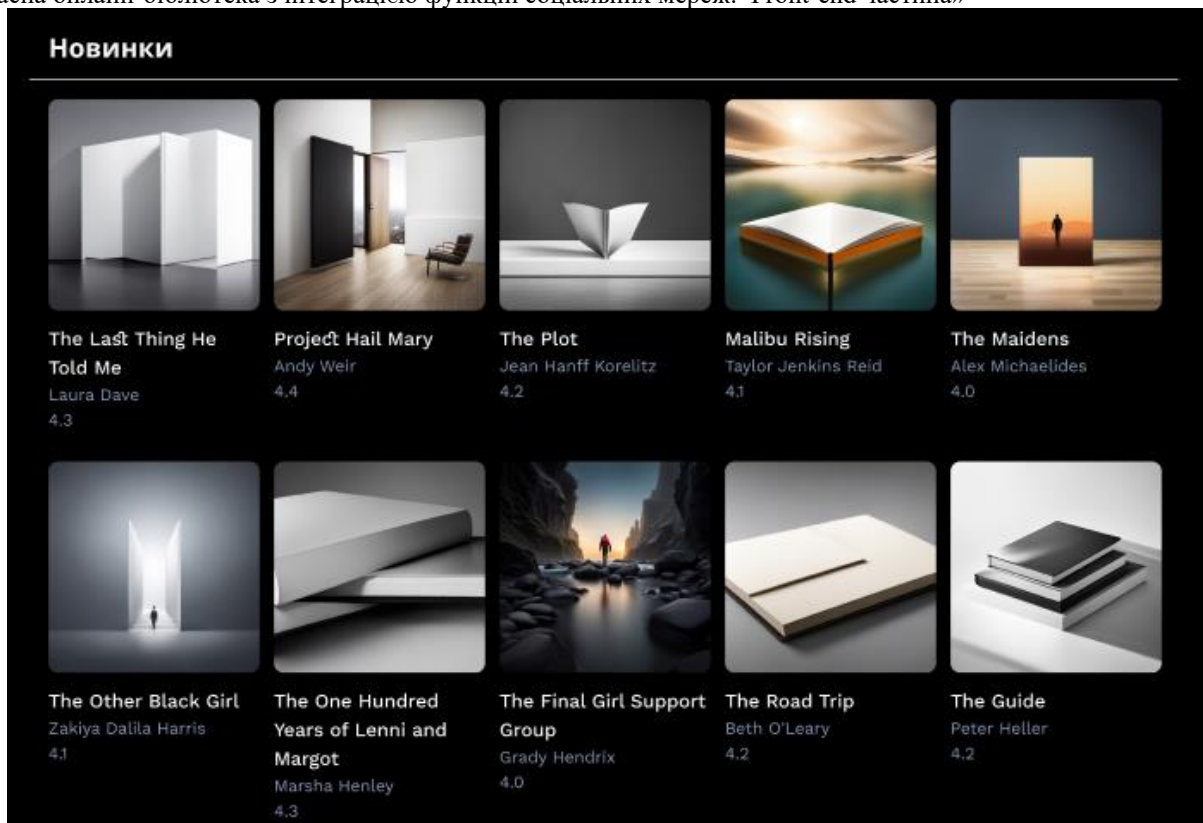


Рисунок 3.11 – Розділ новинок

Наступною в черзі є секція, що складається з двох колонок: "Блоги" та "Події та анонси" (рис. 3.12). Кожна колонка містить перелік карток, у яких відображаються назва твору, ім'я автора та середня оцінка. Такий підхід дозволяє користувачеві швидко ознайомитися як із новими блогами, так і з актуальними подіями чи анонсами на платформі.

Використання двох паралельних колонок забезпечує зручне порівняння та швидкий доступ до різних типів контенту. Картки мають лаконічний дизайн із чітким типографічним виділенням основних елементів, що сприяє легкому сприйняттю інформації. Горизонтальна лінія під заголовками візуально відокремлює секцію від інших блоків сторінки, підвищуючи структурованість і загальну естетику інтерфейсу.

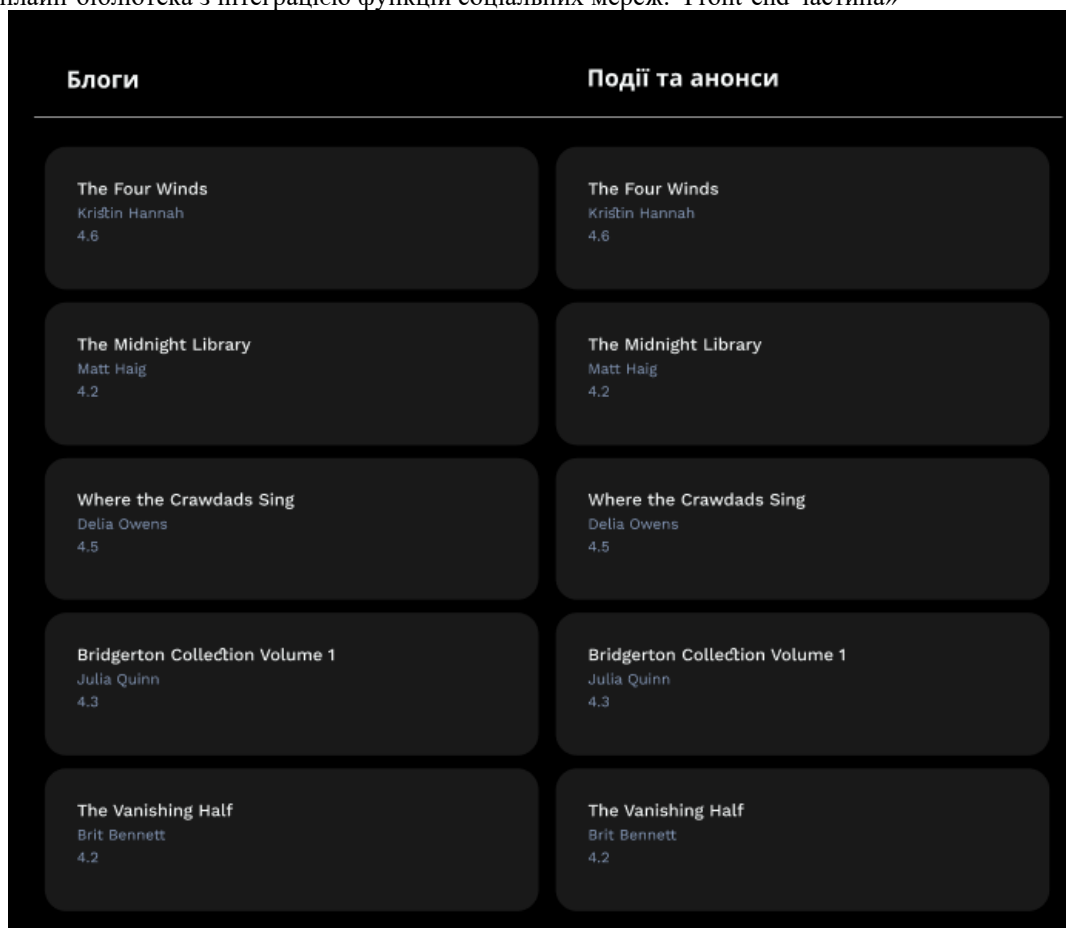


Рисунок 3.12 – Блоги та події

Далі представлено сторінку каталогу книг із можливістю фільтрації (рис 3.13). У лівій частині розташована панель фільтрів, де користувач може обрати вікову категорію (наприклад, 16+ чи 18+) та жанри (комедія, драма, бойовик) за допомогою інтерактивних кнопок. Над фільтрами розміщені кнопки для приховування фільтрів та перемикання вигляду каталогу (наприклад, у вигляді списку).

Основна частина сторінки займає сітка карток книг. Кожна картка містить обкладинку (або її заглушку), назву книги, автора, рейтинг у вигляді зірочок та позначку вікової категорії у верхньому правому куті. Дизайн карток виконаний у світлих тонах, що контрастує з темним фоном сторінки, забезпечуючи хорошу видимість і зручність для користувача. Така структура дозволяє швидко знаходити потрібні книги за заданими параметрами та візуально оцінювати їх основні характеристики.

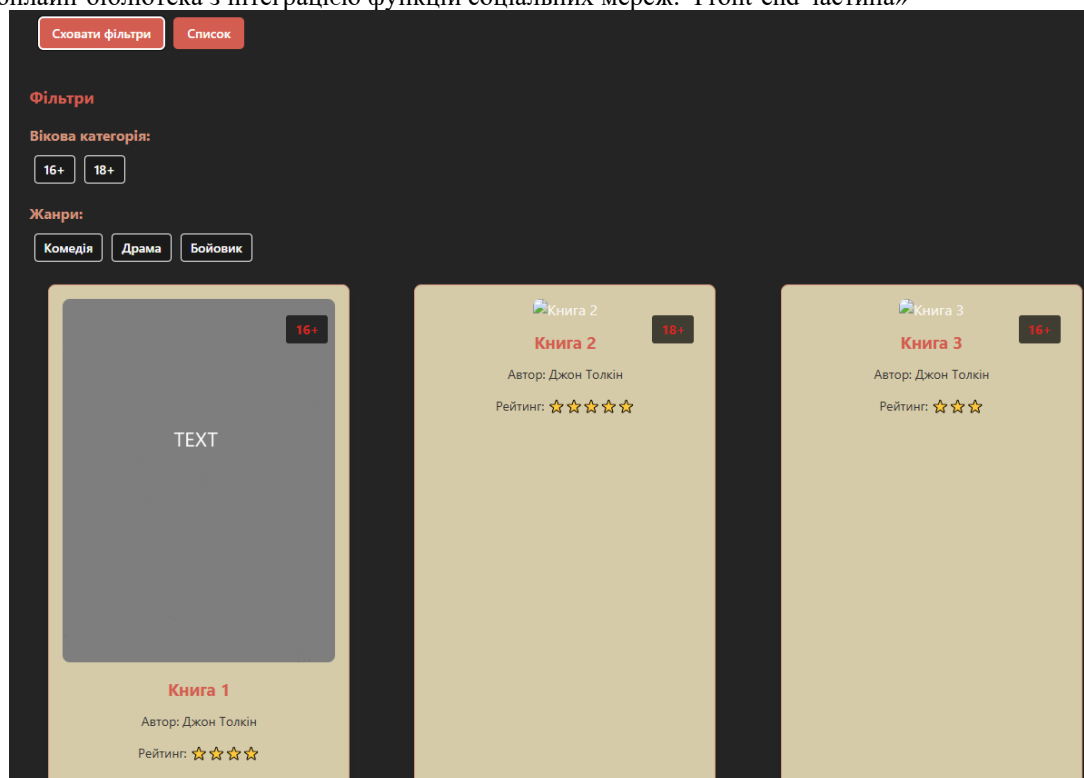


Рисунок 3.13 – Список книг

Сторінка "Моя бібліотека" дозволяє користувачу переглядати, шукати та впорядковувати збережені книги. Зліва розташований sidebar із категоріями (наприклад, "Усі", "Читаю", "В планах", "Брошено", "Прочитано", "Улюблені"), перемикачем вигляду (список або плитка) та кнопками сортування. Основна частина містить список або сітку карток книг, які користувач додав у свою бібліотеку. Є поле для пошуку за назвою. Це дозволяє швидко знаходити потрібну книгу серед особистої колекції, а також організовувати читання за статусом.

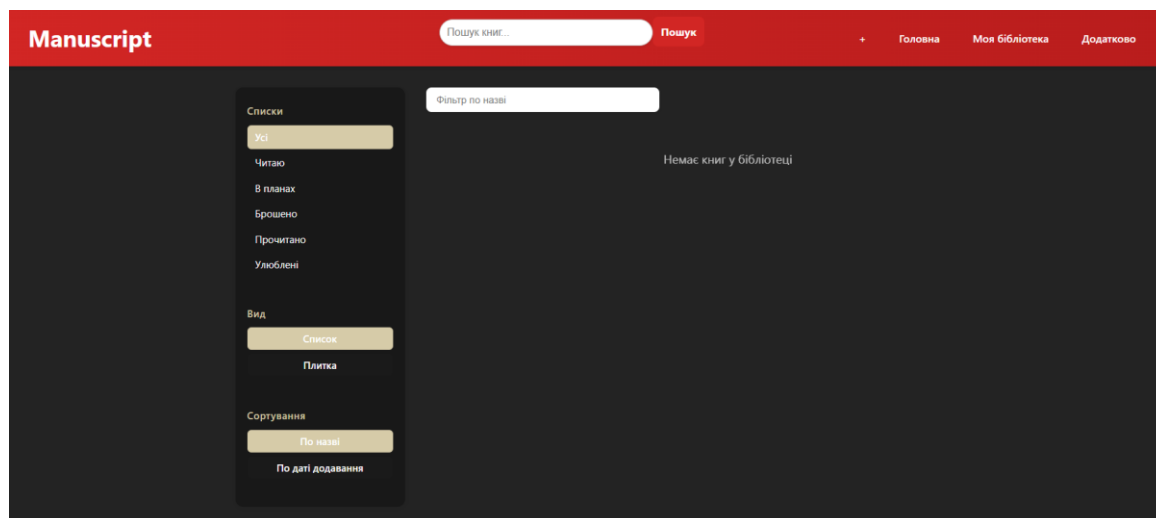


Рисунок 3.14 – Моя бібліотека

На сторінці профілю відображаються персональні дані користувача: аватар, ім'я, біографія. Нижче розташовані вкладки для перемикання між закладками (збережені книги), інвентарем, коментарями та постами користувача. У вкладці "Закладки" показано всі книги, які користувач додав у бібліотеку, з можливістю перейти на сторінку книги. Інші вкладки дозволяють переглядати власні коментарі, пости або інвентар. Таким чином, профіль є персональним центром активності користувача у системі.

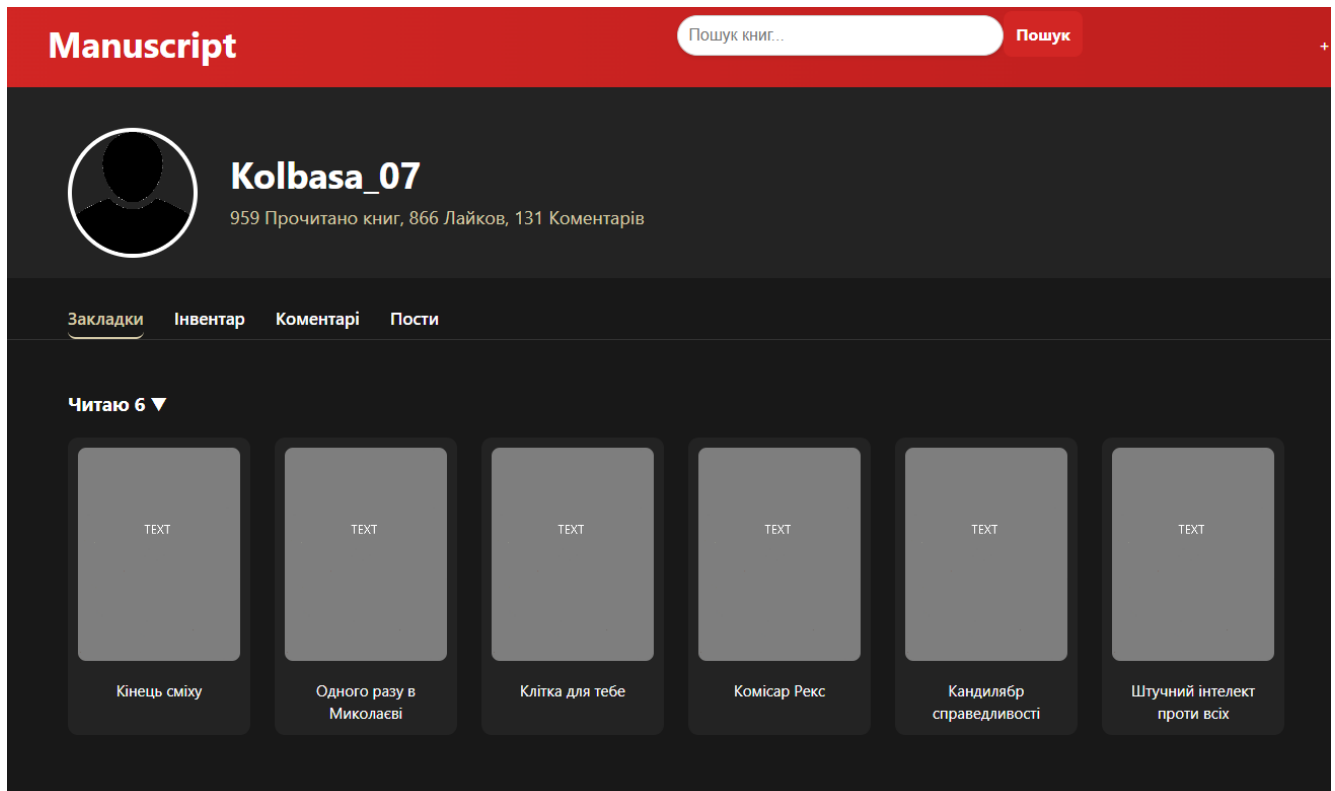


Рисунок 3.15 – Сторінка профілю

Сторінка авторів містить перелік усіх авторів із короткою інформацією: фото, ім'я, біографія та кількість книг, написаних цим автором. Для кожного автора є кнопка "Переглянути книги", яка веде на сторінку з усіма творами цього автора. Це дозволяє користувачам знайомитися з творчістю різних письменників, дізнаватися більше про них та швидко знаходити всі їхні книги у каталозі.

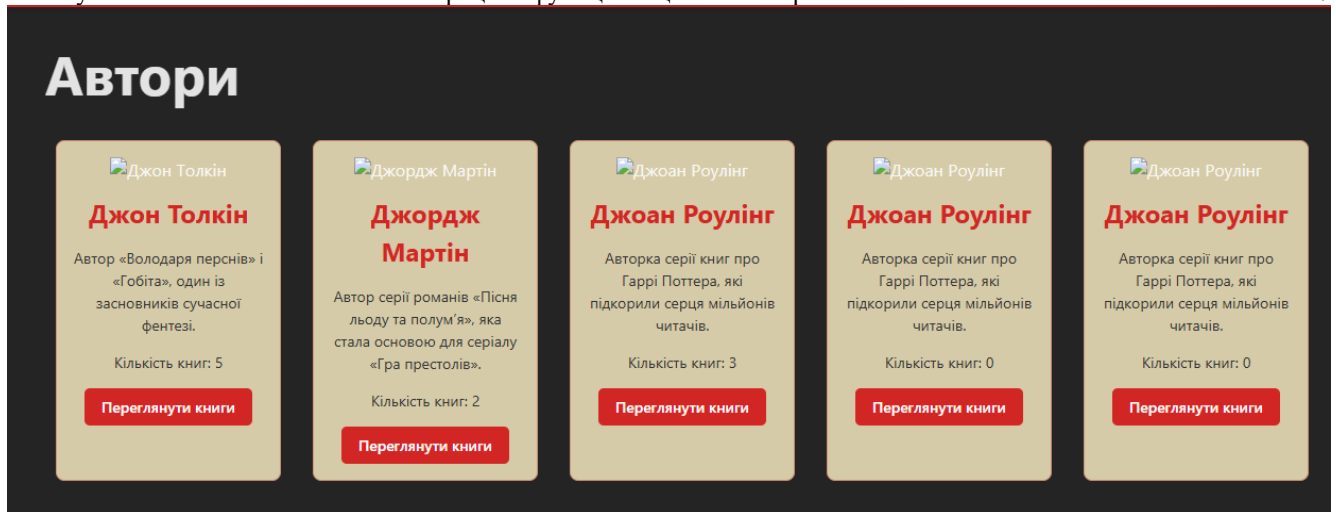


Рисунок 3.16 – Сторінка авторів

Користувач може додавати книги у свою бібліотеку, переглядати їх у різних статусах, шукати та сортувати. У профілі він бачить свою активність і персональні дані. На сторінці авторів можна знайти улюбленого письменника та ознайомитися з його творами. Усі ці функції забезпечують зручну навігацію, персоналізацію та залучення користувача до взаємодії із системою.

Висновки до розділу 3

У третьому розділі було здійснено моделювання основних функцій та процесів фронтенд-частини програмного забезпечення для вебзастосунку електронної бібліотеки. Детально розглянуто функціональні можливості інтерфейсу користувача, зокрема процеси реєстрації, автентифікації, перегляду та пошуку книг, взаємодії з бібліотекою, профілем користувача, а також роботу з авторами.

Особливу увагу приділено користувацьким сценаріям (Use Cases), які демонструють типові взаємодії користувачів із системою – від реєстрації й авторизації до перегляду творів, додавання їх до бібліотеки та залишення коментарів.

Було описано структуру основних сторінок сайту, їхню логіку та призначення. Особливу увагу приділено реалізації каталогу книг із розвиненою системою фільтрації, персоналізованої бібліотеки користувача, профілю з

відображенням активності та сторінки авторів. Наведено UML-діаграми, які ілюструють архітектуру, структуру та взаємозв'язки компонентів системи, а також діаграми діяльності для ключових сценаріїв взаємодії користувача із застосунком.

Розроблений інтерфейс забезпечує інтуїтивно зрозумілу навігацію, адаптивність до різних пристроїв та зручний доступ до всіх основних функцій. Моделювання процесів дозволило формалізувати алгоритми роботи системи, визначити ролі користувачів та їхні права доступу, а також закласти основу для подальшої реалізації та тестування програмного продукту. Таким чином, результати цього розділу створюють фундамент для ефективної розробки, впровадження та супроводу вебзастосунку електронної бібліотеки.

4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Реалізація програмного забезпечення

У кореневій папці `src` розміщені всі основні ресурси клієнтської частини: компоненти, сторінки, дані, стилі, маршрутизація та допоміжні файли. Такий підхід дозволяє логічно розділити функціонал за призначенням та спростити навігацію по коду.

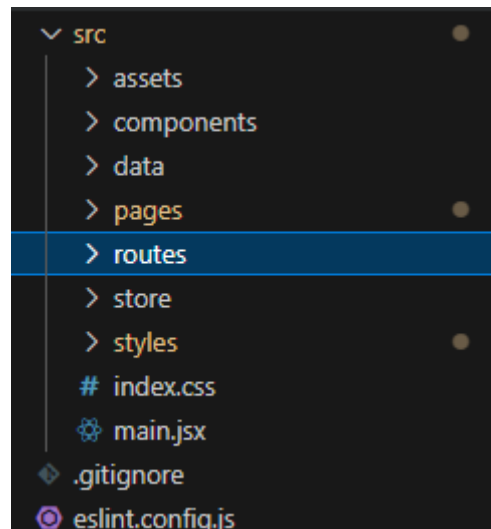


Рисунок 4.1 – Загальна структура папок `src`/

Папка `components` містить багаторазові елементи інтерфейсу, які використовуються на різних сторінках. Наприклад, `Header.jsx` відповідає за верхню панель навігації, а `Footer.jsx` – за нижню частину сайту. Компоненти ізольовані, мають власні стилі та можуть приймати пропси для налаштування поведінки.

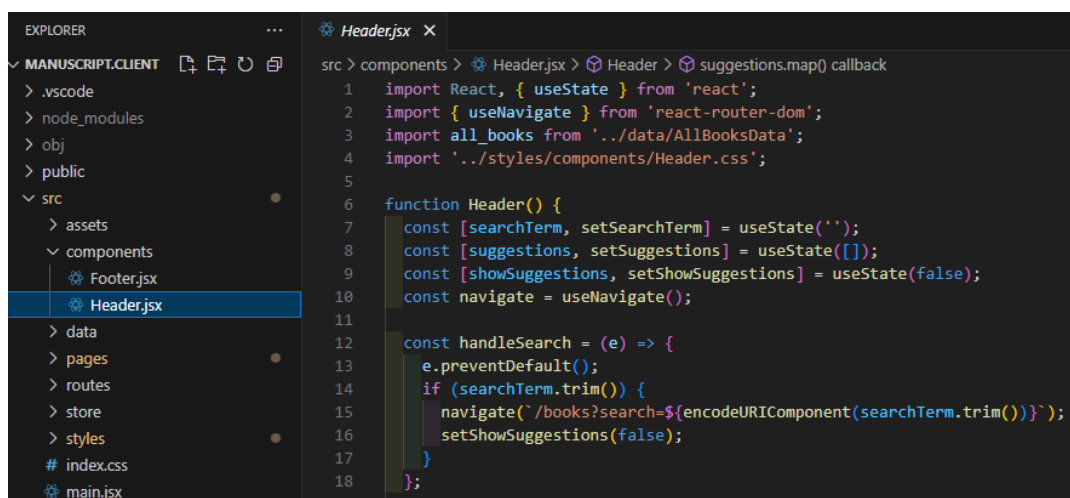


Рисунок 4.2 – Приклад компонента `Header`

У папці `pages` знаходяться всі сторінки застосунку, згруповані за напрямками:

- `main/` – головна сторінка та пов'язані секції;
- `books/` – сторінки для перегляду книг, деталей книги, подій тощо;
- `mylibrary/` – особиста бібліотека користувача;
- `userContent/` – форми для додавання власних творів, перекладів, фанфіків, а також сторінка "Мої додані";
- `authors/` – сторінки авторів та їх деталей;
- `profilePage.jsx`, `LoginRegisterPage.jsx` – профіль користувача та сторінка входу/реєстрації.

Кожна сторінка може містити вкладені підсторінки або форми, що відповідають за окремі сценарії взаємодії.

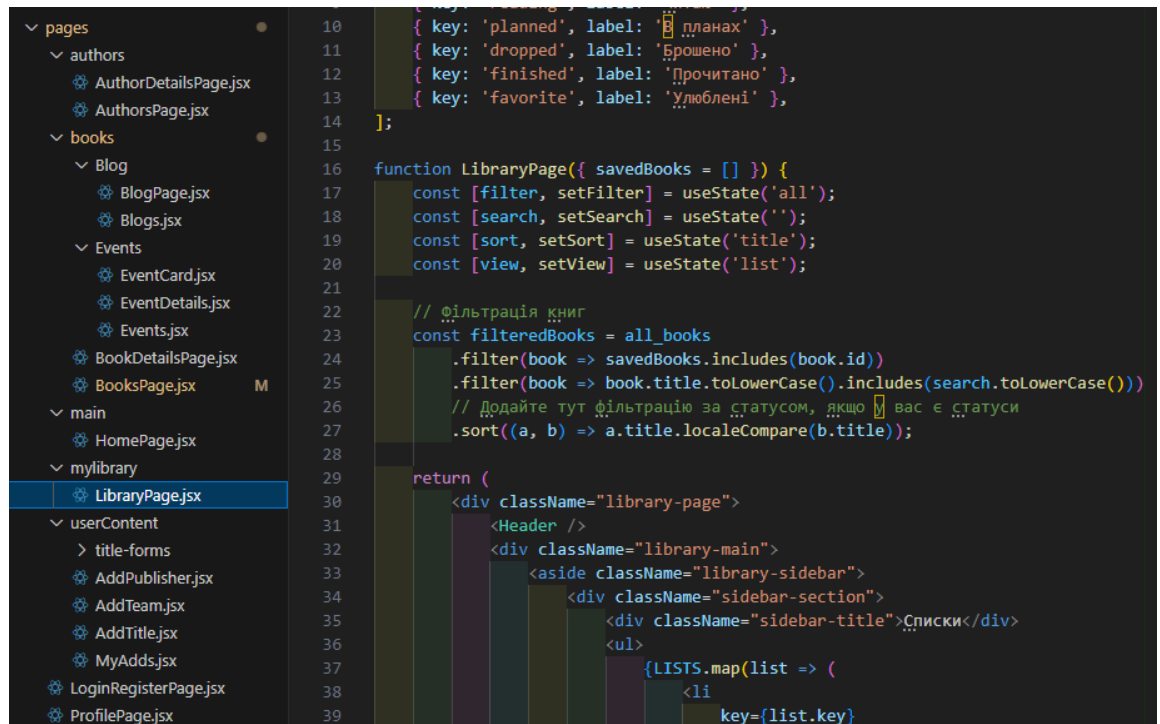


Рисунок 4.3 – Взаємозв'язок сторінок (`pages/`)

Папка `styles` містить CSS-файли, згруповані за функціоналом:

- `components/` – стилі для окремих компонентів (наприклад, `Header`, `Footer`).
- `books/`, `library/`, `main/`, `profile/`, `forms/` – стилі для відповідних сторінок та форм.
- Глобальні стилі (`index.css`) застосовуються до всього застосунку.

Структура стилів дозволяє легко підтримувати єдиний дизайн та швидко знаходити потрібні правила.

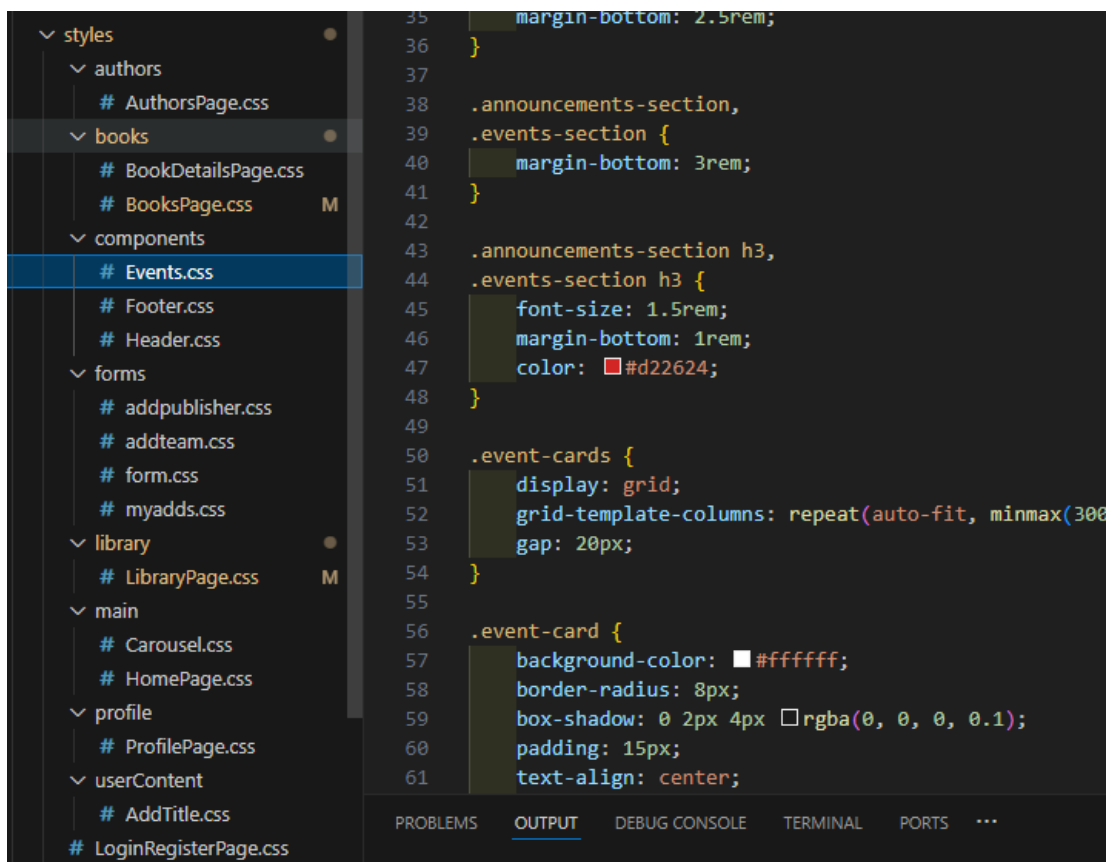


Рисунок 4.4 – Приклад структури стилів

Файл `AppRouter.jsx` у папці `routes` відповідає за маршрутизацію між сторінками. Він використовує `react-router-dom` для визначення шляхів та відображення відповідних компонентів залежно від URL.

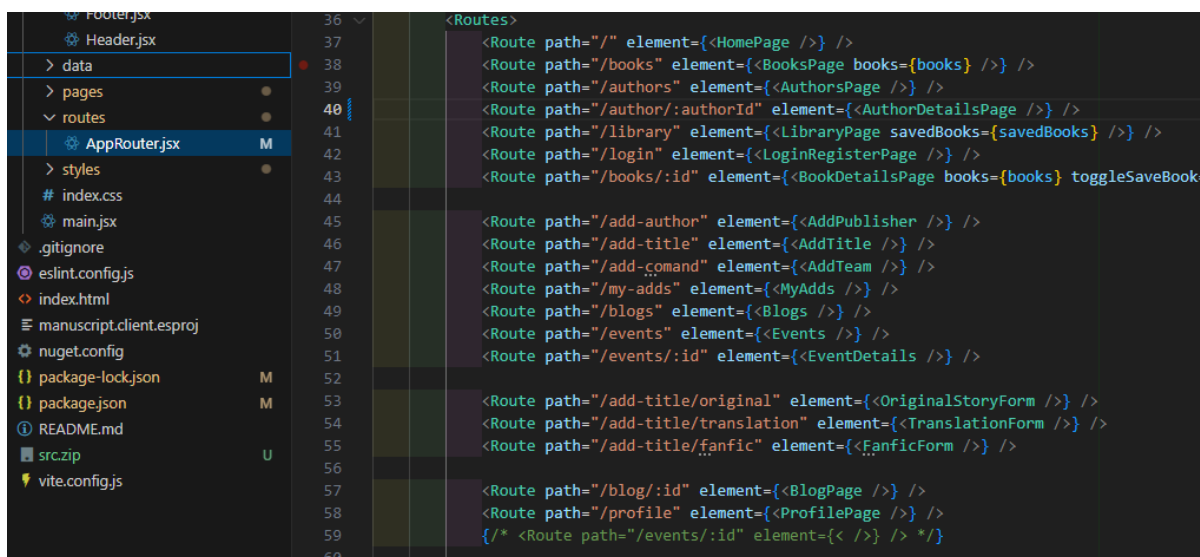


Рисунок 4.5 – Схема маршрутизації

Папка `store` призначена для зберігання глобального стану застосунку (наприклад, через `Context API` або `Redux`). Тут можуть бути файли для управління авторизацією, бібліотекою користувача, налаштуваннями тощо.

Папка `assets` містить статичні ресурси, такі як зображення, іконки, логотипи, які використовуються у компонентах та сторінках.

Файл `main.jsx` є точкою входу застосунку. Тут відбувається підключення глобальних стилів, ініціалізація маршрутизатора та рендеринг кореневого компонента.

Компоненти отримують дані через пропси або імпортують їх із файлів у папці `data`. Сторінки використовують компоненти для побудови складних інтерфейсів, а маршрутизація забезпечує навігацію між ними. Стили підключаються окремо для кожної частини, що дозволяє уникати конфліктів.

Така структура дозволяє легко орієнтуватися у коді, швидко знаходити потрібні файли та розширювати функціонал без ризику порушити роботу інших частин застосунку. Для дипломної роботи рекомендується додати схеми, які ілюструють структуру папок, взаємозв'язки між сторінками, приклади даних та компонування інтерфейсу.

4.2 Тестування програмного забезпечення

Для забезпечення високої якості програмного забезпечення було проведено комплексне тестування ключових функціональних можливостей системи. Тести виконувались згідно з розробленими сценаріями, які відображають реальні дії користувачів. Нижче наведено докладний опис кожного тесту, його мету, послідовність дій, а також очікувані результати.

Перший тест спрямований на перевірку функціональності переходу з основної сторінки, що містить перелік книг, до детальної сторінки конкретної книги. Такий сценарій є базовим для будь-якої платформи, що працює з каталогом книг, оскільки відсутність або некоректність даного переходу суттєво погіршує користувацький досвід.

Спершу користувач повинен виконати вхід у систему, щоб отримати доступ до повного функціоналу сайту. Після успішного логіну необхідно перейти на сторінку зі списком книг, де відображаються всі доступні для перегляду книги у вигляді карток з короткою інформацією. Користувач вибирає певну книгу, натискаючи на її картку. Очікується, що після цієї дії відкриється окрема сторінка з детальним описом книги, яка містить інформацію про її назву, автора, а також інші важливі метадані.

Коректність роботи цього функціоналу перевіряється шляхом порівняння фактичного відображення сторінки з очікуваним. Якщо детальна інформація про книгу відображається без помилок, тест вважається успішним.

Таблиця 4.1 – Перевірка переходу на сторінку книги

№	Сценарій тест-кейсу	Очікуваний результат	Фактичний результат
1	Увійти в систему	Користувач успішно увійшов	Пройдено
2	Перейти на сторінку зі списком книг	Сторінка зі списком книг відкрита	Пройдено
3	Натиснути на картку книги	Відкрилась сторінка з деталями книги (заголовок, автор)	Пройдено

Другий тест призначений для перевірки механізму додавання книги до закладок користувача – важливої функції, що дозволяє зберігати обрані книги для швидкого доступу у майбутньому. Такий функціонал підвищує зручність користування системою та заохочує користувачів повертатися до раніше обраних матеріалів.

У процесі тестування спочатку відбувається вхід у систему для аутентифікації користувача. Далі користувач переходить на сторінку конкретної книги, яку він хоче додати в закладки. Після цього необхідно натиснути кнопку "Додати в закладки". Очікується, що після виконання цієї дії книга буде додана до спеціального розділу в профілі користувача.

Для підтвердження успішності операції після додавання користувач переходить у свій профіль, відкриває вкладку "Закладки" і перевіряє, що обрана книга з'явилася у списку. Якщо книга відображається у вкладці, функціонал працює коректно, і тест вважається пройденим.

Таблиця 4.2 – Додавання книги в закладки

№	Сценарій тест-кейсу	Очікуваний результат	Фактичний результат
1	Увійти в систему	Користувач успішно увійшов	Пройдено
2	Перейти на сторінку книги	Сторінка книги відкрита	Пройдено
3	Натиснути кнопку "Додати в закладки"	Книга додана до закладок	Пройдено
4	Перейти у профіль → вкладка "Закладки"	Книга присутня у списку закладок	Пройдено

Третій тест присвячений перевірці коректності роботи фільтрації книг за жанрами – функції, яка дозволяє користувачам швидко відсортувати список і знайти книги за своїми уподобаннями. Ця можливість є важливою для зручної навігації по великому каталогу.

Таблиця 4.3 – Фільтрація книг за жанром

№	Сценарій тест-кейсу	Очікуваний результат	Фактичний результат
1	Перейти на сторінку книг	Сторінка зі списком книг відкрита	Пройдено
2	Вибрати фільтр за жанром	Показуються лише книги вибраного жанру	Пройдено

Четвертий тест направлений на перевірку можливості користувача додавати коментарі до книги та коректне відображення цих коментарів у системі.

Процедура тестування починається з входу користувача у систему. Потім він переходить на сторінку вибраної книги, де доступна форма або поле для введення коментаря. Користувач залишає свій коментар і підтверджує його додавання.

Очікуваним результатом є те, що новий коментар негайно з'являється у списку коментарів під книгою, що свідчить про коректну роботу функціоналу. Якщо коментар не відображається або з'являються помилки, це вказує на проблему в роботі системи.

Таблиця 4.4 – Додавання коментаря до книги

№	Сценарій тест-кейсу	Очікуваний результат	Фактичний результат
1	Увійти в систему	Користувач успішно увійшов	Пройдено
2	Перейти на сторінку книги	Сторінка книги відкрита	Пройдено
3	Додати коментар	Коментар з'явився у списку коментарів	Пройдено

П'ятий тест перевіряє коректність навігації між різними вкладками у профілі користувача. Профіль містить кілька вкладок, таких як "Закладки", "Інвентар", "Коментарі" та "Пости", кожна з яких відображає відповідний тип інформації.

Тест починається зі входу в систему і переходу до профілю користувача. Далі користувач послідовно перемикає вкладки, натискаючи на кожну з них. Очікується, що при перемиканні вкладок змінюється контент, відповідний обраній вкладці – тобто відображаються закладки, інвентар, коментарі або пости.

Якщо контент змінюється коректно і немає затримок чи помилок у відображенні, тест вважається успішним. Цей тест допомагає гарантувати зручність користування профілем і правильність відображення персоналізованої інформації.

Таблиця 4.5 – Перехід між вкладками профілю

№	Сценарій тест-кейсу	Очікуваний результат	Фактичний результат
1	Увійти в систему	Користувач успішно увійшов	Пройдено
2	Перейти у профіль	Профіль відкритий	Пройдено
3	Перемикає вкладки ("Закладки", "Інвентар", "Коментарі", "Пости")	Контент змінюється відповідно до вкладки	Пройдено

Таким чином, проведене тестування охопило ключові функціональні можливості розробленої системи і підтвердило їхню коректну роботу. Всі критичні сценарії, що відтворюють типову поведінку користувачів, були ретельно перевірені, а отримані результати відповідали очікуванням. Виявлені незначні помилки були оперативно виправлені, що дозволило підвищити стабільність і зручність роботи платформи. Завдяки систематичному підходу до тестування забезпечено високу якість програмного продукту, що є важливою передумовою для його подальшого впровадження та експлуатації.

4.3 Інструкція користувача

Після запуску веб-застосунку користувач потрапляє на головну сторінку, де представлений список популярних і нових книг. У правому верхньому куті розташована кнопка для авторизації або реєстрації. Якщо користувач не має облікового запису, він може пройти реєстрацію, ввівши свою електронну пошту, пароль та псевдонім. Після успішної реєстрації або входу він автоматично потрапляє на головну сторінку у режимі авторизованого користувача.

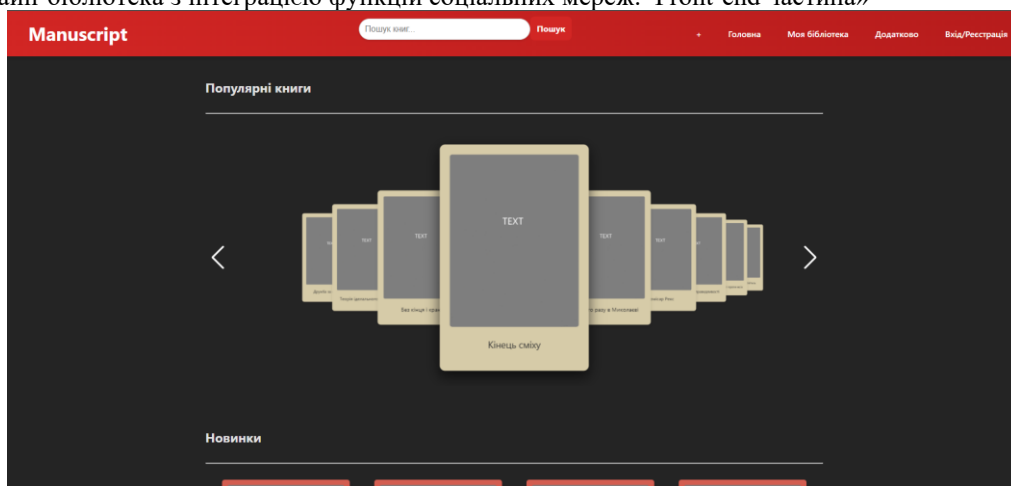


Рисунок 4.6 – Головна сторінка

Після входу користувач може переглядати каталог книг, який доступний через головне меню або посилання «Книги» на головній сторінці. Книги представлені у вигляді карток із зображенням обкладинки, назвою, автором, рейтингом та віковою категорією. Натиснувши на картку книги, відкривається сторінка з детальною інформацією, яка включає опис, автора, жанр, рейтинг та блок коментарів.

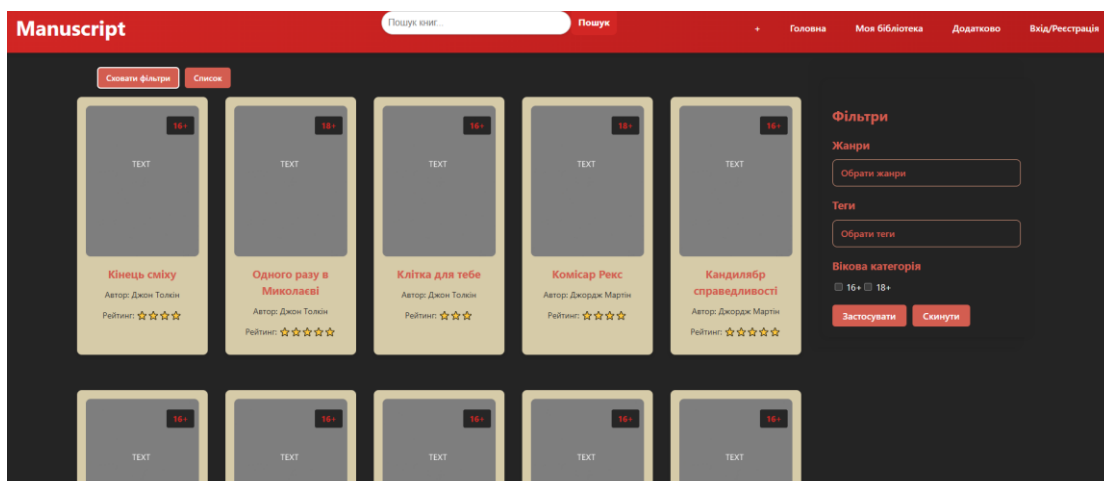


Рисунок 4.7 – Сторінка списку книг

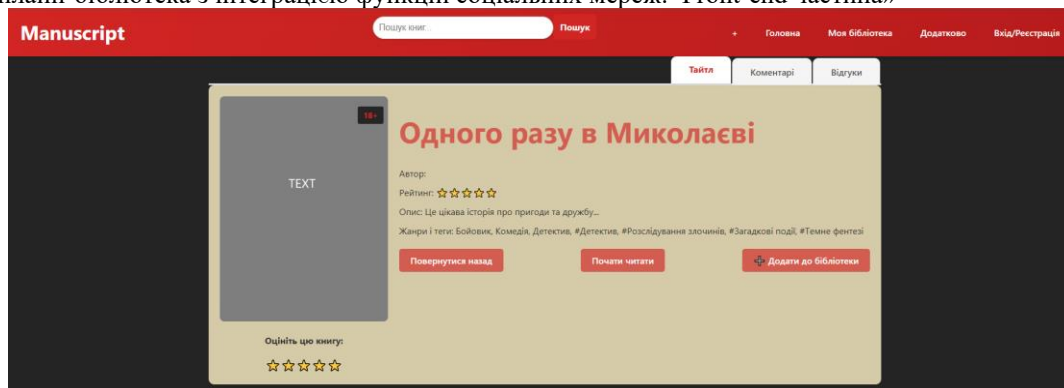


Рисунок 4.8 – Сторінка з деталями книги

Користувач також може залишити коментар до книги. Для цього необхідно перейти на сторінку обраної книги, прокрутити сторінку донизу, ввести текст у поле для коментарів і натиснути кнопку надсилання. Коментар одразу з'являється у списку нижче, разом із іменем користувача та датою публікації.

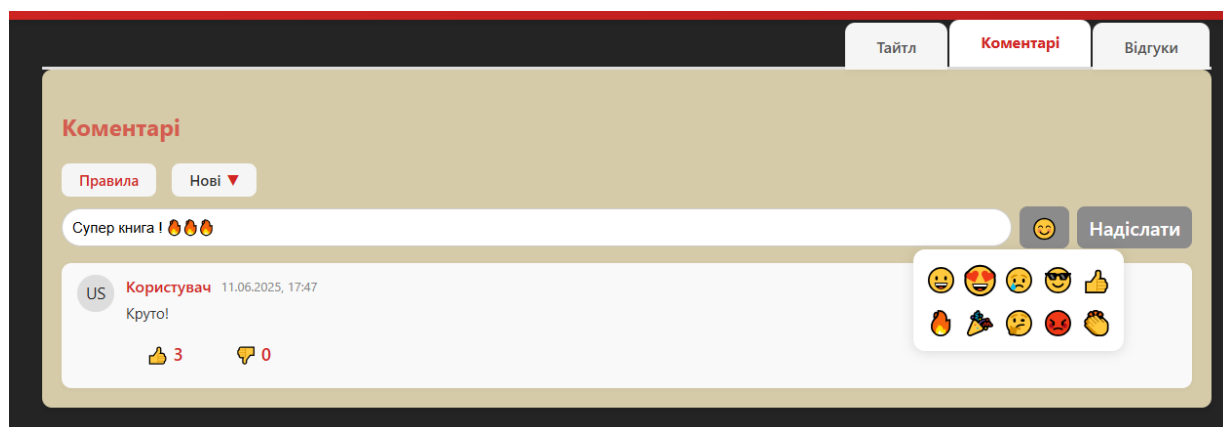


Рисунок 4.9 – Додавання коментаря

На сторінці книги користувач має можливість додати її до закладок за допомогою відповідної кнопки. Після додавання книги до закладок її можна знайти у профілі користувача, у вкладці «Закладки». Перехід до профілю здійснюється через головне меню. У профілі доступно декілька вкладок: «Закладки», «Інвентар», «Коментарі» та «Пости». У вкладці «Закладки» відображаються всі збережені книги.

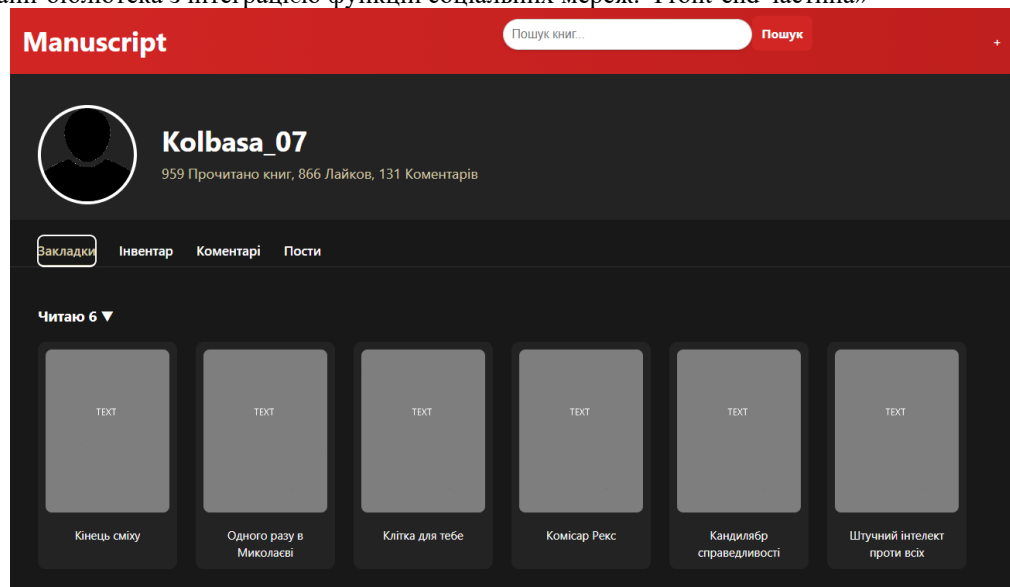


Рисунок 4.10 – Сторінка користувача

Для завершення роботи користувач може вийти зі свого облікового запису, натиснувши кнопку «Вийти» у меню профілю. Після цього застосунок повертає користувача на головну сторінку у гостьовому режимі. Таким чином, інтерфейс платформи є зручним для користувача, а основні функції легко доступні завдяки зрозумілому навігаційному меню та структурі сторінок.

Висновки до розділу 4

У четвертому розділі було детально розглянуто реалізацію клієнтської частини програмного забезпечення, включно з описом структури проєкту, основних компонентів, сторінок, стилів, маршрутизації та механізмів взаємодії між частинами системи. Логічне групування коду за функціональними ознаками сприяє зручності підтримки, розширення й масштабування застосунку.

Проведена реалізація забезпечує чітку архітектуру, ізолюваність компонентів та гнучкість при модифікації інтерфейсу. Структура папок дозволяє ефективно організувати навігацію, повторне використання коду та централізоване управління станом.

Також було виконано тестування розробленого програмного забезпечення: підібрано відповідну методику тестування, сформовано тести для перевірки ключових функцій та проведено аналіз результатів. Тестування показало коректну

роботу застосунку при різних сценаріях використання, що підтверджує якість реалізації функціоналу.

Таким чином, програмне забезпечення, розроблене в межах кваліфікаційної роботи, відповідає вимогам до зручності, модульності та надійності, що робить його придатним до подальшого використання, розширення та впровадження.

ВИСНОВКИ

У межах виконаної кваліфікаційної роботи було повністю реалізовано цикл проектування, моделювання та розробки клієнтської частини вебплатформи для публікації та обміну літературними творами під назвою *Manuscript*. Робота охоплює як теоретичні аспекти аналізу предметної області, так і практичні етапи створення сучасного вебзастосунку з адаптивним інтерфейсом і базовим функціоналом.

У першому розділі здійснено системний аналіз сучасних онлайн-платформ для самостійної публікації літературних творів. Визначено ключові компоненти таких систем, їхні переваги й недоліки, що дозволило обґрунтувати актуальність створення власного рішення, яке б поєднувало найкращі практики та уникало типових проблем існуючих сервісів.

У другому розділі сформовано технічне підґрунтя для реалізації програмного продукту. Проведено обґрунтований вибір інструментів та технологій для розробки, створено специфікацію вимог до системи, що забезпечило структурований підхід до подальшої реалізації інтерфейсу.

Третій розділ присвячено моделюванню інтерфейсу користувача, визначенню основних функцій і сценаріїв взаємодії з системою. За допомогою UML-діаграм, Use Case-сценаріїв та логічного опису сторінок створено детальну модель функціонування вебзастосунку, що дозволяє чітко уявити його поведінку та логіку.

У четвертому розділі реалізовано клієнтську частину застосунку відповідно до попередньо визначених вимог. Розроблено модульну архітектуру, що дозволяє легко масштабувати, підтримувати та оновлювати продукт. Виконане тестування підтвердило правильність роботи основних функцій, надійність і зручність інтерфейсу для кінцевого користувача.

Отже, результати виконаної роботи підтверджують досягнення поставленої мети – створення основи для функціонального вебзастосунку, який дозволяє користувачам зручно публікувати, переглядати, зберігати та оцінювати літературні

твори. Запропоноване рішення є сучасним, адаптивним, масштабованим та відкритим до подальшого розвитку й впровадження.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Розробка веб-сайтів для початківців: HTML – CSS – JavaScript. Самвидав, 2022.
2. Розробка масштабованих веб-додатків з використанням фреймворку React та бази даних MongoDB. / Н. Христинець та ін. *COMPUTER-INTEGRATED TECHNOLOGIES: EDUCATION, SCIENCE, PRODUCTION*. 2021. № 45. С. 97–102. URL: <https://doi.org/10.36910/6775-2524-0560-2021-45-14> (дата звернення: 02.12.2024).
3. API Documentation & Design Tools for Teams | Swagger. *API Documentation & Design Tools for Teams | Swagger*. URL: <https://swagger.io/> (date of access: 09.01.2025).
4. bin Uzayr S. Basics of Vue.js. *Mastering Vue.js*. Boca Raton, 2022. P. 23–58. URL: <https://doi.org/10.1201/9781003310464-2> (date of access: 01.05.2025).
5. Brandmark - make your logo in minutes. *Brandmark - make your logo in minutes*. URL: <https://app.brandmark.io/v3/> (date of access: 30.12.2024).
6. Business Name Generator. *Business Name Generator - free AI-powered naming tool - Namelix*. URL: <https://namelix.com/> (date of access: 30.01.2025).
7. CSS Reference. *W3Schools Online Web Tutorials*. URL: <https://www.w3schools.com/cssref/index.php> (date of access: 22.04.2025).
8. Fox R. Programming and Programming Languages. *Information Technology*. Second edition. | Boca Raton : CRC Press, 2020., 2020. P. 285–327. URL: <https://doi.org/10.1201/9781003050971-8> (date of access: 01.05.2025).
9. Kennedy A., de León I. CSS Style Guide. *Pro CSS for High Traffic Websites*. Berkeley, CA, 2011. P. 37–66. URL: https://doi.org/10.1007/978-1-4302-3289-6_2 (date of access: 01.05.2025).
10. Max B. React: The Complete Beginner's Guide to React. Independently Published, 2019. 138 p.
11. React documentation. *React*. URL: <https://react.dev/> (date of access: 21.04.2025).

12. REACT.JS: TOOLS AND TECHNIQUES TO OPTIMISE PERFORMANCE. *International Research Journal of Modernization in Engineering Technology and Science*. 2023. URL: <https://doi.org/10.56726/irjmets47911> (date of access: 01.05.2025).
13. S S. K., M G. Web Application Using HTML, CSS, Java script and Java. *International Journal of Innovative Research in Engineering*. 2023. P. 124–127. URL: <https://doi.org/10.59256/ijire.2023040363> (date of access: 01.05.2025).
14. Uzayr S. b. Bootstrap and JavaScript. *Bootstrap*. Boca Raton, 2022. P. 157–182. URL: <https://doi.org/10.1201/9781003309383-5> (date of access: 01.05.2025).
15. WhatTheFont Font Finder - Identify Fonts by Image. *MyFonts | Buy and Download Fonts*. URL: <https://www.myfonts.com/pages/whatthefont?step=crop> (date of access: 14.04.2025).
16. Bhat R., Agarmore S., Mahalle H., Pitale P. Exploring advanced web development technologies: A comprehensive overview. *AIP Conference Proceedings*. Vol. 3227. No. 1. AIP Publishing LLC, 2025. URL: <https://doi.org/10.1063/5.0241589>

