

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем

_____Юрій КОНДРАТЕНКО

«____»_____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

РОЗРОБКА ВЕБЗАСТОСУНКУ ОНЛАЙН-БІБЛІОТЕКИ

Спеціальність 122 Комп'ютерні науки

Освітня програма «Комп'ютерні науки»

Здобувач

_____Єлизавета СІНЧЕНКО

«____»_____ 2025 р.

Керівник старший викладач кафедри ІІЗ

_____Марина ФАЛЕНКОВА

«____»_____ 2025 р.

Миколаїв – 2025

Чорноморський національний університет імені Петра Могили
(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем

_____ Юрій КОНДРАТЕНКО

« ____ » _____ 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Сінченко Єлизавети Костянтинівни

((прізвище, ім'я, по батькові здобувача))

1. Тема кваліфікаційної роботи: «Розробка вебзастосунку онлайн-бібліотеки».
Керівник роботи: Фаленкова Марина Володимирівна, старший викладач кафедри ІІЗ.
Затверджена наказом ЧНУ ім. Петра Могили від «18» листопада 2024 р. № 321.
2. Строк представлення кваліфікаційної роботи студентом «13» червня 2025 р.
3. Очікуваний результат: розробка вебзастосунку онлайн-бібліотеки включатиме: базу даних для інформації про книги та авторів; механізм пошуку; інтуїтивно зрозумілий і адаптивний інтерфейс користувача; адміністративну панель для редагування контенту;
4. Перелік питань, що підлягають розробці: аналіз існуючих онлайн-бібліотек і платформ для розповсюдження літературних ресурсів; вибір технологій та

інструментів для створення вебзастосунку; розробка дизайну вебзастосунку для забезпечення адаптивності інтерфейсу та зручності користування; розробка вебзастосунку.

5. Перелік графічних матеріалів: презентація.

Керівник роботи

(Особистий підпис)

Марина ФАЛЕНКОВА

Здобувач

(Особистий підпис)

Єлизавета СІНЧЕНКО

Дата видачі завдання «17» грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН
кваліфікаційної роботи

Тема: Розробка вебзастосунку онлайн-бібліотеки

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	17.12.2024	17.12.2024	Виконано
2	Аналіз предметної області та постановка задачі	24.04.2025	25.04.2025	Виконано
3	Аналіз існуючих онлайн-бібліотек і платформ для розповсюдження літературних ресурсів	26.04.2025	28.04.2025	Виконано
4	Огляд існуючих технологій та інструментів для вирішення поставленої задачі. Розробка дизайну вебзастосунку	30.04.2025	14.05.2025	Виконано
5	Реалізація обраних технологій з аналізом отриманих результатів	23.05.2025	28.05.2025	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	16.05.2025	16.05.2025	Виконано
7	Корегування роботи за результатами попереднього захисту	19.05.2025	21.05.2025	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	30.05.2025	30.05.2025	Виконано
9	Доробка та остаточне оформлення КР	31.05.2025	31.05.2025	Виконано
10	Подання КРБ на рецензію	04.06.2025	09.06.2025	Виконано
11	Подання КР, її електронної копії та інших документів до захисту	12.06.2025	12.06.2025	Виконано
12	Захист КРБ перед екзаменаційною комісією (ЕК)	20.06.2025	20.06.2025	

Керівник роботи

(Особистий підпис)

Марина ФАЛЕНКОВА

Здобувач

(Особистий підпис)

Єлизавета СІНЧЕНКО

Дата складання календарного плану
«14» квітня 2025 р.

АНОТАЦІЯ

до кваліфікаційної роботи
здобувачки групи 401 ЧНУ ім. Петра Могили

Сінченко Єлизавети Костянтинівни

на тему: **“РОЗРОБКА ВЕБЗАСТОСУНКУ ОНЛАЙН-БІБЛІОТЕКИ”**

Актуальність роботи. З розвитком цифрових технологій виникає потреба у створенні ефективних онлайн-платформ для доступу до літературних ресурсів. Онлайн-бібліотеки стають важливим інструментом для популяризації читання, збереження культурної спадщини та забезпечення вільного доступу до знань для широкої аудиторії.

Об’єктом роботи є процес створення вебзастосунку онлайн-бібліотеки для збереження, пошуку та перегляду інформації про літературні твори та їх авторів.

Предметом є сукупність інструментів для розробки вебзастосунку онлайн-бібліотеки.

Метою роботи є створення зручної платформи для пошуку та перегляду даних про літературні твори, де кожен користувач зможе легко отримати доступ до детальної інформації про книги, біографії авторів та їхні творчі здобутки.

Важливу роль у роботі відіграє інтуїтивний дизайн, що забезпечує легку навігацію та швидкий доступ до необхідної інформації. Особливу увагу приділено розробці пошукової системи, що дозволяє шукати книги за назвою. В рамках дослідження буде проведено аналіз існуючих платформ та визначено ключові функції.

Кваліфікаційна робота складається з чотирьох розділів. Загальний обсяг роботи – 84 сторінки. Робота включає 56 рисунків і 31 використаних джерел.

Ключові слова: вебсайт, енциклопедія, література, онлайн-бібліотека, пошукова система, база даних.

ABSTRACT

to the qualification work by the student of the group 401 of Petro Mohyla Black Sea
National University

Sinchenko Yelyzaveta

on the topic: **“ONLINE LIBRARY WEB APPLICATION DEVELOPMENT”**

The relevance With the development of digital technologies, there is a need to create effective online platforms for accessing literary resources. Online libraries are becoming an important tool for promoting reading, preserving cultural heritage, and ensuring free access to knowledge for a wide audience.

The object of the work is the process of creating an online library web application for storing, searching, and viewing information about literary works and their authors.

The subject is a set of tools for developing an online library web application.

The goal of the work is to create a convenient platform for searching and viewing data about literary works, where each user will be able to easily access detailed information about books, biographies of authors, and their creative achievements.

An important role in the work is played by intuitive design, which provides easy navigation and quick access to the necessary information. Particular attention is paid to the development of a search engine that allows you to search for books by title. The study will analyze existing platforms and identify key functions.

The qualification work consists of four sections. The total volume is 84 pages. The work includes 56 figures and 31 references.

Keywords: website, encyclopedia, literature, online library, search engine, database.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ІСНУЮЧИХ ОНЛАЙН-БІБЛІОТЕК І ПЛАТФОРМ ДЛЯ РОЗПОВСЮДЖЕННЯ ЛІТЕРАТУРНИХ РЕСУРСІВ.....	5
1.1 Опис предметної сфери.....	5
1.2 Приклади платформ для розповсюдження літературних ресурсів.....	6
1.3 Електронна енциклопедія. Аналіз українських онлайн-бібліотек.....	9
1.4 Постановка задачі. Вимоги до програмної реалізації.....	12
Висновки до розділу 1.....	13
2 ВИБІР ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ВЕБЗАСТОСУНКУ ОНЛАЙН-БІБЛІОТЕКИ.....	14
2.1 UseCaseDiagram сайту.....	14
2.2 Технології для створення вебзастосунку. Інструменти для роботи.....	18
2.3 Visual Studio Code.....	19
2.4 Figma.....	21
2.5 HTML, CSS та JavaScript.....	22
Висновки до розділу 2.....	26
3 РОЗРОБКА ДИЗАЙНУ ВЕБЗАСТОСУНКУ ОНЛАЙН-БІБЛІОТЕКИ.....	28
3.1 Робота над дизайном. Вебдизайн.....	28
3.2 Логотип.....	30
3.3 Загальний дизайн. Сторінки сайту.....	33
Висновки до розділу 3.....	40
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН-ЕНЦИКЛОПЕДІЇ.....	42

4.1 Етапи роботи над розробкою вебзастосунку	42
4.2 Результат програмної реалізації. Тестування функцій вебзастосунку	43
Висновки до розділу 4.....	54
ВИСНОВКИ	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	56
ДОДАТОК А Лістинг коду головної сторінки	59
ДОДАТОК Б Лістинг коду сторінки книги	63
ДОДАТОК В Лістинг коду сторінки авторизації	69
ДОДАТОК Г Лістинг коду загальної сторінки жанрів	72
ДОДАТОК Ґ Лістинг коду сторінки окремого жанру.....	76

ВСТУП

Обрана тема є актуальною в сучасному світі. З розвитком цифрових технологій виникає потреба у створенні ефективних онлайн-платформ для доступу до літературних ресурсів. Онлайн-бібліотеки стають важливим інструментом для популяризації читання, збереження культурної спадщини та забезпечення вільного доступу до знань для широкої аудиторії.

У сучасному середовищі існує проблема обмеженого доступу до якісної літературної інформації в єдиному зручному форматі. Багато існуючих ресурсів мають незручну навігацію або обмежений обсяг контенту. У зв'язку з цим актуальною є розробка вебзастосунку онлайн-бібліотеки, що об'єднає велику кількість книг та інформації про авторів у зручному, доступному форматі.

Основні завдання включатимуть:

- аналіз існуючих онлайн-бібліотек і платформ для розповсюдження літературних ресурсів;
- вибір технологій та інструментів для створення вебзастосунку;
- розробка дизайну вебзастосунку;
- розробка програмної реалізації онлайн-бібліотеки.

Об'єктом кваліфікаційної роботи є процес створення вебзастосунку онлайн-бібліотеки для збереження, пошуку та перегляду інформації про літературні твори та їх авторів.

Предметом кваліфікаційної роботи є сукупність інструментів для розробки вебзастосунку онлайн-бібліотеки.

Метою кваліфікаційної роботи є створення зручної платформи для пошуку та перегляду даних про літературні твори, де кожен користувач зможе легко отримати доступ до детальної інформації про книги, біографії авторів та їхні творчі здобутки.

1 АНАЛІЗ ІСНУЮЧИХ ОНЛАЙН-БІБЛІОТЕК І ПЛАТФОРМ ДЛЯ РОЗПОВСЮДЖЕННЯ ЛІТЕРАТУРНИХ РЕСУРСІВ

1.1 Опис предметної сфери

У час розвитку інформаційних технологій звичайні бібліотеки поступово інтегруються в цифровий простір. Онлайн-бібліотеки та електронні енциклопедії стають невід’ємною частиною сучасного суспільства, популяризуючи літературну спадщини та роблячи доступною для всіх верств населення.

Головною метою є створення умов для максимально зручного та швидкого доступу користувачів до знань, наукових праць, художніх текстів, енциклопедичних статей та інших літературних ресурсів. Для цього впроваджуються сучасні технології каталогізації, пошуку інформації, персонального доступу, що дозволяють користувачеві зручно знаходити потрібні матеріали.

Також завданням електронних бібліотек є розробка та підтримка бібліографічних баз даних, електронних каталогів та автоматизованих систем пошуку. Використання інтернет-технологій дозволяє створювати динамічні веб-платформи з інтеграцією гіпертекстових посилань.

Тому предметною сферою даної роботи є системи пошуку та перегляду енциклопедичної інформації про літературні твори й авторів. Онлайн-енциклопедії книг виконують такі функції як: збереження структурованої інформації про книги, авторів та жанри, забезпечення швидкого та зручного доступу до довідкової інформації про літературні твори; дотримання вимог щодо авторського права та ліцензій відкритого доступу. Таким чином, цифрові бібліотеки та енциклопедії сприяють розвитку культури читання, наукової комунікації та глобального доступу до знань, відповідаючи критеріям сучасного інформаційного суспільства.

1.2 Приклади платформ для розповсюдження літературних ресурсів

У сучасному цифровому середовищі існує достатня кількість платформ, які забезпечують розповсюдження літературних ресурсів серед широкої аудиторії. Такі платформи виконують важливу роль у популяризації читання, доступі до художньої та наукової літератури, а також у підтримці освітніх ініціатив [6][5]. Розглянемо аналоги таких платформ:

Електронні бібліотеки. Системи, що забезпечують доступ до електронних копій книг, статей, наукових робіт та інших документів. Гарними прикладами електронних бібліотек є Google Books, Open Library та Project Gutenberg.

Основна ідея Google Books полягає в тому, щоб зробити надруковані книги доступними для читання та пошуку в інтернеті, забезпечивши користувачам необмежений доступ до знань. Open Library в свою чергу дозволяє читати книги онлайн (ті, що перебувають у відкритому доступі), брати електронні копії книг напрокат та шукати старі або рідкісні видання. Платформа налічує мільйони назв книг різними мовами, включно з художньою, науковою та технічною літературою. Project Gutenberg також зосереджується на книгах, що знаходяться у вільному доступі і є першою спробою оцифрувати літературні твори та зробити їх доступними для всіх. У бібліотеці зібрано понад 60 000 творів різних жанрів, які можна книги можна завантажити в різних форматах (ePub, Kindle, HTML, текст).

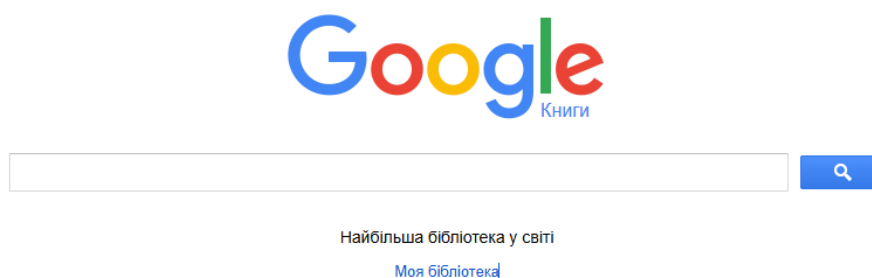


Рисунок 1.1 – Інтерфейс Google Books [4]

Кафедра інтелектуальних інформаційних систем
Розробка вебзастосунку онлайн-бібліотеки

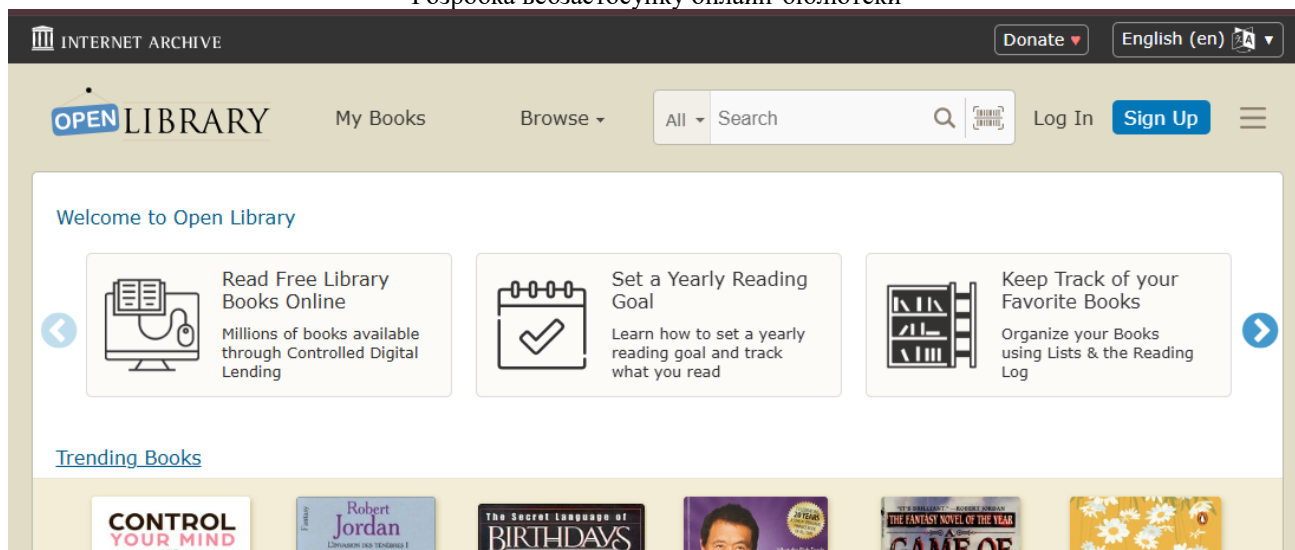


Рисунок 1.2 – Інтерфейс Open Library [2]

Онлайн-енциклопедії. Вебплатформи, що надають структуровану інформацію про книги, авторів та літературні напрями. Найвідомішим прикладом такої платформи є Wikipedia, де існують окремі розділи, присвячені літературі. Українська Вікіпедія в свою чергу була заснована 30 січня 2004 року.



Рисунок 1.3 – Інтерфейс Wikipedia

Також платформами для розповсюдження літературних ресурсів є платформи електронного самвидаву. Сервіси, де автори можуть самостійно публікувати свої твори. Прикладами платформ електронного самвидаву є Wattpad – це онлайн-спільнота для письменників, в якій користувачі можуть розміщувати статті, фанфіки, вірші та романи в режимі онлайн або через додаток; Amazon Kindle Direct Publishing - платформа для видання електронних книг від Amazon.com, запущена в листопаді 2007 року, одночасно з першим пристроєм Amazon Kindle. Вона дозволяє авторам і видавцям публікувати свої книги в магазині Amazon Kindle.



Рисунок 1.4 – Лого Wattpad [30]



Рисунок 1.5 – Amazon Kindle Direct Publishing [29]

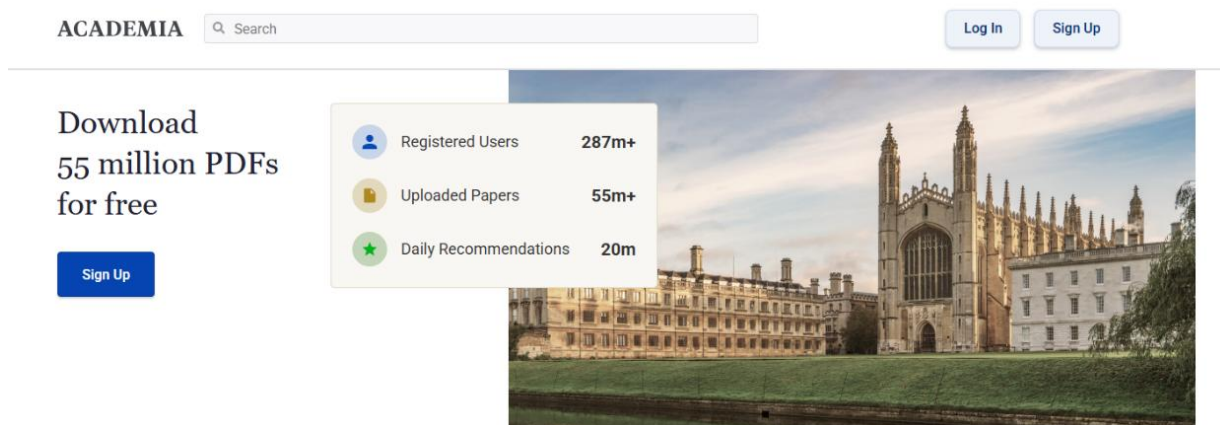


Рисунок 1.6 – Інтерфейс Academia.edu [28]

Академічні репозитарії. Спеціалізовані платформи для зберігання та поширення наукових праць, серед яких і літературознавчі дослідження. Прикладами є ResearchGate та Academia.edu, платформи, які орієнтовані на публікацію наукових робіт та доступ до досліджень інших користувачів. Також платформа ResearchGate дозволяє ставити запитання та отримувати відповіді від інших дослідників у певній галузі.

Завдяки розвитку платформ для розповсюдження літературних ресурсів користувачі отримують можливість зручно і швидко знаходити необхідну літературну інформацію.

1.3 Електронна енциклопедія. Аналіз українських онлайн-бібліотек

Цифрова трансформація світового суспільства XXI століття глибоко вплинула на всі галузі знань, зокрема й на енциклопедистику. Вона спричинила стрімке поширення та зростання популярності електронних енциклопедичних ресурсів. Сьогодні електронна форма енциклопедистики частково витісняє традиційну, яка передбачала створення багатотомних друкованих видань і дотримання лаконічного стилю подачі матеріалу [1]. З'явилася значна кількість нових енциклопедичних проєктів, що існують виключно у цифровому форматі та функціонують лише онлайн. Завдяки цифровим технологіям забезпечуються доступність, відкритість і гнучкість у процесі здобуття знань.

Електронна енциклопедія - це цифрове довідкове джерело, яке містить інформацію з однієї або кількох галузей знань і практики. Матеріали викладено у формі статей із можливим використанням зображень, анімацій, аудіо- чи відеофрагментів. Такі енциклопедії мають зручну систему навігації та дають змогу здійснювати переходи між статтями за допомогою гіперпосилань, будучи розміщеними на електронних носіях або у мережі Інтернет.

Звернемо увагу на електронні енциклопедії як один із різновидів літературного

ресурсу. В інтернет-просторі існують тисячі онлайн-бібліотек, що містять мільйони електронних книг. В останні роки в Україні з'явилося кілька проєктів, які пропонують літературу українською мовою в цифровому форматі. Серед найвідоміших і найпопулярніших українських онлайн-бібліотек можна назвати такі ресурси, як «УкрЛіб», «Українська література», «Читанка» та «УКРЛІТ.ORG» [9].

«УКРЛІТ.ORG» це публічна онлай-бібліотека художньої літератури, що розпочала свою діяльність у 2005 році зі збирання творів декількох авторів. Відтоді проєкт активно розвивався, поступово доповнюючись новими книгами. Основними напрямками бібліотеки є класична українська література, тлумачні словники, матеріали для транслітерації, а також інші ресурси, призначені для школярів, студентів, учителів і широкої аудиторії читачів.

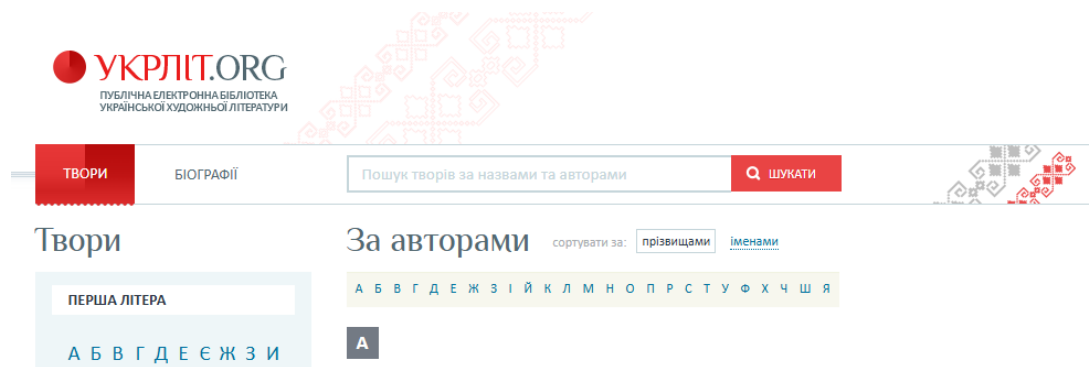


Рисунок 1.7 – Онлайн-бібліотека «УКРЛІТ.ORG» [27]

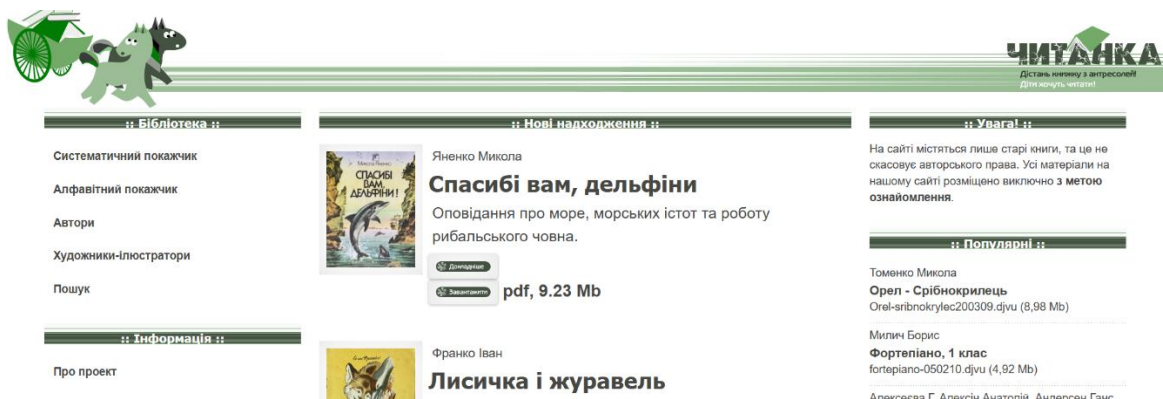


Рисунок 1.8 – Онлайн-бібліотека «Читанка» [26]

Дитяча публічна онлайн-бібліотека «Читанка» спеціалізується на ілюстрованих книгах українською мовою. Автори проекту сканують старі дитячі книги з картинками, щоб поділитися ними з сучасними дітьми.

Щоб знайти потрібну книгу, слід скористатися авторським, абетковим або систематичним каталогом. Усі книги в «Читанці» можна завантажити в єдиному форматі DJVU, тому читати їх можна лише на планшетах або рідерах, які підтримують цей стандарт електронних книг.

«УкрЛіб» це онлайн-бібліотека, що була заснована у 2000 році. Зараз вона є найбільшою електронною бібліотекою української літератури. Мета проєкту це популяризація української мови й культури, а також виховання любові до літератури серед молодого покоління.

На головній сторінці бібліотеки розміщено рубрику «Книга дня», де щоденно пропонується новий твір для ознайомлення. Також є розділ «Видатні діячі української літератури», що містить біографічні відомості та добірки найважливіших творів українських письменників. Книги доступні для читання онлайн або для завантаження у зручних форматах: DOCX, RTF, HTML, EPUB, MOBI та PDF.

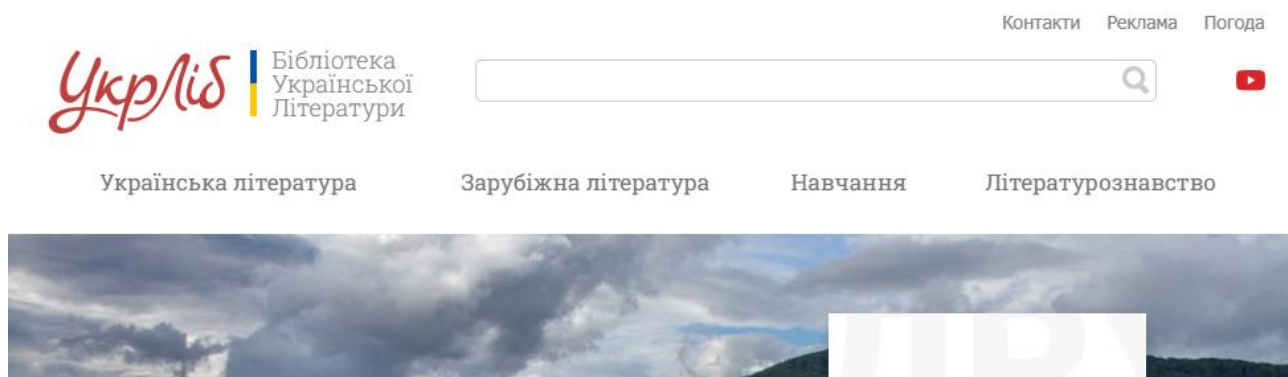


Рисунок 1.9 – Онлайн-бібліотека «УкрЛіб» [25]

Електронна бібліотека «Українська література» орієнтована передусім на учнів, студентів та всіх, хто цікавиться українською класикою. Цей ресурс надає відкритий

доступ до найвідоміших творів української літератури, з якими читачі здебільшого знайомляться ще зі шкільної програми.

Для зручності користувачів усі автори на сайті згруповані в алфавітному порядку. Обравши потрібного письменника, читач потрапляє на сторінку, де зібрано всі матеріали, пов'язані з його творчістю.

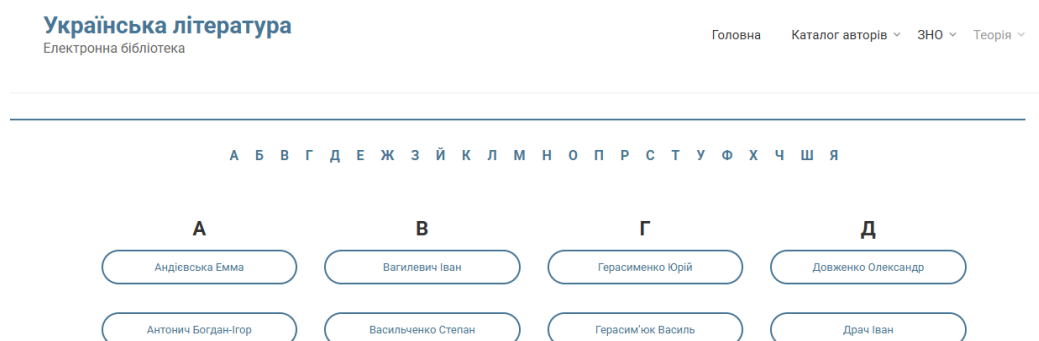


Рисунок 1.10 – Онлайн-бібліотека «Українська література» [24]

Книги з «Українська література» доступні як для онлайн-читання на сайті, так і для завантаження у форматах DOCX, FB2, PDF та DJVU — з можливістю читати їх на смартфоні, планшеті або електронному рідері.

Також можна виділити підрозділ «ЗНО» (якого немає в інших представлених бібліотеках). Там студенти зможуть знайти аналізи книг, характеристики, скорочені версії творів та літературну теорію.

1.4 Постановка задачі. Вимоги до програмної реалізації

Для розробки вебзастосунку онлайн-енциклопедії спрямованої на літературу, що дозволяє зручно переглядати та шукати інформацію про літературні твори, необхідно обрати відповідні інструменти та технології для розробки

Система має реалізовувати механізми пошуку за ключовими словами. Вебінтерфейс повинен бути інтуїтивно зрозумілим, практичний дизайн.

Для реалізації даної задачі можна використати мову програмування, JavaScript, а також HTML/CSS, щоб візуалізувати та стилізувати вебзастосунок.

Зважаючи на характер системи, доречно застосувати монолітну архітектуру, оскільки вона краще підходить для невеликого за масштабом застосунку з централізованою логікою. Доцільно буде обрати базу даних, враховуючи необхідність перегляду структурованої інформації про книги та авторів. У якості інтегрованого середовища розробки можна обрати Visual Studio Code.

Висновки до розділу 1

Цифрові технології наразі змінили традиційні підходи до збереження, поширення та споживання літературної інформації. Інтеграція бібліотек у цифровий простір забезпечує швидкий і зручний доступ до знань, сприяє популяризації читання, розвитку наукової комунікації та збереженню літературної спадщини. Електронні енциклопедії також стали важливою складовою сучасного інформаційного середовища.

Огляд міжнародних платформ засвідчує глобальну тенденцію до створення відкритого доступу до літературних і наукових ресурсів. Це відкриває нові можливості для освіти, саморозвитку та наукової діяльності.

Аналіз українських онлайн-бібліотек, таких як «УКРЛІТ.ORG», «Читанка», «УкрЛіб» та «Українська література», показав, що вони успішно виконують функцію популяризації української літератури, надаючи безкоштовний доступ до якісних літературних матеріалів. Особливо важливо, що багато з них орієнтовані на молодь, школярів і студентів, що сприяє вихованню нових поколінь читачів.

Таким чином, розробка електронних бібліотек та платформ для розповсюдження літературних ресурсів є актуальною в сучасному світі.

2 ВИБІР ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ВЕБЗАСТОСУНКУ ОНЛАЙН-БІБЛІОТЕКИ

2.1 UseCaseDiagram сайту

Для початку розробки вебзастосунку онлайн-енциклопедії, орієнтованої на літературні твори, необхідно визначити структуру сайту. Інтерфейс системи повинен бути зрозумілим, функціональним і візуально привабливим. Зазвичай діаграма прецедентів створюється на ранній етапах розробки застосунку.

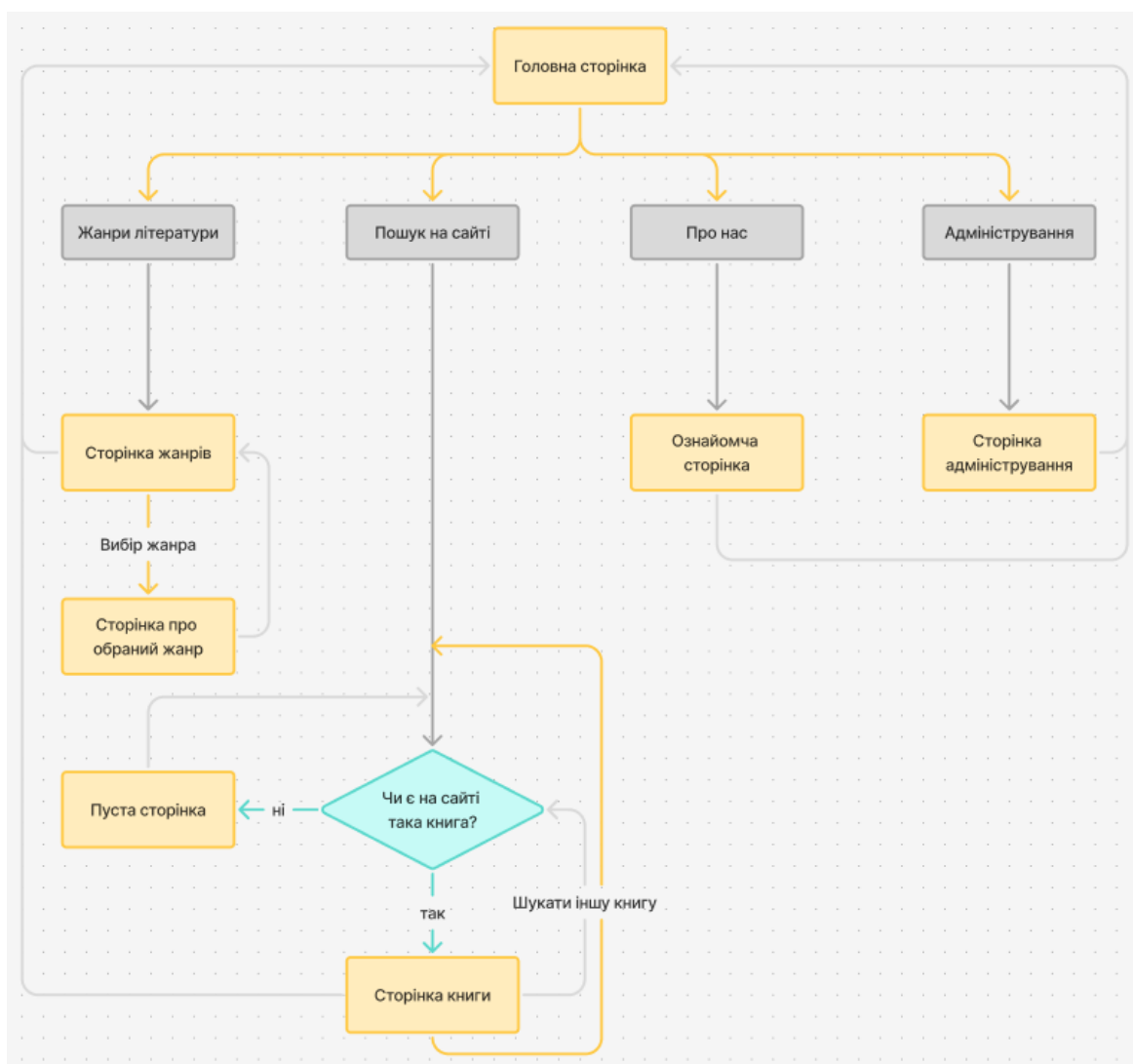


Рисунок 2.1 – UseCaseDiagram сайту

За визначенням, UseCaseDiagram – це діаграма, яка показує функції системи та взаємодію цих функцій з користувачами. Вона показує, як властивості системи бачать користувачі з зовнішньої точки зору. Грубо кажучи, говорить про те, що повинен робити певний вебзастосунок, і складається з основних елементів, які розробник хоче, щоб вона включала.

Для створення UseCaseDiagram (рис. 2.1) доцільно використати програми для створення UI/UX-дизайну, такі як Figma. Враховуючи основну тему застосунку, а саме надання інформації про літературні твори, система повинна включати такі сторінки:

- головна сторінка сайту;
- сторінка з інформацією про книгу, знайдену користувачем;
- сторінка з жанрами літератури;
- сторінка з інформацією про жанр, обрану користувачем;
- пуста сторінка при невдалому пошуку;
- ознайомча сторінка;
- сторінка адміністрування.

Use Case – це послідовність дій (сценарій), за якою користувач взаємодіє із застосунком для виконання конкретного завдання, передбаченого його функціональністю. Такий підхід дозволяє краще зрозуміти, як поводитиметься застосунок у певних ситуаціях, а також допомагає передбачити дії користувача [12].

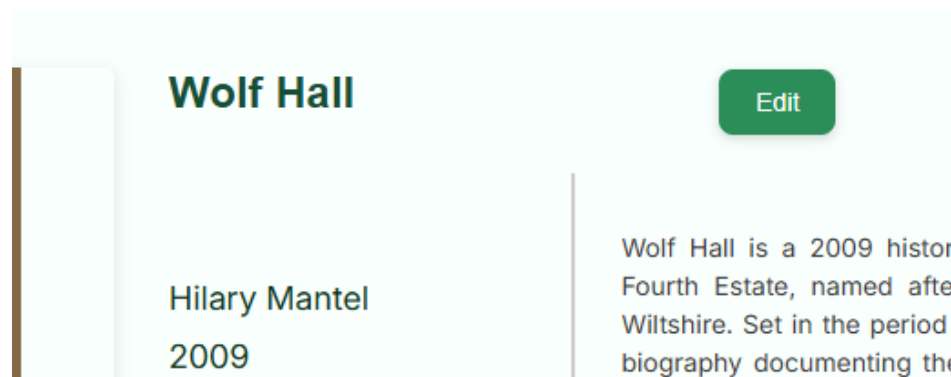


Рисунок 2.2 – Режим editor

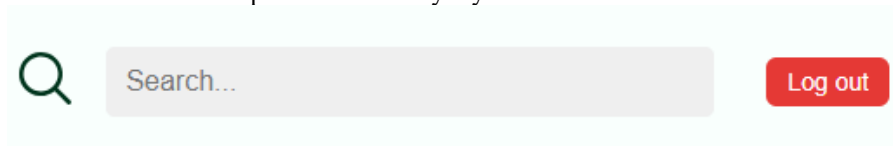


Рисунок 2.3 – UseCaseDiagram сайту (editor)

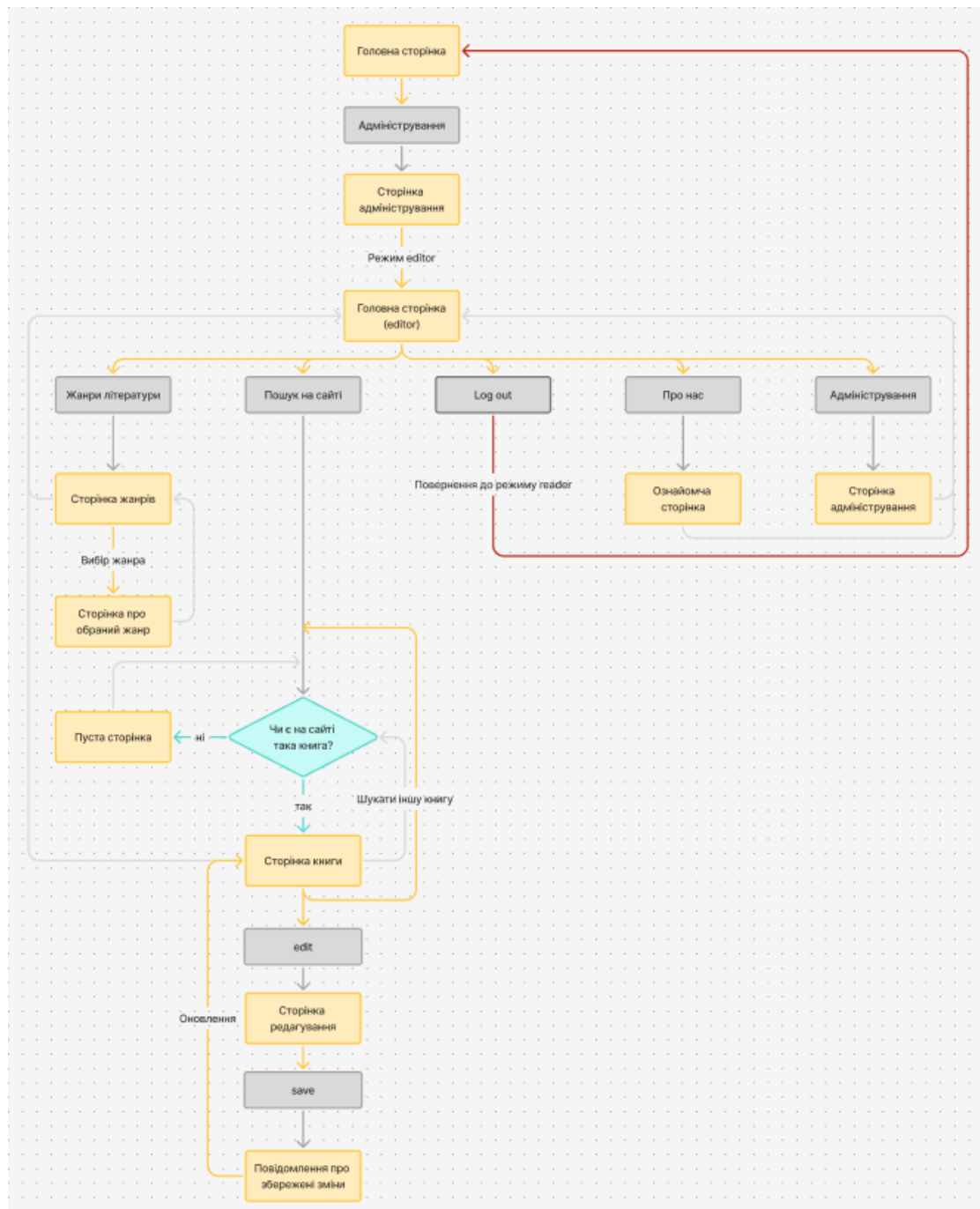


Рисунок 2.4 – UseCaseDiagram сайту (editor)

Стандартний сценарій: Користувач заходить на сайт. За замовчуванням він має статус reader. На головній сторінці для користувача з статусом reader є можливість перейти на ознайомчу сторінку, на загальну сторінку жанрів, на сторінку кожного жанру та користуватися пошуком на сайті. Користувач з статусом reader може шукати та переглядати книги, які є на сайті. З кожної сторінки сайту користувач може повернутися на головну сторінку сайту або на інші сторінки і також перейти на сторінку авторизації.

Editor сценарій (рис. 2.4): Користувач заходить на сайт. За замовчуванням він має статус reader, тому з головної сторінки користувач переходить на сторінку авторизації. На цій сторінці знаходиться форма авторизації. Користувач авторизується на сайті як editor. У користувача з статусом editor є можливість перейти на ознайомчу сторінку, на загальну сторінку жанрів, на сторінку кожного жанру та користуватися пошуком на сайті. Користувач з статусом editor може шукати, переглядати, а також редагувати книги, які є на сайті (рис. 2.2). З кожної сторінки сайту користувач може повернутися на головну сторінку сайту або на інші сторінки і також перейти на сторінку авторизації. На головній сторінці сайту користувач з статусом editor має можливість вийти з цього режиму (рис. 2.3).

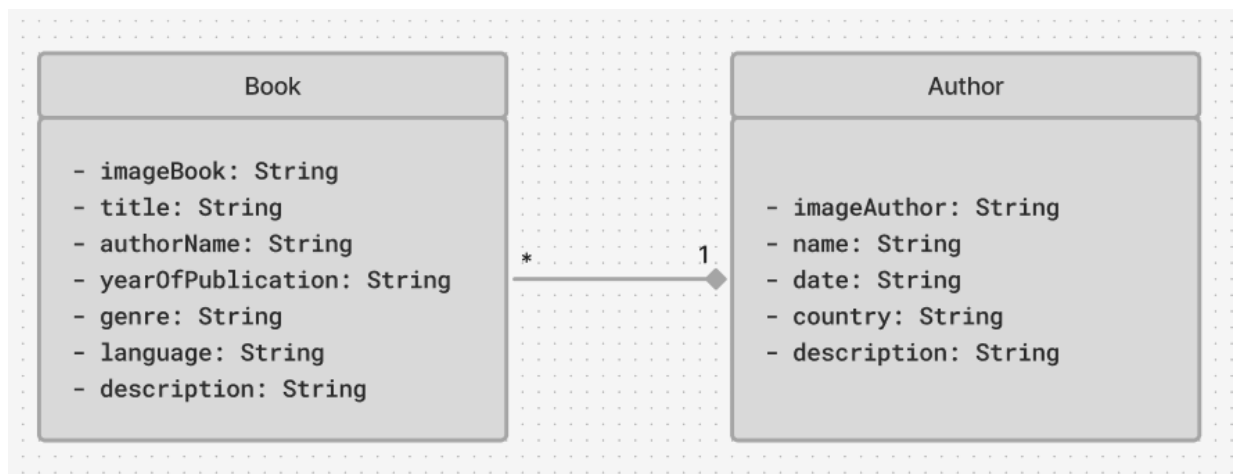


Рисунок 2.5 – Діаграма класів (book page)

Діаграма класів - це графічне зображення класів, які входять до складу системи, а також зв'язків між ними. Вона слугує інструментом для моделювання структури програмного забезпечення та візуалізації взаємозв'язків між окремими класами. Використання такої діаграми дозволяє заздалегідь виявити й уникнути потенційних помилок і труднощів під час розробки.

2.2 Технології для створення вебзастосунку. Інструменти для роботи

Фронтенд-технології – це інструменти, мови програмування, утиліти та методи, що впливають на створення тих частин веб-сайтів і додатків, з якими безпосередньо взаємодіють користувачі [10]. Ці інструменти покращують робочий процес веб-розробки, а також вони можуть бути корисними для конкретних вимог проекту. Фронтенд-технології розвиваються дуже швидко – деякі зникають, але інші революціонізують спосіб створення веб-сайтів та додатків, надаючи розробникам інструменти для створення захопливих та динамічних користувацьких інтерфейсів. Від основних мов HTML, CSS та JavaScript до передових фреймворків, таких як React та Svelte, варіанти різноманітні. Ключ до успіху полягає у виборі правильних технологій з широкого спектру доступних варіантів [13].

Можна визначити кілька категорій фронтенд-технологій (рис. 2.3):

- фронтенд-фреймворки: попередньо створені бібліотеки коду, що забезпечують структуру та компоненти багаторазового використання, що дозволяють розробникам уникнути винаходження велосипеда щоразу, коли вони створюють простий веб-сайт;
- практики розробки: методи та підходи, які визначають, як має виконуватися фронтенд-розробка. Це охоплює все: від методів адаптивного дизайну до методологій управління проектами, які допомагають фронтенд-розробникам послідовно підходити до своєї роботи;

- утиліти для кодування: інструменти, що покращують сам процес розробки.

До них належать редактори коду, інструменти розробника браузера та, наприклад, виконавці завдань, які автоматизують повторювані завдання;

- мови програмування: основні будівельні блоки фронтенд-розробки. Вона включає добре встановлений список мов розробки, які суттєво не змінилися протягом останнього десятиліття.

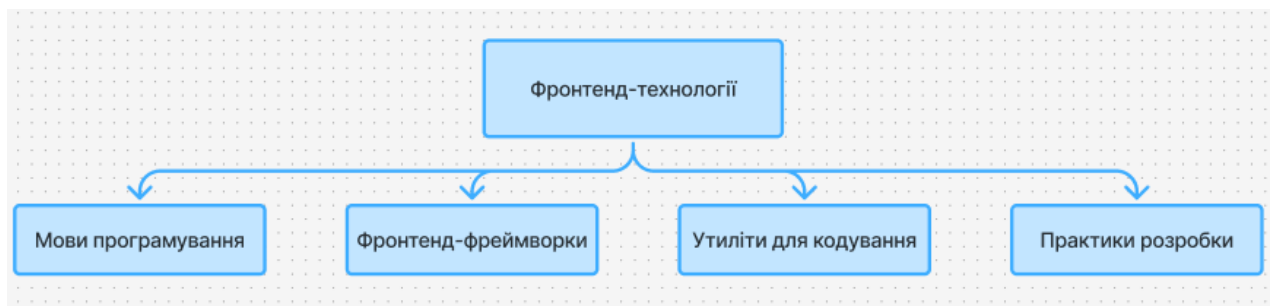


Рисунок 2.3 – З чого складаються фронтенд-технології

2.3 Visual Studio Code

Visual Studio Code (рис. 2.4) – це редактор вихідного коду, який підтримує роботу з багатьма мовами програмування, зокрема C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust. Він побудований на базі технології Electron, що використовується для створення вебдодатків на Node.js і функціонує за допомогою рендеринг-движка Blink.

З коробки VS Code забезпечує базову підтримку більшості мов, включаючи підсвічування синтаксису, автоматичне визначення пар дужок, згортання блоків коду та використання шаблонів коду. Можливість роботи з додатковими мовами забезпечується через безкоштовні розширення, які можна встановити за потреби.

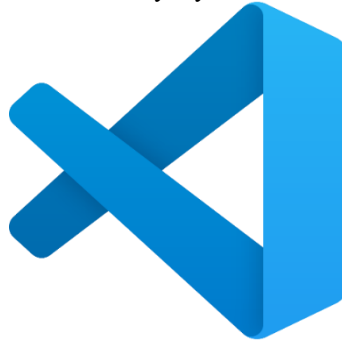


Рисунок 2.4 – Visual Studio Code

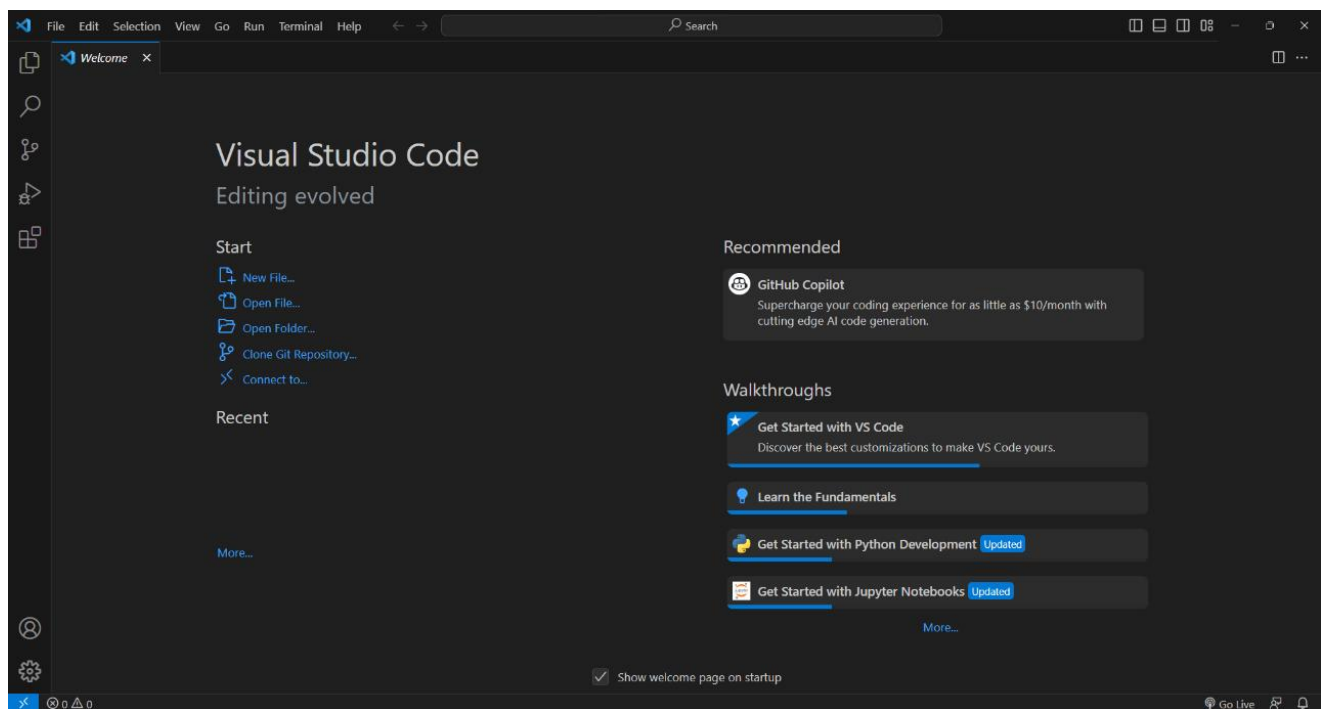


Рисунок 2.5 – Інтерфейс Visual Studio Code

Станом на 2025 рік більшість професійних розробників використовують цей редактор коду під час процесу розробки. Останні оновлення цього редактора зосереджені на покращенні продуктивності та розширенні функцій кодування за допомогою штучного інтелекту. Редактор тепер пропонує покращену інтеграцію з хмарними середовищами розробки та розширені інструменти для віддаленої співпраці.

2.4 Figma

Figma [16] (рис. 2.6) – це інструмент для роботи з векторною графікою, який дає змогу створювати макети, елементи інтерфейсу, ілюстрації, зображення, проєкти та інші креативні матеріали. Платформа доступна як у вигляді вебзастосунку, так і у форматі десктопної програми. Figma користується широкою популярністю серед веб-і UI/UX-дизайнерів, SMM-фахівців, авторів stories-контенту та інших спеціалістів у сфері digital.



Рисунок 2.6 – Figma

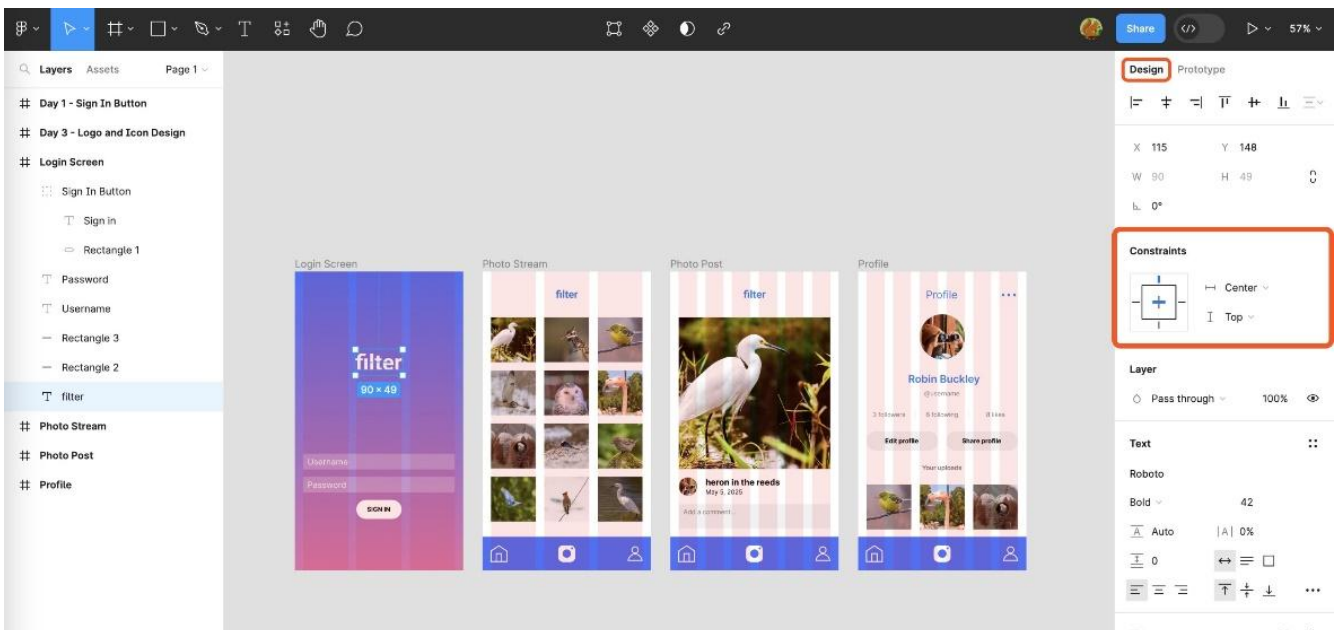


Рисунок 2.7 – Інтерфейс Figma

Figma [14] була створена для реалізації макетів і прототипів UI/UX. В розробці вебзастосунку Figma потрібна для безпосереднього створення дизайну цього вебзастосунку, а також вимірювання відступів, кольорів, шрифтів тощо. Вона також надає розробникам та дизайнерам зручний і наочний простір для комунікації навколо макетів. Figma має безліч переваг для роботи, включно з можливістю роботи в режимі реального часу, спільною роботою, доступністю та зручністю використання. Вона дає змогу створювати компоненти, спрощуючи процес дизайну та повторного використання елементів інтерфейсу.

2.5 HTML, CSS та JavaScript

Мови програмування є основою будь-якого процесу розробки вебзастосунку. Основні складові це HTML, CSS та JavaScript все ще утворює вичерпний список мов, необхідних фронтенд-розробникам. Однак вони не статичні. Регулярно вводять нові функції та покращують синтаксис, розширюючи можливості розробників створювати більш динамічні та адаптивні веб-інтерфейси [23].

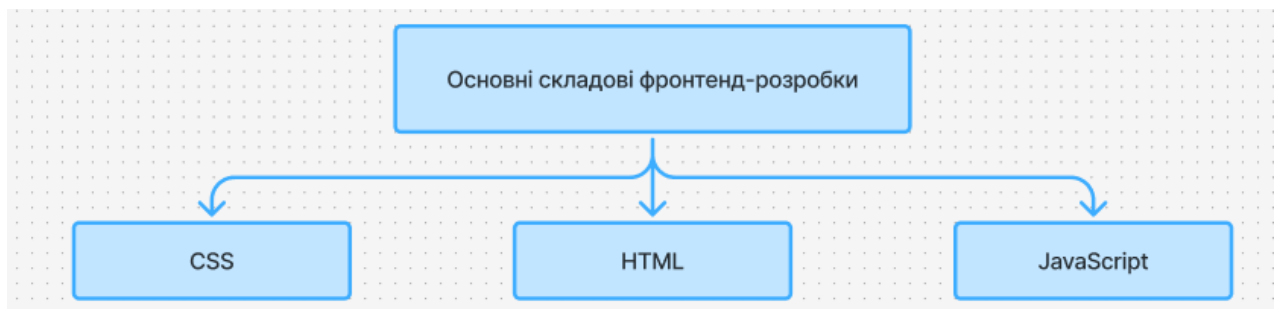


Рисунок 2.8 – Мови програмування в фронтенд-розробці

HTML (HyperText Markup Language) - це стандартизована мова розмітки, яка використовується для створення клієнтської частини веб-продукту. Процес розробки інтерфейсу починається саме з розмітки сторінки за допомогою HTML. Основне призначення цієї мови - визначення структури вебсторінки, тобто організація її

основних елементів і змісту. За допомогою HTML (рис. 2.9) прописується, які елементи будуть присутні на сторінці: абзаци, заголовки, зображення, списки, таблиці, форми тощо. Кожен елемент розмічається тегами, які вказують браузеру, як відображати контент. HTML забезпечує семантичну структуру, роблячи контент зрозумілим.

Мова гіпертекстової розмітки формує структуру веб-сторінок. Найпростіший спосіб побачити – це скористатися елементом «Inspect» у браузері. Він визначає вміст і розташування елементів. Останні оновлення HTML в основному зосереджені на семантичних елементах. Це робить веб-сторінки доступнішими та легшими для розуміння пошуковими системами одночасно.



Рисунок 2.9 – HyperText Markup Language



Рисунок 2.10 – Cascading Style Sheets [11]

CSS (Cascading Style Sheets). [11]HTML – це каркас сторінки, в чистому вигляді має дуже обмежений набір інструментів, що не дозволяє реалізувати ті чи інші дизайнерські та функціональні ідеї. Тому за вигляд та розміщення елементів на вебсторінці відповідає CSS (рис. 2.10). Cascading Style Sheets – мова стилю вебсторінок, що написані мовами розмітки даних. Частіше за все, CSS застосовується для оформлення документів, створених за допомогою HTML, XML або XHTML. Вона дає змогу задавати кольори, шрифти, розміри елементів, відступи та інші параметри оформлення.

Однією з основних переваг CSS є здатність розділяти зміст вебсторінки та її оформлення. Таке розмежування покращує сприйняття й доступність інформації, надає більше можливостей для керування зовнішнім інтерфейсом, а також сприяє структурі й зручнішому представленню даних [13].

Використання CSS дає:

- відображати документ в різних стилях;
- додати адаптивність. Здатність сайту підлаштовуватися під різні екрани, пристрої екрани та формати дисплея;
- скорочення часу завантаження вебсторінок завдяки винесенню стилістичних правил в окремий CSS-файл. У цьому випадку браузер спочатку завантажує лише структуру документа та вміст сторінки, тоді як стилі підвантажуються окремо, зберігаються в кеші та використовуються повторно;
- зручність у подальшому редагуванні дизайну сайту;
- додаткові можливості для візуального оформлення елементів.

CSS має простий синтаксис та використовує мало англійських слів для назв складових стилю. Стилi складаються з правил. Правила в CSS працюють по пріоритету. Це дозволяє отримати передбачуваний результат, коли до одного елемента, застосовуються декілька стильових правил.

Всі правила складаються з блоку оголошень та селектора. Всередині блоку

оголошень може знаходитися одне або кілька оголошень. Кожне CSS-правило починається з селектора, який визначає HTML-елементи, до яких це правило буде застосовано. У блоці оголошень задаються параметри відображення вибраних елементів, тобто визначаються їхні стилістичні властивості.

JavaScript. Окрім верстки та візуалу сторінок, в розробці вебзастосунку було використано JavaScript (рис. 2.11). Ця мова програмування допомагає перетворити статичні HTML-сторінки на інтерактивні вебзастосунки. З її допомогою реалізують різноманітний функціонал сайту.

JavaScript (JS) — це динамічна мова програмування з об'єктно-орієнтованою прототипною структурою. Найчастіше вона застосовується для створення інтерактивних елементів на вебсторінках, забезпечуючи взаємодію з користувачем на стороні клієнта, керування поведінкою браузера, асинхронний обмін даними з сервером, а також зміну структури та зовнішнього вигляду сторінки в реальному часі.

Крім того, JavaScript частково реалізує інші програмні парадигми - імперативну, функціональну та їхні похідні. Мова підтримує динамічну та слабку типізацію, автоматичне управління пам'яттю, прототипне наслідування та дозволяє працювати з функціями як з об'єктами першого класу.

JavaScript використовують для реалізації інтерактивності на вебсторінках, а також для створення прогресивних і односторінкових вебзастосунків із використанням таких фреймворків, як Vue.js, AngularJS і React. Крім того, ця мова застосовується для серверного програмування, розробки десктопних застосунків за допомогою Electron, а також мобільних — за допомогою Cordova чи React Native. JavaScript також може використовуватись для написання сценаріїв у прикладних програмах.



Рисунок 2.11 – Лого JavaScript [31]

Попри схожість у назвах, Java та JavaScript - це дві різні мови програмування. Вони мають відмінну семантику, хоча частково подібні у стилі іменування та деяких елементах стандартних бібліотек. Обидві мови походять від синтаксису мови C.

JavaScript є однією з найпоширеніших мов програмування в Інтернеті. Вона відіграє ключову роль у фронтенд-розробці, дозволяючи створювати динамічні й інтерактивні користувацькі інтерфейси. Розгалужена екосистема бібліотек і фреймворків робить розробку зручною та гнучкою, розширюючи можливості програмістів.

Висновки до розділу 2

У розділі 2 було розглянуто вибір технологій та інструментів, необхідних для створення вебзастосунку – онлайн-енциклопедії літературних творів. Приділено увагу побудові діаграми прецедентів (UseCaseDiagram), яка дозволила чітко окреслити функціональні можливості майбутнього вебзастосунку та взаємодію користувачів з ним.

Проаналізовано фронтенд-технології, які використовуються для реалізації подібного інтерфейсу. Розглянуто основні складові фронтенд розробки: JavaScript, HTML і CSS. Було наведено характеристику утиліт і редакторів коду, таких як Visual Studio Code, що відіграють важливу роль у створенні функціонального та зручного

вебзастосунку.

Окремо описано значення прогами Figma – сучасного інструменту для UI/UX-дизайну, що дозволяє створювати інтерфейс вебзастосунку з урахуванням зручності та естетики.

Отже, було обґрунтовано вибір технологій та інструментів, які дозволяють сформувати досконалу архітектуру вебзастосунку, що забезпечить зручність використання і підтримку подальшого його розвитку.

3 РОЗРОБКА ДИЗАЙНУ ВЕБЗАСТОСУНКУ ОНЛАЙН-БІБЛІОТЕКИ

3.1 Робота над дизайном. Вебдизайн

За планування та упорядкування контенту на вебсайті відповідає вебдизайн. Вебдизайн, що є поєднанням естетичних та функціональних елементів, – це тип цифрового дизайну, який визначає зовнішній вигляд вебсайту, такий як його кольори, шрифти, графіка та інтерфейс користувача [17]. Він постійно розвивається, включаючи мобільні додатки та дизайн інтерфейсу користувача, щоб задовольнити зростаючі потреби власників вебсайтів та відвідувачів.

Розробка вебдизайну вебзастосунку є важливою для залучення та утримання аудиторії та побудови позитивної репутації. Інтуїтивна навігація, логічне розташування елементів, швидке завантаження сторінки, адаптивність дизайну – все це невід'ємні складові високоякісного UX та UI [15].



Рисунок 3.1 – Складові UX та UI

З висновків другого розділу, зрозуміло, що розробка фронтенду також має на меті створення користувацького інтерфейсу вебзастосунку. Інтерфейс, будучи своєрідним посередником між програмним кодом та людиною, є ключовою частиною процесу взаємодії користувача з вебсайтом, використовуючи візуальні та аудіо елементи, такі як шрифти, значки, кнопки, анімацію та звуки тощо [21][22].

Гарний UI дотримується принципів дизайну, які дозволяють користувачам переміщатися по інтерфейсу та легко використовувати його за призначенням. Він визначає зручність навігації, доступність контенту та загальне враження від використання ресурсу. Продуманий та інтуїтивно зрозумілий інтерфейс забезпечує ефективну комунікацію між користувачем та функціоналом сайту, створюючи позитивний досвід для споживача.

Процес створення дизайну інтерфейсу вебзастосунку включає такі кроки:

- оцінка потреб користувача. Інтерфейс завжди має цільове призначення, і його дизайн зрештою повинен відповідати потребам цільової аудиторії вебсайту;
- вивчення існуючих функцій;
- створення ескізів. Це передбачає створення креслень низької якості, щоб намітити бачення того, як може виглядати кінцевий інтерфейс користувача;
- створення wireframe (каркасу). Цей крок включає створення більш детальних візуальних зображень інтерфейсів користувача, включаючи місце та спосіб збирання компонентів;
- створення компонентів дизайну;
- створення прототипу. Прототип інтерфейсу допомагає виявити проблеми та протестувати функціональність різних компонентів, щоб переконатися в їх безперебійній роботі перед впровадженням;
- реалізація інтерфейсу вебсайту.

Отже, проаналізувавши сферу та предмет дослідження кваліфікаційної роботи в

розділі 1, можна виділити мету створення вебзастосунку онлайн-енциклопедії. Враховуючи це, основним функціоналом вебсайту є надання інформації про обрані користувачем книги.

Відобразивши структуру сайту (рис. 2.1) та всі необхідні його складові, було визначено, що система повинна включати такі сторінки:

- головна сторінка сайту;
- сторінка з інформацією про книгу, знайдену користувачем;
- сторінка з жанрами літератури;
- сторінка з інформацією про жанр, обрану користувачем;
- пуста сторінка при невдалому пошуку;
- ознайомча сторінка;
- сторінка адміністрування.

Отже, інтерфейс системи повинен мати описані вебсторінки та бути зрозумілим, функціональним і візуально привабливим. Дизайн має підтримувати зручну навігацію, щоб забезпечити простоту сприйняття текстового контенту, що є ключовим для онлайн-енциклопедії.

Для реалізації цього завдання доцільно використати програми для створення UI/UX-дизайну, такі як Figma (рис. 2.5), яка дозволяє створити інтерактивні прототипи та дизайн-систему. Візуальне оформлення повинно відповідати принципам мінімалізму з дотриманням єдиного стилю та кольорової гами. Враховуючи основну тему застосунку, а саме надання інформації про літературні твори, основними акцентами дизайну повинні бути: зручність читання та швидкий доступ до основного функціоналу: пошуку, навігації за категоріями, відображення сторінок про книги.

3.2 Логотип

Логотип [18] – це емблема або графічний знак, або символ, що використовується для сприяння публічній ідентифікації та впізнаваності. Логотип може мати

абстрактний або фігуративний дизайн або містити текст назви, яку він представляє.

Так як застосунок є онлайн-енциклопедії, орієнтованої на літературні твори, її логотип та назва повинні викликати асоціації з літературною та культурною спадщиною. Для назви слова «улюблена бібліотека» було перекладено на італійську мову. В результаті отримано назву застосунку - «Libreria Preferita».

Переглянувши декілька аналогів логотипів для бібліотек, стало очевидним, що в більшості випадків для головного символу використовується книга. Зважаючи на це, було вирішено використати певну відмінність і обрати за основний символ письменника.



Рисунок 3.2 – Створеному логотип застосунку

Данте Аліг'єрі – видатний італійський поет, письменник і політик доби Відродження, а його головний твір «la Divina Commedia» є шедевром світової літератури. Зважаючи на це та на обрану мову для назви застосунку, саме його було обрано як центральну фігуру логотипу.

Логотип для вебзастосунку було створено в векторному графічному редакторі, Adobe Illustrator 2021 [19]. Це монохромний логотип у класичному стилі. У центрі розташовано силует Данте Аліг'єрі, референсом якого є його портрет атрибутований Сандро Боттічеллі. Силует оздоблений лавровим вінком.

Вся композиція обрамлена тонким кільцем, яке створює ефект герба - додає формальності та гармонійності створеному логотипу.

Назва застосунку написана під фігурою, уздовж нижнього краю кола. Текст виконаний декоративним шрифтом з невеликими лавровим листочками з обох боків. Логотип виконаний у сірому кольорі, але для загального дизайну було обрано певну кольорову гаму, яка складається з 6 основних кольорів (рис. 3.3).

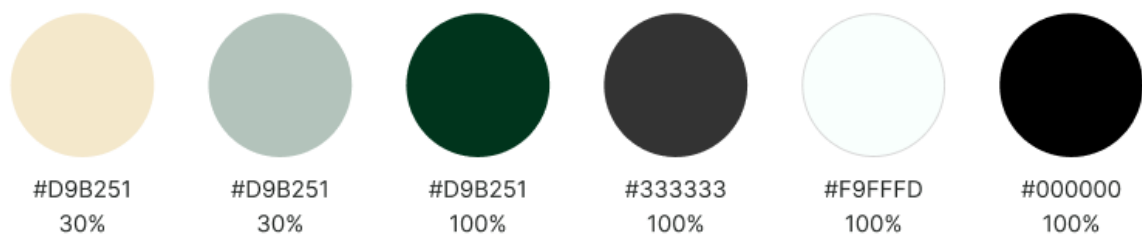


Рисунок 3.3 – Кольорова гама сайту

Мінімалізм [20] у веб-дизайні відзначається простотою, чіткістю та виваженістю композиції. Такий підхід передбачає створення інтерфейсу з мінімальною кількістю візуальних елементів, що сприяє концентрації уваги користувача на основній дії.

Ключові принципи мінімалістичного дизайну:

- усунення зайвих деталей та використання лише найнеобхідніших компонентів;
- відмова від всього непотрібного;
- кожен елемент виконує чітко визначену функцію;
- можливість зміни пароля у випадку його втрати;
- наявність достатнього вільного простору, що полегшує сприйняття контенту.

Лаконічний дизайн вебзастосунку сприяє швидкому доступу до важливої інформації, покращує зручність користування та підвищує загальну ефективність сайту.

3.3 Загальний дизайн. Сторінки сайту

Після створення логотипу, визначення з стилем та кольоровою гамою, було почато роботу над основними сторінками сайту, використовуючи, створені в розділі 2, UseCaseDiagram (рис. 2.1) та (рис. 2.4).

На головній сторінці сайту (рис. 3.4) у верхній частині знаходиться логотип і назва: "Your online library LIBRERIA PREFERITA". Далі на сторінці зображений невеликий блок, в якому знаходиться слоган та ознайомчий текст, який описує тему та функціонал онлайн-енциклопедії. Є навігаційне меню в header та пошуковий рядок. Напівпрозорий логотип доданий на фон текстового блоку для фірмового стилю. Головна сторінка сайту для editor містить в header також кнопку log out (рис. 3.5).

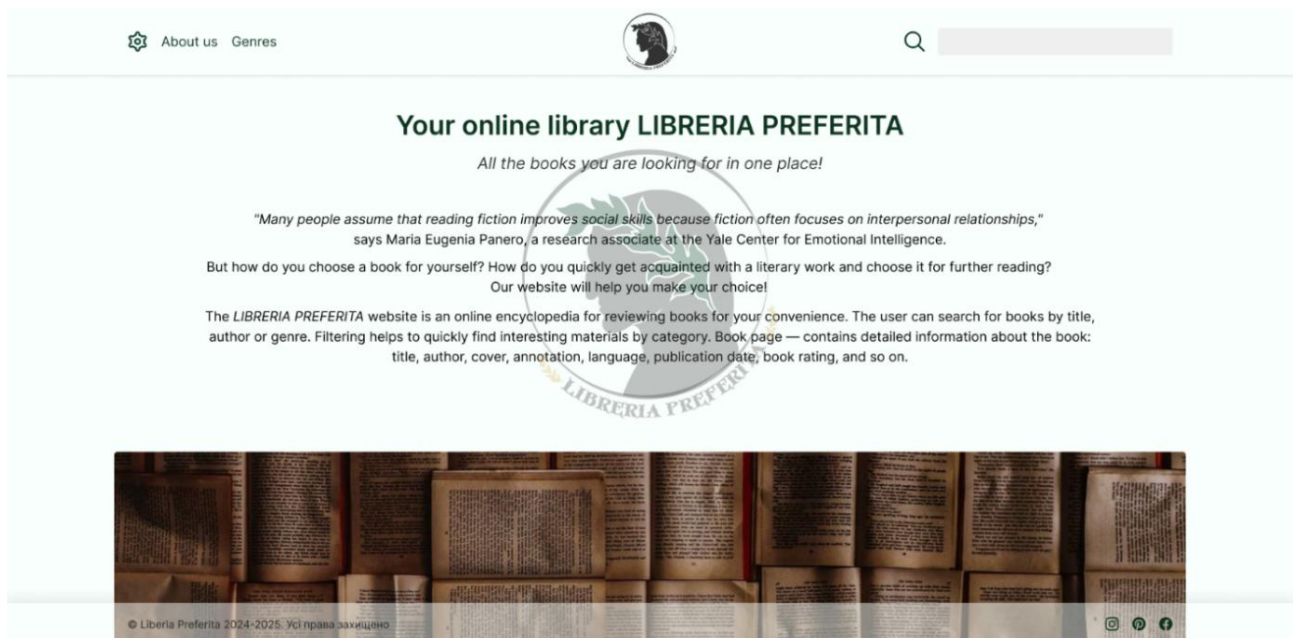


Рисунок 3.4 – Головна сторінка сайту

Кафедра інтелектуальних інформаційних систем
Розробка вебзастосунку онлайн-бібліотеки

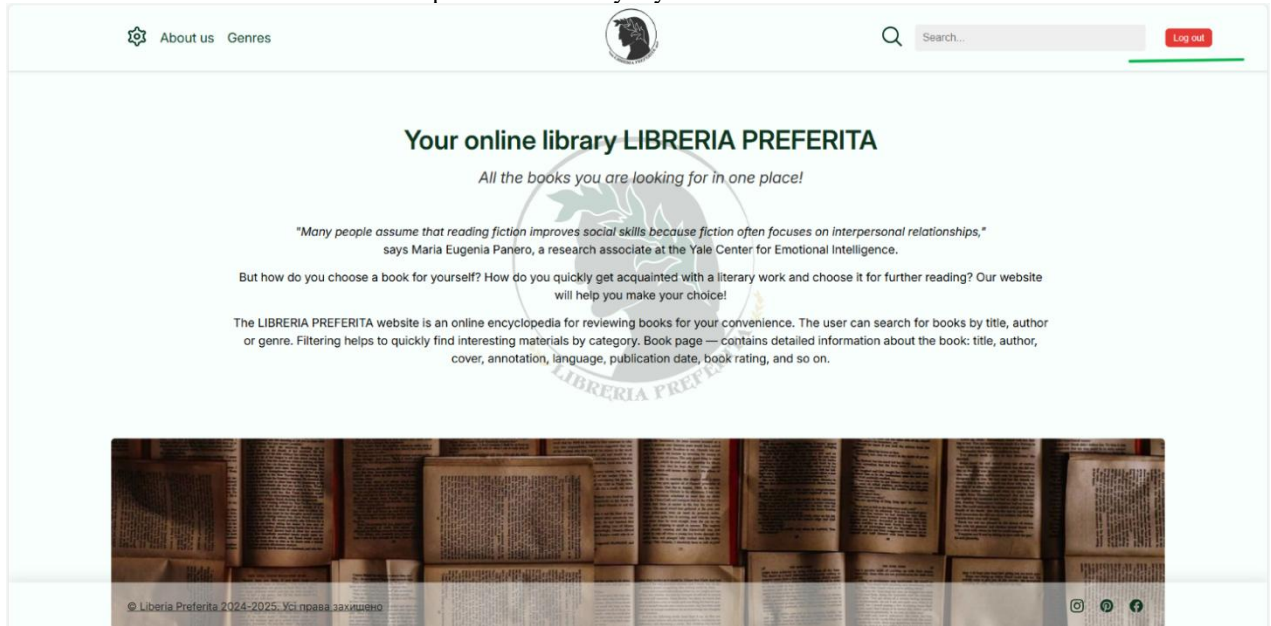


Рисунок 3.5 – Головна сторінка сайту для editor

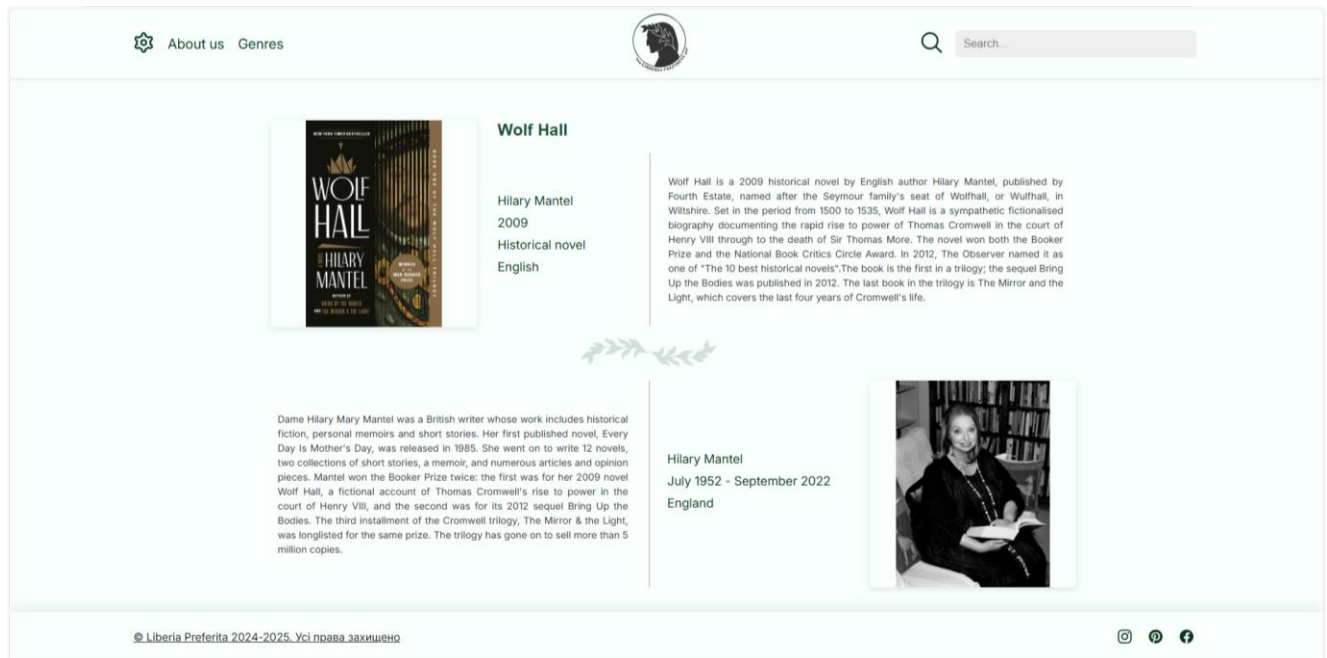


Рисунок 3.6 – Сторінка книги

Кафедра інтелектуальних інформаційних систем
Розробка вебзастосунку онлайн-бібліотеки

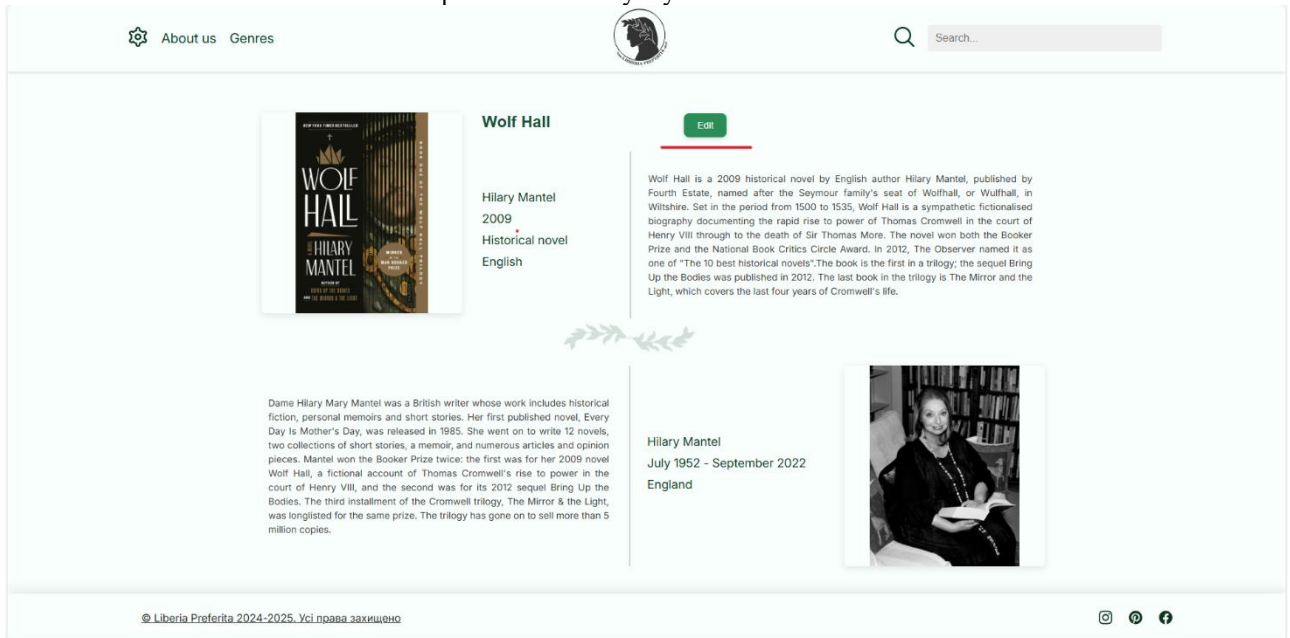


Рисунок 3.7 – Сторінка книги з можливістю редагування

Основні компоненти сторінки для інформації про книгу (рис. 3.6):

- картка книги: зображення обкладинки книги, коротка інформація (автор, рік публікації, видавництво, мова) та анотація до твору;
- картка автора: фото письменника, коротка інформація (Ім'я, дата народження, країна, мова) та біографічна інформація.

Для фону сторінки та для роздільного декоративного елементу було використано лаврові листя. Сторінка книги з можливістю в режимі editor містить кнопку edit (рис. 3.7).

Згідно UseCaseDiagram (рис. 2.1), вебсайт також надає користувачам можливість ознайомитись з літературними жанрами. Для цього було створено сторінку жанрів (рис. 3.8) на яку можна перейти з головної сторінки сайту.

Кафедра інтелектуальних інформаційних систем
Розробка вебзастосунку онлайн-бібліотеки

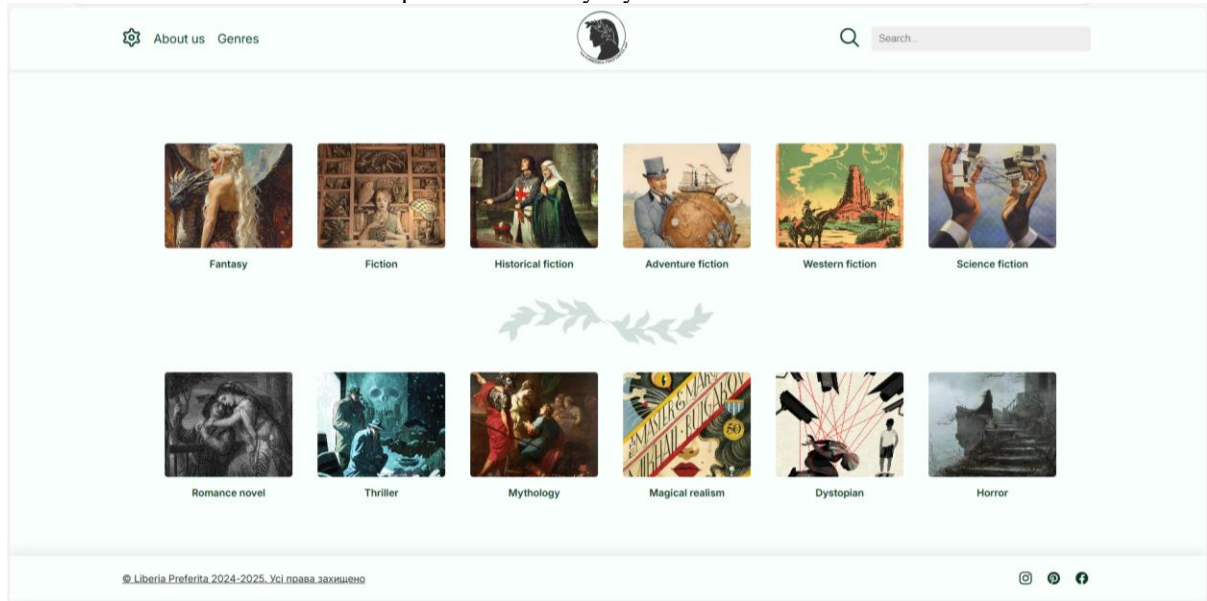


Рисунок 3.8 – Сторінка жанрів

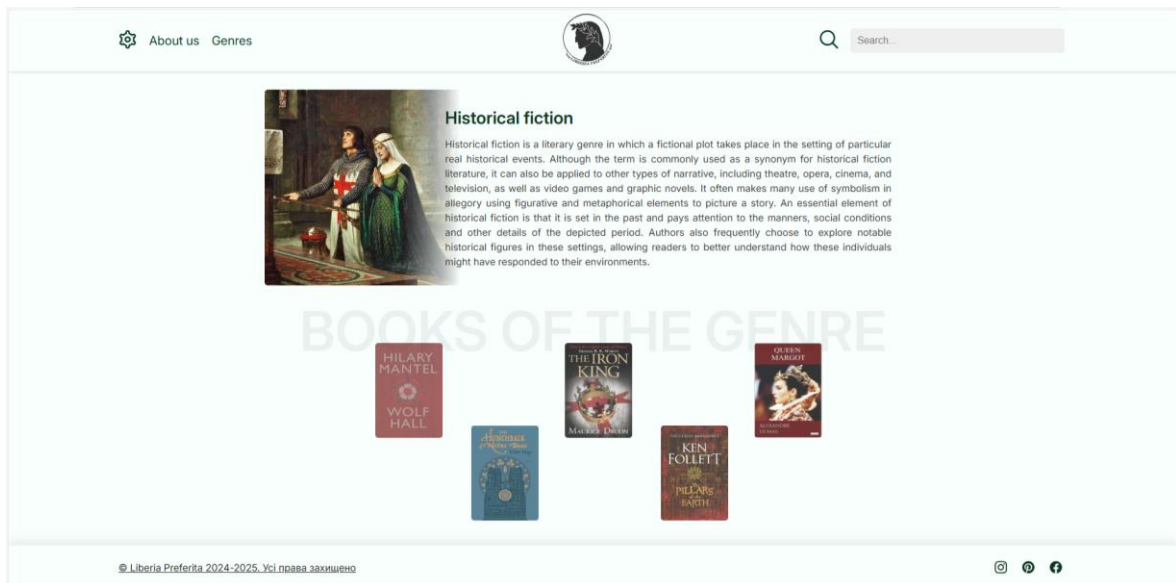


Рисунок 3.9 – Сторінка про жанр

Сторінка жанрів складається з 12 інтерактивних карток, кожна з яких представляє собою один з основних літературних жанрів:

– fantasy (фентезі). Література, дія якої відбувається в уявному всесвіті, часто без будь-яких місць, подій чи людей з реального світу;

- fiction (художня література). Будь-який творчий твір, головним чином будь-який оповідний твір, що зображує осіб, події чи місця, які є уявними або уявним чином. Таким чином, вигадані зображення не відповідають правдоподібності, історії чи фактам;
- adventure fiction (пригодницька проза). Вид художньої літератури, який зазвичай становить небезпеку або викликає у читача почуття захоплення;
- historical fiction (історична проза). Літературний жанр, у якому вигаданий сюжет розгортається в контексті реальних історичних подій;
- western fiction (західна художня література). Жанр літератури, дія якого відбувається на кордоні американського Дикого Заходу, зазвичай з кінця вісімнадцятого до кінця дев'ятнадцятого століття;
- science fiction (наукова фантастика). Жанр спекулятивної літератури, який стереотипно має справу з уявними та футуристичними концепціями: наука та технології, дослідження космосу, подорожі в часі, паралельні всесвіти та позаземне життя;
- romance novel (романтична проза). Художній роман, який зосереджується переважно на стосунках та романтичному коханні між двома людьми, зазвичай з емоційно задовільним та оптимістичним фіналом;
- thriller (трилер). Жанр художньої літератури з численними, часто перекриваючимися, піджанрами, включаючи кримінал, жахи та детектив;
- mythology (міфологія, або міфічна проза). Література, що черпає натхнення з тропів, тем і символіки міфів, легенд, фольклору та казок;
- magical realism (магічний реалізм). Жанр художньої літератури та мистецтва, який представляє реалістичний погляд на світ, включаючи магічні елементи, часто розмиваючи межі між спекуляціями та реальністю;
- dystopian (антиутопія художня). Піджанр спекулятивної художньої літератури, що досліджує крайні форми соціальних та політичних структур;

– horror (жахи). Жанр фантастики, метою якого є стурбувати, налякати або залякати аудиторію.

Сторінка кожного жанру (рис. 3.9) представляє собою інформацію про жанр, який обрав користувач, та 5 прикладів книг цього жанру наведених нижче.

Також була створена ознайомча сторінка (рис. 3.10), за структурою схожу на головну сторінку сайту. Перейти на цю сторінку користувач може за посиланням в header. Сторінка невдалого пошуку являє собою пусту сторінку з декоративним елементом та повідомленням про те, що книгу не було знайдено (рис. 3.11).

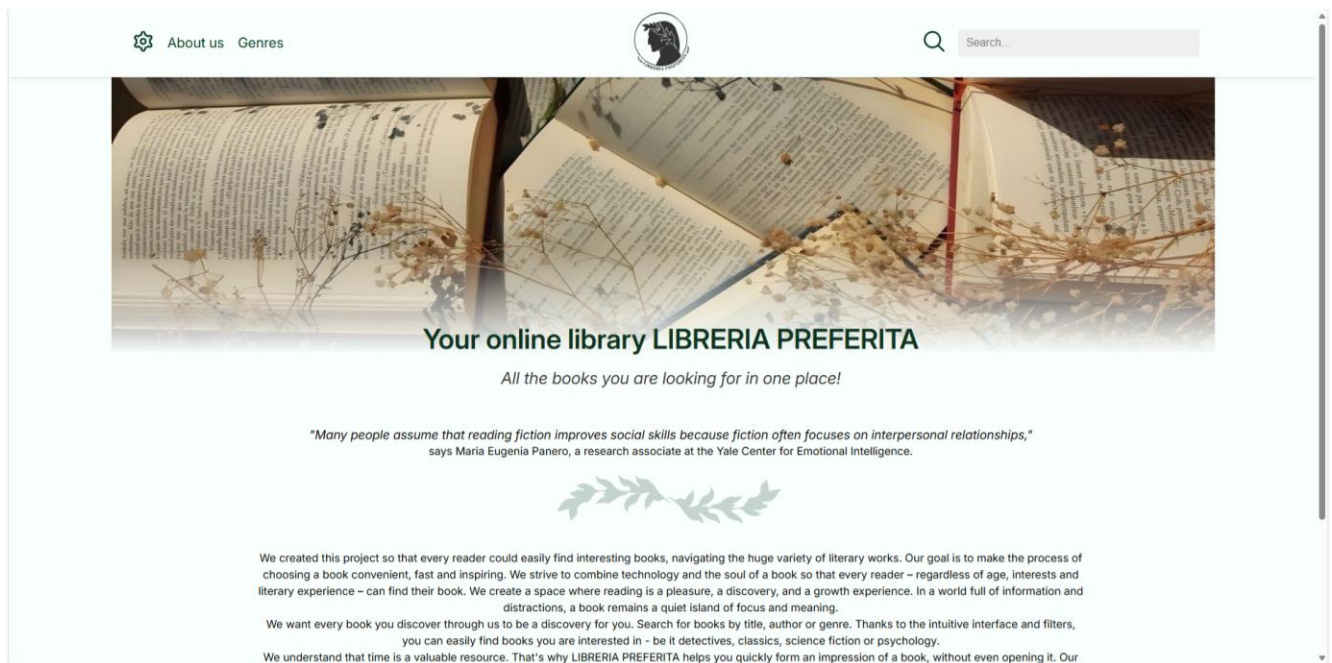


Рисунок 3.10 – Ознайомча сторінка

Останнім етапом в створенні інтерфейсу була розробка сторінок для авторизації та для редагування книги згідно з UseCaseDiagram (рис. 2.4). Переходячи за посиланням в header, користувач потрапляє на сторінку авторизації, де має змогу авторизуватися як editor, щоб мати можливість редагувати обрану книгу. На цій сторінці була створена невелика мінімалістична форма авторизації (рис. 3.12).

Сторінка для редагування книги являє собою сторінку з формою для

редагування інформації про обрану користувачем книгу (рис. 3.13). Вона виконана в тому ж стилі, що і форма для авторизації. Після збереження змін, користувач бачить невелике повідомлення (рис. 3.14). Воно було виконане в зеленому кольорі, щоб зробити його більш помітним.

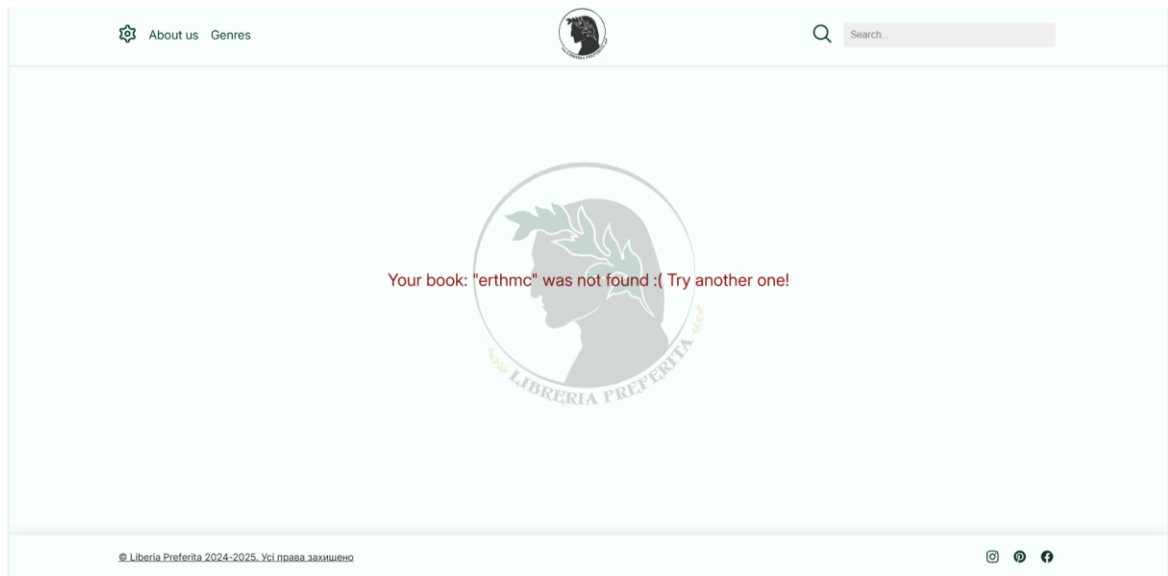


Рисунок 3.11 – Сторінка невдалого пошуку

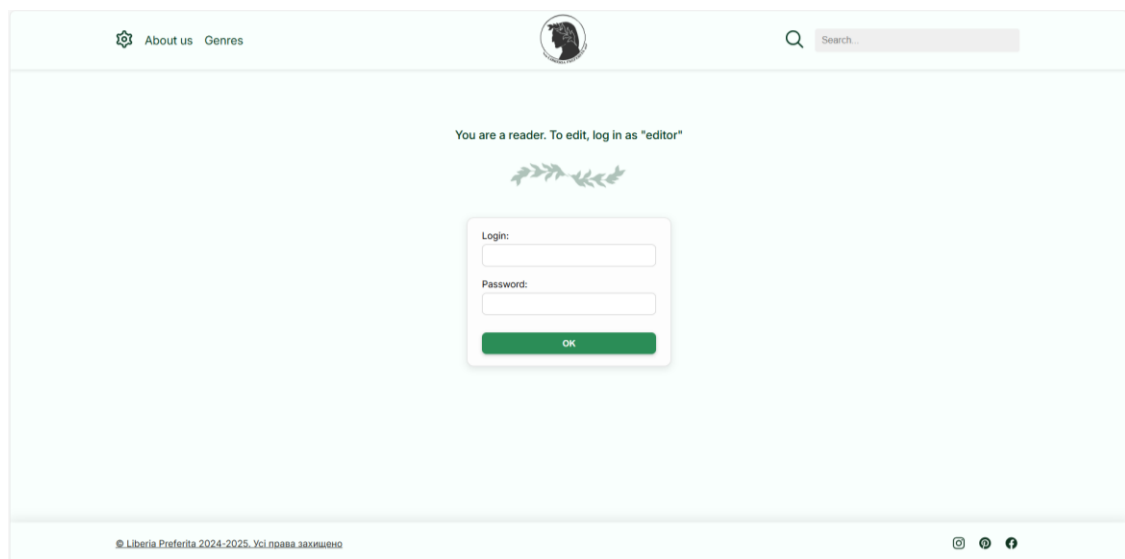


Рисунок 3.12 – Сторінка авторизації

Кафедра інтелектуальних інформаційних систем
Розробка вебзастосунку онлайн-бібліотеки

The screenshot shows a web application interface for editing a book. At the top, there is a navigation bar with a gear icon, 'About us', and 'Genres'. A search bar is on the right. The main content area features a central 'EDIT BOOK' form. The form has the following fields: 'Book title' (Wolf Hall), 'Author' (Hilary Mantel), 'Year' (2009), 'Genre' (Historical novel), 'Language' (English), 'Date' (July 1952 - September), 'Country' (England), and 'Author's name' (Hilary Mantel). Below these are two text areas: 'Book description' and 'Author info', both containing placeholder text. A green 'Save' button is at the bottom of the form. The footer of the page includes the copyright notice '© Libreria Preferita 2024-2025. Усі права захищено' and social media icons for Instagram, Pinterest, and Facebook.

Рисунок 3.13 – Сторінка для редагування книги



Рисунок 3.14 – Повідомлення про збережені зміни в книзі

Висновки до розділу 3

У розділі 3 було розроблено UI/UX-дизайн вебзастосунку онлайн-енциклопедії «Libreria Preferita». Постановка задачі включала розробку структури інтерфейсу з урахуванням зручності користування, візуальної привабливості та функціональності. Побудована в програмі Figma, UseCaseDiagram (рис. 2.1) дозволила визначити основні компоненти сайту.

Було приділено увагу створенню візуального стилю вебзастосунку: створений логотип вдало поєднує естетику класичної літератури з сучасним дизайном; ретельно

підібрана кольорова гама та декоративні елементи.

У результаті роботи над дизайном було створено візуально привабливий інтерфейс, який відповідає поставленим задачам, так як розроблений дизайн забезпечує зручне користування вебзастосунком.

За списком потрібних сторінок зазначених в розділі 2, було створено: головну сторінку сайту; сторінку, на якій буде збережена інформація про книгу, знайдену користувачем; сторінку з жанрами літератури; сторінку, на якій буде збережена інформація про жанр, обраний користувачем; пусту сторінку при невдалому пошуку; ознайомчу сторінку та сторінку адміністрування.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН-ЕНЦИКЛОПЕДІЇ

4.1 Етапи роботи над розробкою вебзастосунку

Процес створення вебсайту починається з постановки задачі. (Розділ 1.4). Чітке розуміння основного завдання майбутнього вебзастосунку – це важлива складова, від якої залежить успіх проєкту.

Етапи розробки сайту [21]:

- визначення теми та основної мети проєкту (розділ 1);
- постановка задачі (розділ 1);
- розробка дизайну вебзастосунку (розділ 3);
- верстка та програмування (розділ 4);
- тестування (розділ 4).

Визначення теми та основної мети проєкту є першим і надзвичайно важливим етапом, оскільки саме від цього залежить тип майбутнього ресурсу, особливості цільової аудиторії та ключові вимоги до сайту. Чітке усвідомлення мети й очікуваного результату дає змогу грамотно сформувати структуру проєкту та визначити необхідні етапи його реалізації. Усі ці аспекти детально обговорюються ще на початковій стадії розробки. Лише після встановлення головних пріоритетів можна переходити до наступних кроків.

Постановка задачі - це основа подальшої роботи над вебресурсом. На цьому етапі визначаються всі ключові параметри: структура (кількість сторінок, розділів, блоків, категорій), вимоги до дизайну, функціоналу, візуального та текстового контенту, а також технічні можливості проєкту.

Розробка дизайну вебзастосунку є невід’ємною складовою процесу створення сайту. На цьому етапі ідеї набувають конкретної форми у вигляді макетів. Дизайн охоплює не лише зовнішній вигляд сторінок, а й логічне розміщення елементів відповідно до принципів зручності користування (UI/UX) та технічних вимог. Після

завершення дизайну всіх сторінок створюються фінальні макети, які дозволяють внести необхідні корективи ще до початку фронтенд-розробки.

Верстка та програмування. Це етап, на якому вебсайт набуває функціональності та перетворюється на повноцінний інструмент. Тут використовуються основи мови HTML, застосовуються стилі CSS, а також додається програмна логіка. Завдання розробника — "оживити" сайт, забезпечивши його всіма необхідними функціями. У результаті дизайн перетворюється на інтерактивний інтерфейс, з яким будуть працювати користувачі.

Тестування - обов'язковий етап, що включає перевірку якості (QA) вебзастосунку. Його мета — гарантувати стабільну та безперебійну роботу ресурсу, а також забезпечити позитивний досвід користувача. Перевіряється правильність роботи функціональних елементів: форм, навігації, інтеграцій, а також швидкість завантаження на різних пристроях. Тестування зручності (usability testing) дозволяє визначити ефективність реалізації з точки зору користувача. Фахівці з QA виявляють проблемні місця в логіці чи інтерфейсі, які можуть знизити рівень задоволеності відвідувачів або вплинути на конверсію сайту.

4.2 Результат програмної реалізації. Тестування функцій вебзастосунку

Згідно з основними етапами розробки вебзастосунку, зазначеними вище у підрозділі 4.1, було перейдено до четвертого етапу. Верстка та програмування. Для створення вебзастосунку онлайн-енциклопедії спрямованої на літературу в розділі 1 було проаналізовано предметну сферу, наведено приклади платформ для розповсюдження літературних ресурсів, визначено основну аудиторію та зазначено постановку задачі з вимогами до програмної реалізації.

В розділі 2 було обрано технології для створення вебзастосунку та інструменти для роботи. Зроблені за двома основними сценаріями UseCaseDiagram, (рис. 2.1) та (рис. 2.4), допомогли швидко визначитися з кількістю сторінок та функціоналом сайту.

Це допомогло перейти до третього етапу – розробки дизайну вебзастосунку онлайн-бібліотеки. В розділі 3 було детально представлено вибір назви, розробку логотипу, вибір основного стилю та палітри кольорів тощо. Було розроблено всі сторінки вебзастосунку онлайн-бібліотеки та отримано візуально привабливий та зручний інтерфейс з інтуїтивно зрозумілим та практичний дизайном (рис. 3.4 - 3.13).

Розробку вебзастосунку біло почато з головної сторінки сайту (додаток А), яка була успішно реалізована за допомогою HTML та CSS (рис. 3.4). За структурою на головній сторінці `<head>` містить назву сторінки, містить метатеги, підключення шрифтів з Google Fonts, підключення стилів (style.css). `<body>` містить основні візуальні компоненти інтерфейсу: шапку сайту (header), головну частину (container) та підвал сайту (footer).

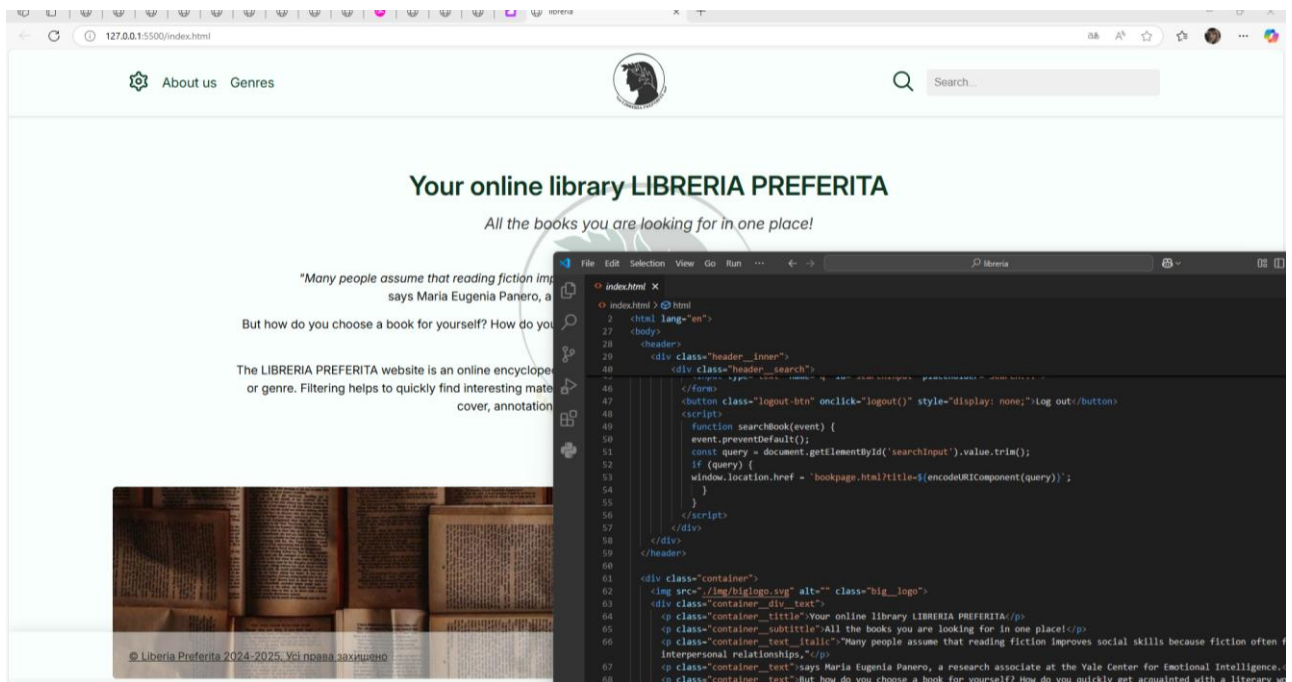


Рисунок 4.1 – Реалізація головної сторінки сайту

Шапка вебсайту (header) складається з посилань на сторінки Genres – загальної сторінки жанрів (рис. 3.8), About us – ознайомчої сторінки (рис. 3.10) а також з

іконки налаштувань, через яку користувач переходить на сторінку авторизації (рис. 3.12). Також header містить пошуковий блок, який складається з форми пошуку та з кнопки log out (яка відображається лише для користувачів в режимі editor (рис. 2.3, рис. 4.4)). Пошук реалізовано за допомогою JavaScript-функції searchBook(event), яка отримує введений користувачем запит (назву книги, ім'я автора або жанр, а потім перевіряє, чи існує книга з відповідною назвою (рис. 4.2)). Якщо книга знайдена, тоді користувач переходить на сторінку з обраною книгою. Якщо ні – на пусту сторінку (сторінку невдалого пошуку (рис. 3.11)). Для обробки табличних даних була використана бібліотека PapaParse, яка дозволяє ефективно парсити великі обсяги текстових даних у форматі об'єктів JavaScript.

```
function searchBook(event) {
    event.preventDefault();
    const query = document.getElementById('searchInput').value.trim().toLowerCase();

    if (!query) return;

    if (bookTitles.includes(query)) {
        smoothRedirect(`bookpage.html?title=${encodeURIComponent(query)}`);
    } else {
        window.location.href = `notfound.html?query=${encodeURIComponent(query)}`;
    }
}
```

Рисунок 4.2 – Реалізація пошуку

```
parsed.data
    .filter(book =>
        book["Book-Title"]?.toLowerCase().includes(searchedTitle) ||
        book["Book-Author"]?.toLowerCase().includes(searchedTitle) ||
        book["Author"]?.toLowerCase().includes(searchedTitle) ||
        book["Genre"]?.toLowerCase().includes(searchedTitle)
    )
```

Рисунок 4.3 – Фільтрація

```
function logout() {
```

```

localStorage.removeItem('role');
location.reload();
}
document.addEventListener("DOMContentLoaded", () => {
  if (localStorage.getItem('role') === 'editor') {
    document.querySelector('.logout-btn').style.display = 'inline-block';
  }
});

```

Рисунок 4.4 – Видимість кнопки log out для editor

Також на головній сторінці сайту (як і на всіх інших сторінках) є стилі та скрипт для реалізації плавного переходу з сторінки на сторінку, щоб забезпечити гарний візуал при користуванні вебзастосунком (рис. 4.5 – 4.6).

```

document.addEventListener("DOMContentLoaded", () => {
  document.body.classList.remove('fade-out');
});
function smoothRedirect(url) {
  document.body.classList.add('fade-out');
  setTimeout(() => {
    window.location.href = url;
  }, 300);
}

```

Рисунок 4.5 – Плавна поява сторінки

```

<style>
  body {
    opacity: 0;
    transition: opacity 0.3s ease;
  }
</style>
<script>
  document.addEventListener("DOMContentLoaded", () => {
    requestAnimationFrame(() => {
      document.body.style.opacity = '1';
    });
  }); </script>

```

Рисунок 4.6 – Плавна поява стилів

Також було створено сторінку з, обраною користувачем, книгою (рис. 3.6). На

сторінці так само реалізовано header та footer. Реалізований пошук, при якому, якщо користувач вводить назву книги, і така книга є на сайті – користувач отримує сторінку, порожній <div>, на якій наповнюється динамічним вмістом: блоком книги (image-book, book-title, book-author, year-of-publication, genre, language, book-description) та блоком автора (image-author, author, date, country, author-description). Якщо користувач вводить ім'я автора, і такий автор є на сайті – користувач отримує сторінку з усіма блоками книг та авторів, де ім'я автора співпадає з запитом користувача. Якщо користувач вводить жанр літератури, і такий жанр літератури є на сайті – користувач отримає сторінку з усіма блоками книги та автора, де жанр літератури співпадає з запитом користувача.

```
const isEditor = localStorage.getItem('role') === 'editor';
```

```
if (isEditor) {
    const editButton = document.createElement('button');
    editButton.textContent = 'Edit';
    editButton.className = 'edit-button';
    editButton.addEventListener('click', () => openEditForm(bookData, wrapper));
    bookTitleWrapper.appendChild(editButton);
}
```

Рисунок 4.7 – Видимість кнопки edit для editor

Якщо ж користувач авторизується як editor, то в блоці книги з'являється кнопка edit (рис. 2.2), за допомогою якої користувач може редагувати інформацію про знайдену ним книгу. Після натискання кнопки на сторінці відображається форма редагування (рис. 3.13). Для збереження змін, внесених користувачем, використовується localStorage. Після збереження змін користувач бачить повідомлення про успішне збереження (рис. 3.14). Після перезавантаження сторінки, збережені зміни відображаються.

```
function openEditForm(bookData, container) {
    const form = document.createElement('div');
```

```

    form.className = 'edit-form';
    form.innerHTML = `
    <h3>EDIT BOOK</h3>
    <div class="form-grid">
        <label>Book title: <input value="${bookData['Book-Title']}" id="editTitle"></label>
        <label>Author: <input value="${bookData['Book-Author']}"
id="editAuthorBook"></label>
        <label>Year: <input value="${bookData['Year-Of-Publication']}"
id="editYear"></label>
        <label>Genre: <input value="${bookData['Genre']}" id="editGenre"></label>
        <label>Language: <input value="${bookData['Language']}" id="editLang"></label>
        <label>Date: <input value="${bookData['Date']}" id="editDate"></label>
        <label>Country: <input value="${bookData['Country']}" id="editCountry"></label>
        <label>Author's name: <input value="${bookData['Author']}" id="editAuthor"></label>
    </div>
    <label>Book description:
        <textarea id="editBookDesc">${bookData['Book-Description']}</textarea>
    </label>
    <label>Author info:
        <textarea id="editAuthorDesc">${bookData['Author-Description']}</textarea>
    </label>
    <button id="saveChanges">Save</button>
    `;
    container.appendChild(form);
    document.getElementById('saveChanges').addEventListener('click', () => {
        bookData["Book-Title"] = document.getElementById('editTitle').value;
        bookData["Book-Author"] = document.getElementById('editAuthorBook').value;
        bookData["Year-Of-Publication"] = document.getElementById('editYear').value;
        bookData["Genre"] = document.getElementById('editGenre').value;
        bookData["Language"] = document.getElementById('editLang').value;
        bookData["Book-Description"] = document.getElementById('editBookDesc').value;
        bookData["Author"] = document.getElementById('editAuthor').value;
        bookData["Date"] = document.getElementById('editDate').value;
        bookData["Country"] = document.getElementById('editCountry').value;
        bookData["Author-Description"] = document.getElementById('editAuthorDesc').value;

        localStorage.setItem(`book_${bookData["Book-Title"]}`, JSON.stringify(bookData));
    });

```

Рисунок 4.8 – Форма редагування

```

localStorage.setItem(`book_${bookData["Book-Title"]}`, JSON.stringify(bookData));

```

Рисунок 4.9 – Збереження

```

const alertBox = document.getElementById('customAlert');
alertBox.style.display = 'block';

```

```

setTimeout(() => alertBox.style.opacity = '1', 10);

setTimeout(() => {
  alertBox.style.opacity = '0';
  setTimeout(() => alertBox.style.display = 'none', 400);
}, 3000);
});
}

```

Рисунок 4.10 – Вивід повідомлення про збереження змін

Для того, щоб надати користувачу можливість авторизуватися як editor, було створено сторінку авторизації (рис. 3.12). На цій сторінці реалізовано форму авторизації (рис. 4.11). Користувач вводить логін та пароль і система переходить в режим editor. Так було реалізовано основні функції застосунку.

```

document.getElementById('loginForm').addEventListener('submit', function(e) {
  e.preventDefault();
  const username = document.getElementById('username').value.trim();
  const password = document.getElementById('password').value.trim();

  if (username === 'editor' && password === '1234') {
    localStorage.setItem('role', 'editor');
    window.location.href = 'index.html';
  } else {
    alert('Error');
  }
});

```

Рисунок 4.10 – Форма авторизації

Під час тестування основні етапи створення сайту добігають кінця. Перед запуском сайт проходить перевірку, щоб впевнитися, що все працює стабільно: сторінки відкриваються коректно, інтеграції працюють швидко та ефективно, а сервер витримує навантаження. На цьому безпосередньо процес розробки веб сайту завершується. Тому після створення основних сторінок та реалізації функціоналу вебзастосунку онлайн-бібліотеки було перейдено до тестування.



Рисунок 4.11 – Некоректний запит користувача

Для перевірки роботи пошуку було здійснено тестування за декількома сценаріями. В першому користувач робить некоректний запит (рис. 4.11), в результаті якого отримує сторінку з невдалим пошуком (рис. 4.12).

Далі користувач вводить спочатку коректну назву книги (рис. 4.13), і отримує коректно відображену інформацію про цю книгу і автора на сторінці (рис. 4.14). Далі так само було перевірено пошук за автором та за жанром літератури, кожен раз отримуючи правильний результат. Таким чином було зроблено висновок, що пошукова система на сайті онлайн-бібліотеки працює коректно без помилок.



Рисунок 4.12 – Повідомлення на сторінці невдалого пошуку

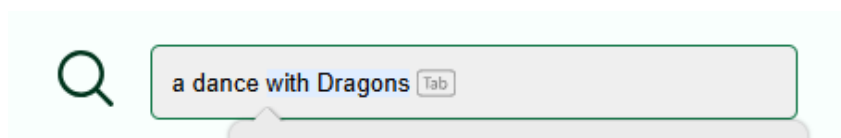


Рисунок 4.13 – Коректний запит користувача

Для перевірки роботи режиму editor було здійснено тестування форми авторизації та форми редагування. Спочатку користувач заходить на сторінку

Кафедра інтелектуальних інформаційних систем
Розробка вебзастосунку онлайн-бібліотеки
авторизації та авторизується як editor, вводить логін та пароль (рис. 4.15).

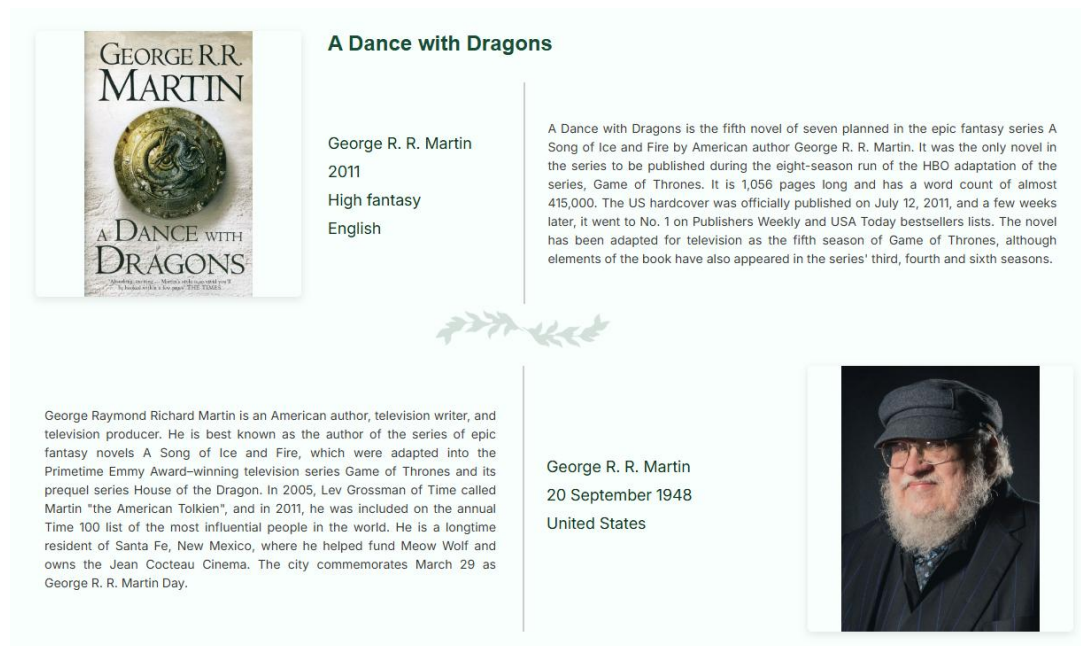


Рисунок 4.14 – Вдалий результат

You are a reader. To edit, log in as "editor"

Login:
editor

Password:
.....

OK

Рисунок 4.15 – Авторизація

Після авторизації на головній сторінці почала відображатися кнопка log out (рис. 4.16), а на сторінці книги кнопка edit (рис. 4.17). Це означає, що авторизація була

успішна. Користувач робить новий запит, перевіряючи роботу пошуку у режимі editor. Після знаходження потрібної книги було перевірено роботу форми редагування.



Рисунок 4.16 – Відображення кнопки log out

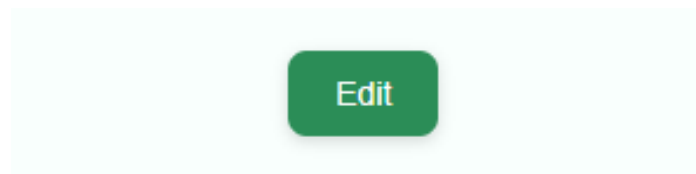


Рисунок 4.17 – Відображення кнопки edit

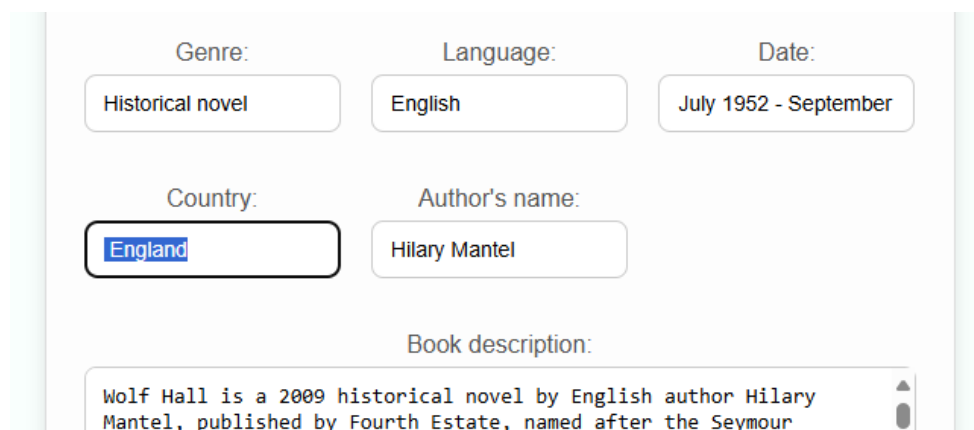


Рисунок 4.18 – Користувач змінює назву країни

Коли користувач натискає кнопку edit на сторінці книги відкривається форма редагування. Користувач змінює назву країни для перевірки (рис. 4.18). Після натискання кнопки save вебсайт показує користувачу повідомлення про збережені зміни (рис. 4.19), що означає функція збереження працює правильно. Користувач перезавантажує сторінку і бачить, що зміни успішно збережені (рис. 4.20 - 4.21).



Рисунок 4.19 – Повідомлення про збережені зміни

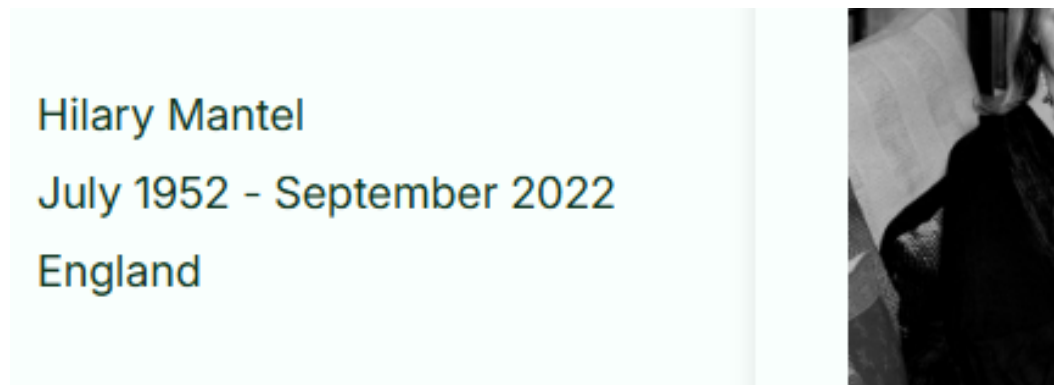


Рисунок 4.20 – Інформація до змін

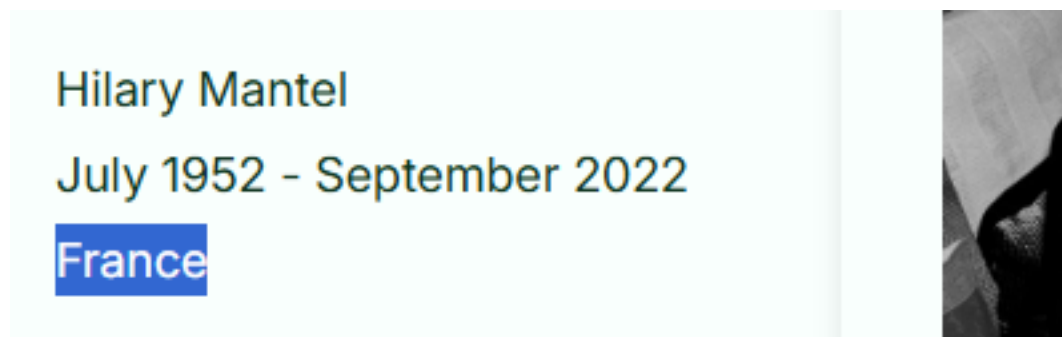


Рисунок 4.21 – Інформація з збереженими змінами

Після перевірки функції редагування, яка працює успішно, користувач натискає на кнопку log out і виходить з режиму editor. Таким чином можна побачити, що функція редагування та зберігання інформації, система пошуку, авторизація та режим editor на вебсайті працюють коректно. Отже весь функціонал вебзастосунку онлайн-бібліотеки успішно пройшов тестування та працює правильно.

Висновки до розділу 4

У розділі 4 було представлено процес реалізації вебзастосунку онлайн-енциклопедії, орієнтованої на літературні ресурси. Розробка пройшла всі необхідні етапи: від постановки задачі та розробки дизайну до безпосереднього програмування та тестування функціональності.

Було створено веб-застосунок з інтуїтивно зрозумілим інтерфейсом та зручним дизайном. Реалізовано основні сторінки сайту: головну сторінку, сторінку жанрів, ознайомчу сторінку, сторінку знайденої книги та сторінку авторизації. Функціонал пошуку книг, авторів та жанрів реалізовано за допомогою JavaScript та бібліотеки PapaParse для обробки даних. Також було реалізовано можливість входу в систему як editor, що дозволяє користувачам редагувати інформації про книги.

Тестування основного функціоналу показало, що роботи вебзастосунку є коректною та працює успішно. Таким чином, завершено програмну реалізацію онлайн-енциклопедії з базовим функціоналом.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено вебзастосунок онлайн-енциклопедії літературних творів «Libreria Preferita», який забезпечує зручний доступ до інформації про книги, авторів та жанри в інтерактивному форматі. Актуальність теми зумовлена зростаючими потребами суспільства у швидкому та доступному отриманні літературних ресурсів через цифрові платформи. Об'єктом дослідження став процес розробки такого вебзастосунку, а предметом – набір технологічних інструментів для його реалізації.

У першому розділі було наведено огляд сучасного стану українських та міжнародних платформ для розподілення літературних ресурсів. На основі аналізу сформульовано вимоги до майбутнього додатку. У другому розділі обґрунтовано вибір технологій для реалізації веб-додатку: HTML, CSS, JavaScript, а також бібліотека PapaParse для обробки даних у форматі CSV. Також було розглянуто інструменти для роботи такі як, Visual Studio Code (редактор коду) та Figma (застосунок для створення UI/UX дизайну). Було побудовано діаграму варіантів, яка допомогла визначити основні функції застосунку. У третьому розділі було створено макети інтерфейсу вебзастосунку, що відповідають сучасним вимогам до зручності та естетики. Було розроблено структуру сайту та дизайн для всіх необхідних сторінок. Особливу увагу було приділено гармонійному поєднанню гарного візуалу та простоти. У четвертому розділі, відповідно до раніше розробленого дизайну та структури вебзастосунку, була реалізована програмна частина. Передбачено функціонал пошуку книг та авторів, перегляду жанрів та можливості редагування. Тестування підтвердило стабільність та працездатність системи.

Отже, в результаті роботи над кваліфікаційною роботою було досягнуто поставлену мету – створено зручну, функціональну та естетично привабливу онлайн-енциклопедію літературних творів. Вебзастосунок може стати основою для подальшого розвитку. Наприклад: додавання нових функцій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Концепт і генеза поняття “онлайн енциклопедія” на сторінках вітчизняної наукової літератури. URL: <https://surl.li/pdcirh> (дата звернення: 25.04.2025).
2. *Open Library*. URL: <https://openlibrary.org/> (last accessed: 25.04.2025).
3. *Project Gutenberg*. URL: <https://gutenberg.org/> (last accessed: 25.04.2025).
4. *Books Google. Google*. URL: <https://books.google.com/> (last accessed: 25.04.2025).
5. Бібліотека XXI століття: перспективи та інновації. Матеріали Всеукраїнської науково-практичної конференції. URL: <https://lib.knukim.edu.ua/wp-content/uploads/2014/12/Z-tezy.pdf> (дата звернення: 25.04.2025).
6. Вісник Книжкової палати. Електронні бібліотеки як вагомий складник національного інформаційного простору України. URL: <http://visnyk.ukrbook.net/article/view/282684> (дата звернення: 25.04.2025).
7. Веб-енциклопедія як інформаційний засіб розвитку інноваційної компетентності педагога. URL: <https://lib.iitta.gov.ua/id/eprint/737185/1/Article%20153.pdf> (дата звернення: 25.04.2025).
8. Usage Analysis of a Digital Library. URL: file:///D:/Downloads/An_Analysis_of_Usage_of_a_Digital_Library.pdf (last accessed: 25.04.2025).
9. 10 найкращих онлайн-бібліотек з електронними книжками українською мовою. URL: <https://vylkove1.school.org.ua/10-najkraschih-onlajnbiibliotek-z-elektronnimi-knizhkami-ukrainskoju-movoju-15-07-16-05-05-2021/> (дата звернення: 25.04.2025).
10. Що таке фронтенд розробка: складові, етапи та технології? URL: <https://wezom.com.ua/ua/blog/chto-takoe-front-end-razrabotka> (дата звернення: 14.05.2025).

11. CSS. URL: https://css.in.ua/article/shcho-take-html_10 (дата звернення: 14.05.2025).
12. Use Case Diagrams: An Introduction. URL: <https://www.altkomsoftware.com/blog/use-case-diagrams-an-introduction/#:~:text=By%20definition%2C%20a%20UCD%20is,early%20stage%20of%20app%20development.> (last accessed: 14.05.2025).
13. Top 22 Latest Front-End Technologies to Use in 2025. URL: <https://elitex.systems/blog/top-latest-front-end-technologies> (last accessed: 14.05.2025).
14. Що таке Figma і навіщо вона потрібна? URL: <https://lemon.school/blog/cho-takoe-figma-i-zachem-ona-nuzhna> (дата звернення: 14.05.2025).
15. User interface (UI). URL: [https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI#:~:text=The%20user%20interface%20\(UI\)%20is,the%20appearance%20of%20a%20desktop.](https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI#:~:text=The%20user%20interface%20(UI)%20is,the%20appearance%20of%20a%20desktop.) (last accessed: 14.05.2025).
16. Figma. URL: <https://cases.media/en/article/sho-take-figma-ta-dlya-kogo-vona-potribna?srsId=AfmBOoqyhxYw96deqd8EweMIFM7P8Rg-4S1XmnNg7hnSaWhgRfhXxcLV> (дата звернення: 14.05.2025).
17. What is web design? A comprehensive guide. URL: <https://surl.li/hwghco> (last accessed: 14.05.2025).
18. Навіщо потрібен логотип? URL: <https://litoprint.com.ua/navischo-potriben-logotip> (дата звернення: 14.05.2025).
19. Where to put your logo on your website? URL: <https://www.one.com/en/online-marketing/logo-placement-website> (last accessed: 14.05.2025).

20. Мініمالізм у веб-дизайні
URL: <https://webtune.com.ua/statti/dyzajn/minimalizm-u-veb-dyzayni/> (дата звернення: 14.05.2025).
21. Етапи створення веб сайтів URL: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorenniya-veb-sajtiv/> (дата звернення: 14.05.2025).
22. 10 етапів створення сайту — від ідеї до запуску
URL: <https://wezom.com.ua/ua/blog/etapy-razrobotki-sajta> (дата звернення: 14.05.2025).
23. Інформаційні системи та технології в лісовому господарстві
URL: <https://www.uzhnu.edu.ua/uk/infocentre/get/83806> (дата звернення: 14.05.2025).
24. Українська література. Електронна бібліотека.
URL: <https://ukrclassic.com.ua/> (дата звернення: 14.05.2025).
25. Бібліотека української літератури “УкрЛіб”.
URL: <https://www.ukrlib.com.ua/> (дата звернення: 14.05.2025).
26. Дитяча публічна онлайн-бібліотека “Читанка”.
URL: <http://chytanka.com.ua/> (дата звернення: 14.05.2025).
27. Публічна електронна бібліотека української художньої літератури “УКРЛІТ.ORG” URL: <http://ukrlit.org/> (дата звернення: 14.05.2025).
28. *Academia*. URL: <https://www.academia.edu/> (last accessed: 14.05.2025).
29. *Amazon*. Amazon Kindle Direct Publishing.
URL: <https://www.amazon.com/stores/author/B008241EAQ> (last accessed: 14.05.2025).
30. *Wattpad*. URL: <https://www.wattpad.com/> (last accessed: 14.05.2025).
31. Wikipedia. Unofficial JavaScript logo by Chris Williams. URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення: 14.05.2025).

ДОДАТОК А

Лістинг коду головної сторінки

```
<script>
  document.addEventListener("DOMContentLoaded", () => {
    requestAnimationFrame(() => {
      document.body.style.opacity = '1';
    });
  });
</script>
</head>

<body>
  <header>
    <div class="header__inner">
      <div class="header__links">
        <a href="./admin.html" class="settings">
          
        </a>
        <a href="./about.html" class="header__text">About us</a>
        <a href="./genres.html" class="header__text">Genres</a>
      </div>
      <a href="./index.html" class="logo">
        
      </a>
      <div class="header__search">
        <a href="" class="settings">
          
        </a>
        <form class="search-form" onsubmit="searchBook(event)">
          <input type="text" name="q" id="searchInput"
placeholder="Search...">
        </form>
        <button class="logout-btn" onclick="logout()" style="display:
none;">Log out</button>
      <script>
        function searchBook(event) {
          event.preventDefault();
          const query =
document.getElementById('searchInput').value.trim();
          if (query) {
            window.location.href =
`bookpage.html?title=${encodeURIComponent(query)}`;
          }
        }
      </script>
```



```

        </div>
    </div>
</header>

<div class="container">
    
    <div class="container_div_text">
        <p class="container__tittle">Your online library LIBRERIA
PREFERITA</p>
        <p class="container__subtittle">All the books you are looking for
in one place!</p>
        <p class="container__text__italic">"Many people assume that
reading fiction improves social skills because fiction often focuses on
interpersonal relationships,"</p>
        <p class="container__text">says Maria Eugenia Panero, a research
associate at the Yale Center for Emotional Intelligence.</p>
        <p class="container__text">But how do you choose a book for
yourself? How do you quickly get acquainted with a literary work and
choose it for further reading? Our website will help you make your
choice!</p>
        <p class="container__text">The LIBRERIA PREFERITA website is an
online encyclopedia for reviewing books for your convenience.
        The user can search for books by title, author or genre.
Filtering helps to quickly find interesting materials by category.
        Book page – contains detailed information about the book:
title, author, cover, annotation, language, publication date, book
rating, and so on.</p>
    </div>
    
</div>
<div class="footer">
    <div class="footer__inner">
        <a href="" class="footer__text">© Liberia Preferita 2024-2025.
Усі права захищено</a>
        <div class="footer__links">
            <a
href="https://www.instagram.com/ksd_bookclub?utm_source=ig_web_button_s
hare_sheet&igsh=ZDNlZDc0MzIxNw==" class="footer__link">
                
            </a>
            <a href="https://ru.pinterest.com/" class="footer__link">
                
            </a>
            <a href="https://www.facebook.com/share/v/15hkv5J7jK/"
class="footer__link">
                
    </a>
</div>
</div>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/PapaParse/5.4.1/papaparse.m
in.js"></script>
<script>
    let bookTitles = [];
    fetch('./dataset/data.txt')
        .then(response => response.text())
        .then(csvText => {
            const parsed = Papa.parse(csvText, {
                header: true,
                skipEmptyLines: true
            });
            bookTitles = parsed.data.map(book => book["Book-
Title"].toLowerCase().trim());
        });
    document.addEventListener("DOMContentLoaded", () => {
        document.body.classList.remove('fade-out');
    });
    function smoothRedirect(url) {
        document.body.classList.add('fade-out');
        setTimeout(() => {
            window.location.href = url;
        }, 300);
    }
    function searchBook(event) {
        event.preventDefault();
        const query =
document.getElementById('searchInput').value.trim().toLowerCase();
        if (!query) return;
        if (bookTitles.includes(query)) {
            smoothRedirect(`bookpage.html?title=${encodeURIComponent(query)}`
);
        } else {
            window.location.href =
`notfound.html?query=${encodeURIComponent(query)}`;
        }
    }
</script>
<script>
    function logout() {
        localStorage.removeItem('role');
    }

```

```
location.reload();  
}  
document.addEventListener("DOMContentLoaded", () => {  
  if (localStorage.getItem('role') === 'editor') {  
    document.querySelector('.logout-btn').style.display = 'inline-  
block';  
  }  
});  
</script>  
</body>  
</html>
```

ДОДАТОК Б

Лістинг коду сторінки книги

```
<script>
  document.addEventListener("DOMContentLoaded", () => {
    requestAnimationFrame(() => {
      document.body.style.opacity = '1';
    });
  });
</script>
</head>
<body>
  <div class="wrapper">
    <header>
      <div class="header__inner">
        <div class="header__links">
          <a href="/admin.html" class="settings">
            
          </a>
          <a href="/about.html" class="header__text">About us</a>
          <a href="/genres.html" class="header__text">Genres</a>
        </div>
        <a href="#" class="logo"
onclick="smoothRedirect('/index.html')">
          
        </a>
        <div class="header__search">
          <a href="" class="settings">
            
          </a>
          <form class="search-form" onsubmit="searchBook(event)">
            <input type="text" name="q" id="searchInput" placeholder="Search...">
          </form>
        </div>
      </div>
    </header>
    <div class="main">
      <div class="book">
        <div class="book__info">
          <div class="book__title">
            <h1>Book Title</h1>
          </div>
          <div class="book__author">
            <p>Author Name</p>
          </div>
          <div class="book__description">
            <p>Book description text</p>
          </div>
        </div>
        <div class="book__img">
          
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

    </div>
</header>
<div class="container">
</div>
</div>
<div class="footer">
    <div class="footer__inner">
        <a href="" class="footer__text">© Liberia Preferita 2024-2025.
Усі права захищено</a>
        <div class="footer__links">
            <a
href="https://www.instagram.com/ksd_bookclub?utm_source=ig_web_button_s
hare_sheet&igsh=ZDNlZDc0MzIxNw==" class="footer__link">
                
            </a>
            <a href="https://ru.pinterest.com/" class="footer__link">
                
            </a>
            <a href="https://www.facebook.com/share/v/15hkv5J7jK/"
class="footer__link">
                
            </a>
        </div>
    </div>
</div>
<script>
fetch('./dataset/data.txt')
    .then(response => {
        const lastModified = response.headers.get('last-modified');
        const localLastModified =
localStorage.getItem('csv_last_modified');
        if (lastModified !== localLastModified) {
            localStorage.clear(); // Очистити кешовані книги
            localStorage.setItem('csv_last_modified', lastModified);
        }
        return response.text();
    })
    .then(csvText => {
        const parsed = Papa.parse(csvText, {
            header: true,
            skipEmptyLines: true
        });
        const container = document.querySelector('.container');

```

```

const urlParams = new URLSearchParams(window.location.search);
const searchedTitle = urlParams.get('title')?.toLowerCase();
const isEditor = localStorage.getItem('role') === 'editor';
parsed.data
  .filter(book =>
    book["Book-Title"]?.toLowerCase().includes(searchedTitle) ||
    book["Book-Author"]?.toLowerCase().includes(searchedTitle) ||
    book["Author"]?.toLowerCase().includes(searchedTitle) ||
    book["Genre"]?.toLowerCase().includes(searchedTitle)
  )
  .forEach(bookData => {
    const saved = localStorage.getItem(`book_${bookData["Book-
Title"]}`);
    if (saved) Object.assign(bookData, JSON.parse(saved));
    const wrapper = document.createElement('div');
    wrapper.className = 'book-author-wrapper';
    const bookCard = document.createElement('div');
    bookCard.className = 'book-card';
    const bookImg = document.createElement('img');
    bookImg.src = bookData["Image-Book"];
    bookImg.alt = `Обкладинка книги ${bookData["Book-Title"]}`;
    bookCard.appendChild(bookImg);
    const bookInfo = document.createElement('div');
    bookInfo.className = 'book-info';
    const bookTitleWrapper = document.createElement('div');
    bookTitleWrapper.className = 'book-title-wrapper';
    bookTitleWrapper.innerHTML = `

## ${bookData["Book- Title"]}</h2>`; if (isEditor) { const editButton = document.createElement('button'); editButton.textContent = 'Edit'; editButton.className = 'edit-button'; editButton.addEventListener('click', () => openEditForm(bookData, wrapper)); bookTitleWrapper.appendChild(editButton); } const bookMeta = document.createElement('div'); bookMeta.className = 'book-meta'; bookMeta.innerHTML = ` <p>${bookData["Book-Author"]}</p> <p>${bookData["Year-Of-Publication"]}</p> <p>${bookData["Genre"]}</p> <p>${bookData["Language"]}</p> `; const bookDescWrapper = document.createElement('div'); bookDescWrapper.className = 'book-desc-wrapper';


```

Кафедра інтелектуальних інформаційних систем
Розробка вебзастосунку онлайн-бібліотеки

```

    bookDescWrapper.innerHTML = `
```

```

<div class="form-grid">
  <label>Book title: <input value="\${bookData['Book-Title']}"
id="editTitle"></label>
  <label>Author: <input value="\${bookData['Book-Author']}"
id="editAuthorBook"></label>
  <label>Year: <input value="\${bookData['Year-Of-Publication']}"
id="editYear"></label>
  <label>Genre: <input value="\${bookData['Genre']}"
id="editGenre"></label>
  <label>Language: <input value="\${bookData['Language']}"
id="editLang"></label>
  <label>Date: <input value="\${bookData['Date']}"
id="editDate"></label>
  <label>Country: <input value="\${bookData['Country']}"
id="editCountry"></label>
  <label>Author's name: <input value="\${bookData['Author']}"
id="editAuthor"></label>
</div>
<label>Book description:
  <textarea id="editBookDesc">\${bookData['Book-
Description']}</textarea>
</label>
<label>Author info:
  <textarea id="editAuthorDesc">\${bookData['Author-
Description']}</textarea>
</label>
<button id="saveChanges">Save</button>
`;
container.appendChild(form);
document.getElementById('saveChanges').addEventListener('click', ()
=> {
  bookData["Book-Title"] =
document.getElementById('editTitle').value;
  bookData["Book-Author"] =
document.getElementById('editAuthorBook').value;
  bookData["Year-Of-Publication"] =
document.getElementById('editYear').value;
  bookData["Genre"] = document.getElementById('editGenre').value;
  bookData["Language"] = document.getElementById('editLang').value;
  bookData["Book-Description"] =
document.getElementById('editBookDesc').value;
  bookData["Author"] = document.getElementById('editAuthor').value;
  bookData["Date"] = document.getElementById('editDate').value;
  bookData["Country"] = document.getElementById('editCountry').value;
  bookData["Author-Description"] =
document.getElementById('editAuthorDesc').value;

```



```

    localStorage.setItem(`book_${bookData["Book-Title"]}`,
JSON.stringify(bookData));
    localStorage.setItem(`book_${bookData["Book-Title"]}`,
JSON.stringify(bookData));
const alertBox = document.getElementById('customAlert');
alertBox.style.display = 'block';
setTimeout(() => alertBox.style.opacity = '1', 10);
setTimeout(() => {
    alertBox.style.opacity = '0';
    setTimeout(() => alertBox.style.display = 'none', 400);
}, 3000);
});
}
</script>
<script>
    document.addEventListener("DOMContentLoaded", () => {
        requestAnimationFrame(() => {
            document.body.style.opacity = '1';
        });
    });
    function smoothRedirect(url) {
        document.body.style.transition = 'opacity 0.5s ease';
        document.body.style.opacity = '0';
        setTimeout(() => {
            window.location.href = url;
        }, 300);
    }
</script>
<div id="customAlert" class="custom-alert">Changes saved! Update
page.</div>
</body>
</html>

```

ДОДАТОК В

Лістинг коду сторінки авторизації

```
<script>
  document.addEventListener("DOMContentLoaded", () => {
    requestAnimationFrame(() => {
      document.body.style.opacity = '1';
    });
  });
</script>
</head>
<body>
  <header>
    <div class="header__inner">
      <div class="header__links">
        <a href="" class="settings">
          
        </a>
        <a href="./about.html" class="header__text">About us</a>
        <a href="./genres.html" class="header__text">Genres</a>
      </div>
      <a href="./index.html" class="logo">
        
      </a>
      <div class="header__search">
        <a href="" class="settings">
          
        </a>
        <form class="search-form" onsubmit="searchBook(event)">
          <input type="text" name="q" id="searchInput"
placeholder="Search...">
        </form>
        <script>
          function searchBook(event) {
            event.preventDefault();
            const query =
document.getElementById('searchInput').value.trim();
            if (query) {
              window.location.href =
`bookpage.html?title=${encodeURIComponent(query)}`;
            }
          }
        </script>
      </div>
```

```

    </div>
</header>
<div class="container">
    <p class="you_are_reader">You are a reader. To edit, log in as
"editor"</p>
    
    <form id="loginForm" class="login-form">
    <label for="username">Login:
    <input type="text" id="username" required>
    </label>
    <label for="password">Password:
    <input type="password" id="password" required>
    </label>
    <button type="submit" id="loginButton">OK</button>
</form>
</div>
<div class="footer">
    <div class="footer__inner">
        <a href="" class="footer__text">© Liberia Preferita 2024-2025.
Усі права захищено</a>
        <div class="footer__links">
            <a
href="https://www.instagram.com/ksd_bookclub?utm_source=ig_web_button_s
hare_sheet&igsh=ZDNlZDc0MzIxNw==" class="footer__link">
                
            </a>
            <a href="https://ru.pinterest.com/" class="footer__link">
                
            </a>
            <a href="https://www.facebook.com/share/v/15hkv5J7jK/"
class="footer__link">
                
            </a>
        </div>
    </div>
</div>
<script>
document.getElementById('loginForm').addEventListener('submit',
function(e) {
    e.preventDefault();
    const username = document.getElementById('username').value.trim();
    const password = document.getElementById('password').value.trim();
    if (username === 'editor' && password === '0000') {

```

```
localStorage.setItem('role', 'editor');  
window.location.href = 'index.html';  
} else {  
    alert('Error');  
} });  
</script>  
</body>  
</html>
```

ДОДАТОК Г

Лістинг коду загальної сторінки жанрів

```
<script>
  document.addEventListener("DOMContentLoaded", () => {
    requestAnimationFrame(() => {
      document.body.style.opacity = '1';
    });
  });
</script>
</head>
<body>
<header>
  <div class="header__inner">
    <div class="header__links">
      <a href="./admin.html" class="settings">
        
      </a>
      <a href="./about.html" class="header__text">About us</a>
      <a href="./genres.html" class="header__text">Genres</a>
    </div>
    <a href="./index.html" class="logo">
      
    </a>
    <div class="header__search">
      <a href="" class="settings">
        
      </a>
      <form class="search-form" onsubmit="searchBook(event)">
        <input type="text" name="q" id="searchInput"
placeholder="Search...">
      </form>
      <button class="logout-btn" onclick="logout()" style="display:
none;">Log out</button>
      <script>
        function searchBook(event) {
          event.preventDefault();
          const query =
document.getElementById('searchInput').value.trim();
          if (query) {
            window.location.href =
`bookpage.html?title=${encodeURIComponent(query)}`;
          }
        }
      </script>
    </div>
  </div>
</header>
</body>
</html>
```

```

        </script>
    </div>
</div>
</header>
<div class="container">
    <div class="genres_inner_1">
        <div class="genre_card">
            
            <a href="./fantasy.html"
class="tittle_genre_card"><p>Fantasy</p></a>
        </div>
        <div class="genre_card">
            
            <a href="./fiction.html"
class="tittle_genre_card"><p>Fiction</p></a>
        </div>
        <div class="genre_card">
            
            <a href="./historical.html"
class="tittle_genre_card"><p>Historical fiction</p></a>
        </div>
        <div class="genre_card">
            
            <a href="./adventure.html" class="tittle_genre_card"><p>Adventure
fiction</p></a>
        </div>
        <div class="genre_card">
            
            <a href="./western.html" class="tittle_genre_card"><p>Western
fiction</p></a>
        </div>
        <div class="genre_card">
            
            <a href="./science.html" class="tittle_genre_card"><p>Science
fiction</p></a>
        </div>
    </div>
    <div class="decorative-divider_genres"></div>
    <div class="genres_inner_2">
        <div class="genre_card">
            
            <a href="./romance.html" class="tittle_genre_card"><p>Romance
novel</p></a>
        </div>
        <div class="genre_card">

```

```

    
    <a href="./thriller.html"
class="tittle_genre_card"><p>Thriller</p></a>
    </div>
    <div class="genre_card">
        
        <a href="./mythology.html"
class="tittle_genre_card"><p>Mythology</p></a>
    </div>
    <div class="genre_card">
        
        <a href="./magicalrealism.html"
class="tittle_genre_card"><p>Magical realism</p></a>
    </div>
    <div class="genre_card">
        
        <a href="./dystopian.html"
class="tittle_genre_card"><p>Dystopian</p></a>
    </div>
    <div class="genre_card">
        
        <a href="./horror.html" class="tittle_genre_card"><p>Horror
fiction</p></a>
    </div>
</div>
</div>
</div>
<div class="footer">
    <div class="footer__inner">
        <a href="" class="footer__text">© Liberia Preferita 2024-2025.
Усі права захищено</a>
        <div class="footer__links">
            <a
href="https://www.instagram.com/ksd_bookclub?utm_source=ig_web_button_s
hare_sheet&igsh=ZDNlZDc0MzIxNw==" class="footer__link">
                
            </a>
            <a href="https://ru.pinterest.com/" class="footer__link">
                
            </a>
            <a href="https://www.facebook.com/share/v/15hkv5J7jK/"
class="footer__link">
                
            </a>

```

```
        </div>
    </div>
</div>
<script>
document.addEventListener("DOMContentLoaded", () => {
    requestAnimationFrame(() => {
        document.body.style.opacity = '1';
    });
});
function smoothRedirect(url) {
    document.body.style.transition = 'opacity 0.5s ease';
    document.body.style.opacity = '0';
    setTimeout(() => {
        window.location.href = url;
    }, 300);
}
</script>
</body>
</html>
```


ДОДАТОК І

Лістинг коду сторінки окремого жанру

```

    <style>
    body {
        opacity: 0;
        transition: opacity 0.3s ease;
    }
</style>
<script>
    document.addEventListener("DOMContentLoaded", () => {
        requestAnimationFrame(() => {
            document.body.style.opacity = '1';
        });
    });
</script>
</head>
<body>
<header>
    <div class="header__inner">
        <div class="header__links">
            <a href="./admin.html" class="settings">
                
            </a>
            <a href="./about.html" class="header__text">About us</a>
            <a href="./genres.html" class="header__text">Genres</a>
        </div>
        <a href="./index.html" class="logo">
            
        </a>
        <div class="header__search">
            <a href="" class="settings">
                
            </a>
            <form class="search-form" onsubmit="searchBook(event)">
                <input type="text" name="q" id="searchInput"
placeholder="Search...">
            </form>
            <button class="logout-btn" onclick="logout()" style="display:
none;">Log out</button>
        </div>
        <script>
            function searchBook(event) {
                event.preventDefault();
            }
        </script>
    </div>
</header>

```

```

        const query =
document.getElementById('searchInput').value.trim();
        if (query) {
            window.location.href =
`bookpage.html?title=${encodeURIComponent(query)}`;
        }
    }
    </script>
</div>
</div>
</header>
<div class="container">
    <div class="genre_inner">
        
        <div class="genre_desc">
            <p class="genre_name">Historical fiction</p>
            <p class="genre_text">...</p>
        </div>
    </div>
    <p class="books_genre">BOOKS OF THE GENRE</p>
    <div class="books_genres_inner_1">
        <a href="bookpage.html?title=Wolf%20Hall"></a>
        <a href="bookpage.html?title=The%20Iron%20King"></a>
        <a href="bookpage.html?title=La%20Reine%20Margot"></a>
    </div>
    <div class="books_genres_inner_2">
        <a href="bookpage.html?title=The%20Hunchback%20of%20Notre-
Dame">
</a>
        <a
href="bookpage.html?title=The%20Pillars%20of%20the%20Earth"></a>
    </div>
</div>
<div class="footer">
    <div class="footer__inner">
        <a href="" class="footer__text">© Liberia Preferita 2024-2025.
Усі права захищено</a>
        <div class="footer__links">
            <a
href="https://www.instagram.com/ksd_bookclub?utm_source=ig_web_button_s
hare_sheet&igsh=ZDNlZDc0MzIxNw==" class="footer__link">

```

```

        
    </a>
    <a href="https://ru.pinterest.com/" class="footer__link">
        
    </a>
    <a href="https://www.facebook.com/share/v/15hkv5J7jK/"
class="footer__link">
        
    </a>
</div>
</div>
</div>
<script>
document.addEventListener("DOMContentLoaded", () => {
    requestAnimationFrame(() => {
        document.body.style.opacity = '1';
    });
});
function smoothRedirect(url) {
    document.body.style.transition = 'opacity 0.5s ease';
    document.body.style.opacity = '0';
    setTimeout(() => {
        window.location.href = url;
    }, 300);
}
</script>
</body>
</html>

```

