МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «**РОЗРОБКА ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ "ОСОБИСТА БІБЛІОТЕКА ЕЛЕКТРОННИХ КНИГ»**

Виконав: студент 4 курсу, групи 6.1222
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)
освітньої програми Комп'ютерні науки
(назва освітньої програми)
В.Є. Богма
(ініціали та прізвище)
Керівник доцент кафедри комп'ютерних наук, к.т.н., доцент Борю С.Ю.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент завідувач кафедри програмної інженерії, доцент, к.ф.-м.н., А. О. Лісняк
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки
(шифр і назва)

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри комп'ютерних наук, к.пед.н., доцент

_____ Пшенична О.С.
(підпис)

" 08 " грудня 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Богмі Владиславу Євгеновичу
(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка інформаційно-довідкової системи "Особиста бібліотека електронних книг"

керівник роботи Борю Сергій Юрійович, к.т.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 02 » грудня 2025 року № 1741-с

2. Строк подання студентом роботи 04.06.2026

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Розробка інформаційно-довідкової системи "Особиста бібліотека електронних книг"

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 05.12.2025

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	23.12.2025	
2.	Збір вихідних даних.	11.01.2026	
3.	Обробка методичних та теоретичних	07.02.2026	
	джерел.		
4.	Розробка першого та другого розділу.	18.03.2026	
5.	Розробка третього розділу.	28.04.2026	
6.	Оформлення та нормоконтроль	16.05.2026	
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	18.06.2026	

Студент _____ В.Є. Богма
(підпис) (ініціали та прізвище)

Керівник роботи _____ С.Ю. Борю
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г.Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка інформаційно-довідкової системи "Особиста бібліотека електронних книг"»: 49 с., 13 рис., 2 табл., 9 джерел, 2 додатки.

БАЗА ДАНИХ, ВЕБ-ДОДАТОК, ЕЛЕКТРОННА БІБЛІОТЕКА, ЕЛЕКТРОННА КНИГА, ІНФОРМАЦІЙНО-ДОВІДКОВА СИСТЕМА, КОЛЕКЦІЯ КНИГ, LARAVEL, MYSQL.

Об’єкт дослідження – процеси зберігання, організації та обробки інформації в електронних бібліотечних системах.

Предмет дослідження − методи та засоби розробки веб-орієнтованої інформаційно-довідкової системи для управління особистою бібліотекою електронних книг.

Мета роботи: розробка інформаційно-довідкової системи «Особиста бібліотека електронних книг», що забезпечує зручне зберігання, пошук, систематизацію та управління електронними книгами користувача.

Методи дослідження – аналіз наукової та технічної літератури, порівняльний аналіз існуючих програмних рішень, методи об’єктно-орієнтованого проєктування, моделювання структури даних, програмування веб-додатків, функціональне тестування програмного забезпечення.

У результаті виконання роботи розроблено та реалізовано веб-орієнтовану інформаційно-довідкову систему для управління особистою бібліотекою електронних книг. Система забезпечує реєстрацію користувачів, завантаження та зберігання книг, створення колекцій, а також пошук і фільтрацію літератури. Проведене тестування підтвердило коректність роботи реалізованого програмного забезпечення.

SUMMARY

Bachelor’s Qualifying Thesis «Development of the Information and Reference System "Personal Library of Electronic Books"»: 49 pages, 13 figures, 2 tables, 9 sources, 2 supplements.

DATABASE, WEB APPLICATION, DIGITAL LIBRARY, E-BOOK, INFORMATION AND REFERENCE SYSTEM, BOOK COLLECTION, LARAVEL, MYSQL.

Object of the study – the processes of storing, organizing, and processing information in electronic library systems.

Subject of the study – methods and tools for developing a web-based information and reference system for managing a personal e-book library.

Aim of the study: to develop an information and reference system called ‘Personal E-book Library’, which enables users to conveniently store, search for, organize and manage their e-books.

Methods of research – analysis of scientific and technical literature, comparative analysis of existing software solutions, object-oriented design methods, data structure modelling, web application programming, and software functional testing.

As a result of this work, a web-based information and reference system for managing a personal library of e-books has been developed and implemented. The system enables user registration, the downloading and storage of books, the creation of collections, as well as the searching for and filtering of literature. Testing carried out has confirmed that the implemented software functions correctly.

ЗМІСТ

Завдання на кваліфікаційну роботу……..2
Реферат 4
Summary 5
Зміст 6
1 Аналіз предметної області та обґрунтування напряму дослідження 10
1.1 Інформаційно-довідкові системи: поняття, класифікація та призначення 10
1.2 Особливості зберігання електронних книг у приватних бібліотеках. 12
1.3 Аналіз існуючих програмних рішень електронних бібліотек 14
1.4 Мета та завдання дослідження 16
2 Розробка та дослідження інформаційно-довідкової системи 18
2.1 Призначення та область застосування системи 18
2.2 Аналіз можливих методів та підходів до розв’язання задачі 19
2.3 Вибір та обґрунтування технічних рішень 20
2.4 Порівняльний аналіз можливих варіантів реалізації 21
3 Реалізація інформаційно-довідкової системи 24
3.1. Загальна структура та модель системи 24
3.2. Розробка алгоритмів та програмної реалізації 25
3.2.1 Алгоритм додавання книги до системи 25
3.2.2 Алгоритм завантаження електронних книг у вигляді архіву 26
3.2.3 Алгоритм пошуку книг 27
3.2.4 Алгоритм збереження прогресу читання 27
3.2.5 Програмна реалізація 27
3.3. Реалізація бази даних та програмних модулів 28
3.3.1 Основні сутності бази даних 28
3.3.2 Службові таблиці Laravel 30
3.3.3. Програмні модулі системи 30

3.4. Аналіз результатів та перевірка працездатності системи 32
3.4.1 Перевірка основних функцій системи 33
3.5. Практична реалізація та приклади впровадження 38
Висновки 40
Перелік посилань 42
Додаток А Обробка книги різних форматів на прикладі сервісу EPUB 43
Додаток Б Метод заповнення метаданих 48

ВСТУП

У сучасному інформаційному суспільстві значно зросла роль електронних джерел інформації, зокрема електронних книг. Використання цифрових бібліотек дозволяє ефективно зберігати, систематизувати та швидко отримувати доступ до великого обсягу літератури. У зв’язку з цим актуальним є розроблення зручних інформаційно-довідкових систем для персонального використання.

Метою даної дипломної роботи є розробка інформаційно-довідкової системи «Особиста бібліотека електронних книг», яка забезпечує користувачам можливість зберігання, організації та управління електронними книгами, а також супутньою інформацією, такою як колекції, жанри, нотатки та прогрес читання.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати існуючі інформаційні системи та підходи до організації електронних бібліотек;
- обґрунтувати вибір технологій для реалізації системи;
- розробити структуру бази даних;
- спроєктувати архітектуру програмного забезпечення;
- реалізувати основні функціональні модулі системи;
- забезпечити можливість роботи з електронними книгами, колекціями та користувацькими даними;
- провести тестування розробленої системи.

Об’єктом дослідження є процеси зберігання та обробки інформації в електронних бібліотечних системах.

Предметом дослідження є методи та засоби розробки веб-орієнтованих інформаційно-довідкових систем.

У процесі виконання роботи використано сучасні технології веб-розробки, зокрема мову програмування PHP, фреймворк Laravel, систему

управління базами даних MySQL, а також HTML, CSS та Blade-шаблони для формування користувацького інтерфейсу.

Практичне значення роботи полягає у створенні функціональної системи, яка може бути використана як персональна електронна бібліотека для організації навчальних та літературних матеріалів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ НАПРЯМУ ДОСЛІДЖЕННЯ

1.1 Інформаційно-довідкові системи: поняття, класифікація та призначення

Постійно зростаючий обсяг даних є головною відзнакою сучасного суспільства. Не дивлячись на те, що наразі немає проблем зі зберіганням великої кількості інформації, є велика різниця між тим, як інформація буде зберігатися. Задля цього існує десятки спеціалізованих програмних засобів, які не тільки зберігають, але й обробляють для подальшого зручного використання. Одним із таких засобів є інформаційно-довідкові системи (ІДС)

Інформаційно-довідкова система – це сукупність програмних, технічних та організаційних засобів, призначених для накопичення, зберігання, пошуку, обробки та надання користувачеві довідкової інформації у зручному вигляді. Основною метою ІДС є забезпечення швидкого отримання актуальних відомостей за заданими критеріями пошуку.

Залежно від сфери застосування та функціонального призначення інформаційно-довідкові системи можна класифікувати за кількома ознаками. За сферою використання виділяють бібліотечні, освітні, медичні, юридичні, наукові та корпоративні системи. За способом розміщення розрізняють локальні та веборієнтовані системи. За масштабом використання ІДС можуть бути персональними, груповими або корпоративними. Окрему категорію становлять спеціалізовані системи, які орієнтовані на роботу з конкретним видом інформації, наприклад електронними книгами, нормативними документами або навчальними матеріалами.

Основними функціями інформаційно-довідкових систем є зберігання даних, пошук інформації за різними параметрами, сортування та фільтрація записів, оновлення відомостей, формування довідок і забезпечення швидкого

доступу до необхідних ресурсів. У сучасних системах додатково реалізуються функції персоналізації, категоризації даних, ведення статистики використання та контролю доступу користувачів.

Під час розробки інформаційно-довідкових систем важливе значення мають принципи їх побудови. До основних принципів належать структурованість даних, цілісність інформації, зручність користувацького інтерфейсу, масштабованість, надійність зберігання даних та ефективність пошукових механізмів. Крім того, система повинна забезпечувати можливість подальшого розширення функціоналу без суттєвої зміни її архітектури.

Особливе місце займають невеликі локальні інформаційні системи, які призначені для використання окремими користувачами або невеликими групами. Такі системи характеризуються простотою впровадження, невисокими вимогами до апаратних ресурсів та відсутністю необхідності у складній серверній інфраструктурі. Незважаючи на обмежений масштаб, вони дозволяють ефективно організовувати інформацію та забезпечувати швидкий доступ до неї.

Однією з головних переваг інформаційно-довідкових систем є підвищення зручності зберігання та підтримки інформації. Використання централізованої бази даних дозволяє уникнути дублювання відомостей, забезпечує їх актуальність та спрощує процес оновлення. Крім того, автоматизований пошук значно скорочує час, необхідний для отримання потрібної інформації, що підвищує ефективність роботи користувача.

Таким чином, інформаційно-довідкові системи є важливим інструментом організації та використання інформаційних ресурсів. Їх застосування дозволяє оптимізувати процеси зберігання, пошуку та обробки даних, забезпечуючи користувачів необхідною інформацією в оперативному та зручному вигляді. Для системи «Особиста бібліотека електронних книг» використання принципів побудови інформаційно-довідкових систем є основою для створення ефективного програмного засобу керування електронною книжковою колекцією.

1.2 Особливості зберігання електронних книг у приватних бібліотеках

З поширенням цифрових технологій електронні книги стали одним із найпопулярніших способів зберігання та споживання літератури. На відміну від традиційних бібліотек, приватні електронні бібліотеки формуються безпосередньо користувачами та можуть містити сотні або навіть тисячі книг різних жанрів і форматів. При цьому способи організації таких колекцій значною мірою залежать від особистих уподобань власника бібліотеки та використовуваних програмних засобів.

Найпоширенішим способом зберігання електронних книг є використання файлів, готових до безпосереднього читання. До таких належать книги у форматах FB2, EPUB, PDF, MOBI та інших, які підтримуються сучасними програмами для читання та електронними рідерами. У найпростішому випадку всі файли зберігаються в одній директорії без додаткової структури, що є зручним на початковому етапі формування бібліотеки, але створює труднощі під час пошуку потрібної книги в великих колекціях.

Досить часто користувачі зберігають книги у вигляді архівів ZIP або RAR. Такий підхід дозволяє зменшити обсяг займаного дискового простору та зберегти оригінальну структуру файлів. Багато програм для читання електронних книг підтримують відкриття архівованих файлів без попереднього розпакування, що робить цей спосіб досить популярним серед власників великих колекцій.

Для полегшення навігації користувачі часто групують книги за авторами. У цьому випадку для кожного автора створюється окрема папка, що містить усі його твори. Усередині таких каталогів можуть знаходитися як окремі файли книг, так і архіви з ними. Подібна структура дозволяє швидко

знаходити твори конкретного автора та підтримувати впорядкованість колекції.

Ще одним поширеним способом організації є зберігання книг за серіями або циклами. Для кожного циклу створюється окрема папка, назва якої відповідає назві серії. Усередині розміщуються файли книг у порядку їх читання або публікації. Такий підхід особливо зручний для художньої літератури, де послідовність ознайомлення з творами має важливе значення для розуміння сюжету.

Окрім готових до читання файлів, користувачі часто зберігають так звані не готові до використання колекції. До них належать великі архіви, що містять значну кількість книг, каталогів або підкаталогів. Такі архіви зазвичай використовуються як резервні копії бібліотеки або завантажуються з тематичних ресурсів в Інтернеті для подальшого сортування. Часто подібні колекції розміщуються на зовнішніх накопичувачах, флеш-пам’яті або мережевих сховищах даних.

Незважаючи на різноманітність способів зберігання електронних книг, більшість із них має спільний недолік – відсутність єдиного механізму обліку та пошуку інформації. Зі збільшенням розміру бібліотеки користувач стикається з проблемами пошуку потрібної книги, визначення прочитаних творів, контролю за дублюванням файлів та ведення статистики читання. Використання лише файлової структури операційної системи не забезпечує достатнього рівня зручності для роботи з великими книжковими колекціями.

Саме тому актуальним є створення спеціалізованих інформаційно-довідкових систем для керування особистими бібліотеками електронних книг. Такі системи дозволяють автоматизувати каталогізацію видань, зберігати додаткові відомості про книги, виконувати швидкий пошук за різними параметрами та забезпечувати зручний доступ до всієї колекції через єдиний інтерфейс. Для користувача це значно спрощує процес зберігання, впорядкування та використання електронних книг незалежно від їх кількості та формату.

1.3 Аналіз існуючих програмних рішень електронних бібліотек

Зі збільшенням обсягів електронної літератури виникла потреба у програмних засобах для її впорядкування, пошуку та зберігання. На сьогодні існує значна кількість програм, призначених для ведення особистих електронних бібліотек. Вони відрізняються функціональними можливостями, підтримуваними форматами та складністю використання.

Одним із найвідоміших рішень є Calibre. Дана система дозволяє зберігати електронні книги, редагувати метадані, конвертувати файли між різними форматами та синхронізувати бібліотеку з електронними рідерами. Перевагою Calibre є широкий набір функцій та підтримка великої кількості форматів електронних книг. Проте для нових користувачів програма може здаватися перевантаженою через складний інтерфейс і велику кількість налаштувань.

Іншим популярним рішенням є Alfa eBooks Manager. Програма орієнтована на створення особистих книжкових каталогів із використанням обкладинок, рейтингів, описів та додаткових метаданих. Її перевагою є сучасний інтерфейс та зручне представлення інформації. Водночас частина функціональних можливостей доступна лише у платній версії програмного продукту.

Для роботи з великими колекціями художньої літератури часто використовується MyHomeLib. Система забезпечує швидкий пошук за автором, назвою твору, серією або жанром, а також підтримує роботу з великими архівами книг у форматі FB2. Основним недоліком є застарілий інтерфейс користувача та обмежена підтримка сучасних форматів електронних видань.

На практиці багато користувачів взагалі не використовують спеціалізовані програми, а організовують бібліотеку за допомогою файлової системи операційної системи. Книги розміщуються в папках за авторами,

жанрами або серіями. Такий спосіб не потребує додаткового програмного забезпечення, проте значно ускладнює пошук інформації, контроль прочитаних книг та зберігання додаткових відомостей про видання.

Проведений аналіз (табл. 1.1) показує, що існуючі програмні рішення мають широкий функціонал, однак часто характеризуються надлишковою складністю, прив'язкою до певних форматів або відсутністю можливостей персоналізації під потреби конкретного користувача. Крім того, багато систем орієнтовані на великі бібліотеки та містять функції, які не є необхідними для ведення особистої книжкової колекції.

Таблиця 1.1 – Порівняльна характеристика існуючих рішень

Функціональна можливість	**Calibre**	**Alfa eBooks Manager**	**MyHomeLib**	**Файлова система**
Зберігання книг	+	+	+	+
Автоматичне отримання метаданих	+	+	-	-
Робота з обкладинками	+	+	+	-
Пошук за автором та назвою	+	+	+	±
Ведення особистих колекцій	+	+	-	-
Облік прогресу читання	-	-	-	-
Безкоштовне використання	+	±	+	+

У межах даної роботи пропонується розробка інформаційно-довідкової системи «Особиста бібліотека електронних книг», яка буде реалізована як вебзастосунок. Такий підхід спрощує впровадження системи, оскільки

користувачу достатньо мати веббраузер без необхідності встановлення додаткового програмного забезпечення. Запропонована система забезпечить централізоване зберігання електронних книг, підтримку основних форматів файлів, створення тематичних колекцій, облік прогресу читання, пошук за різними критеріями та зручний доступ до бібліотеки через мережу Інтернет. Це дозволить усунути недоліки розглянутих рішень та забезпечити більш просте й ефективне використання системи для ведення особистої електронної бібліотеки.

На основі наведеної таблиці можна зробити висновок, що жодна з розглянутих систем повністю не задовольняє потреби користувача щодо простого керування особистою бібліотекою та контролю процесу читання. Саме усунення виявлених недоліків буде враховано під час проєктування розроблюваної інформаційно-довідкової системи.

1.4 Мета та завдання дослідження

Метою дипломної роботи є розробка інформаційно-довідкової системи «Особиста бібліотека електронних книг», призначеної для зручного зберігання, каталогізації, пошуку та перегляду електронних книг, а також контролю процесу читання користувача.

Об'єктом дослідження є процес організації та управління особистими колекціями електронних книг.

Предметом дослідження є методи та програмні засоби створення інформаційно-довідкових систем для зберігання й обробки даних про електронні книги.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– проаналізувати принципи функціонування інформаційно-довідкових систем та особливості їх використання;

- дослідити способи зберігання електронних книг у приватних бібліотеках;
- виконати аналіз існуючих програмних рішень для ведення електронних бібліотек;
- визначити функціональні та нефункціональні вимоги до розроблюваної системи;
- спроєктувати структуру бази даних та архітектуру програмного забезпечення;
- реалізувати механізми додавання, редагування, видалення та пошуку електронних книг;
- реалізувати функціонал створення користувацьких колекцій книг;
- забезпечити підтримку основних форматів електронних книг та можливість їх перегляду;
- реалізувати механізм збереження прогресу читання користувача;
- провести тестування розробленої системи та оцінити результати її роботи.

Результатом виконання дипломної роботи має стати веборієнтована інформаційно-довідкова система, що забезпечує ефективне керування особистою бібліотекою електронних книг та спрощує доступ користувача до власної книжкової колекції.

2 РОЗРОБКА ТА ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ

2.1 Призначення та область застосування системи

Інформаційно-довідкова система «Особиста бібліотека електронних книг» призначена для організації, зберігання та зручного доступу до електронних книг користувача. Основною метою системи є забезпечення швидкого пошуку літератури, структурованого зберігання книжкових ресурсів, а також можливості їх перегляду та управління особистою бібліотекою в цифровому форматі. Система дозволяє користувачам додавати нові книги, редагувати їхні дані, згрупувати книги за категоріями або колекціями, а також здійснювати пошук за різними параметрами, такими як назва та автор.

Цільовою аудиторією даної системи є широке коло користувачів, зокрема студенти, викладачі, дослідники, а також усі особи, які активно користуються електронною літературою. Окрему категорію користувачів становлять люди, які прагнуть систематизувати власну цифрову бібліотеку та мати швидкий доступ до необхідних джерел інформації без необхідності використання сторонніх сервісів.

Умови використання системи передбачають наявність пристрою з доступом до веб-браузера та підключенням до мережі Інтернет. Система реалізується у вигляді веб-застосунку, що забезпечує кросплатформеність і можливість використання як на персональних комп’ютерах, так і на мобільних пристроях. Для повноцінної роботи користувач повинен мати обліковий запис, який забезпечує персоналізований доступ до бібліотеки та збереження даних.

Обмеження системи полягають у залежності від якості інтернет-з’єднання та серверної інфраструктури. Також система не передбачає автоматичного наповнення бібліотеки контентом із зовнішніх джерел, а лише

працює з файлами, які додає сам користувач. Система підтримує роботу з основними форматами електронних документів, зокрема PDF, EPUB та TXT, що забезпечує сумісність із більшістю джерел цифрової літератури.

2.2 Аналіз можливих методів та підходів до розв'язання задачі

Для реалізації інформаційно-довідкової системи «Особиста бібліотека електронних книг» можуть бути застосовані різні підходи до організації архітектури, зберігання даних та реалізації функціоналу роботи з електронними книгами. Основною задачею є забезпечення ефективного управління великим обсягом структурованої інформації, що включає метадані книг, файли електронних документів, а також дані користувачів.

Одним із базових підходів є використання класичної багатошарової архітектури веб-застосунку, яка передбачає розділення системи на рівні представлення, бізнес-логіки та доступу до даних. Така архітектура є зручною для супроводу та масштабування, оскільки дозволяє ізолювати окремі компоненти системи та спрощує їх подальший розвиток.

Для зберігання даних доцільно використовувати реляційні бази даних, зокрема MySQL, яка забезпечує надійне зберігання структурованої інформації та підтримує складні запити для пошуку і фільтрації книг. Альтернативним підходом може бути використання NoSQL-рішень, однак у контексті даної задачі вони не є критично необхідними, оскільки структура даних є чітко визначеною та відносно стабільною.

Для реалізації серверної частини системи можуть бути використані різні технологічні стеки, зокрема PHP-фреймворки (наприклад, Laravel) або інші сучасні веб-фреймворки. Використання фреймворку дозволяє прискорити розробку за рахунок готових інструментів для маршрутизації, роботи з базою даних, автентифікації та управління файлами.

Окрему увагу слід приділити методам обробки та зберігання електронних книг. Файли можуть зберігатися як безпосередньо на сервері файлової системи, так і у хмарних сховищах. У даній системі доцільно використовувати файлове зберігання на сервері з фіксацією шляхів до файлів у базі даних, що спрощує реалізацію та зменшує залежність від зовнішніх сервісів.

Щодо реалізації функціоналу пошуку можуть застосовуватися як стандартні SQL-запити з використанням операторів LIKE, так і більш складні механізми індексації даних. Для невеликих та середніх обсягів даних достатньо використання вбудованих можливостей реляційної бази даних.

Таким чином, найбільш доцільним підходом для реалізації системи є використання веб-орієнтованої багатошарової архітектури з реляційною базою даних, серверною логікою на основі PHP-фреймворку та файловим зберіганням електронних книг на сервері.

2.3 Вибір та обґрунтування технічних рішень

Для реалізації інформаційно-довідкової системи «Особиста бібліотека електронних книг» було обрано набір технічних рішень, які забезпечують простоту розробки, надійність роботи та можливість подальшого масштабування системи. Вибір технологій базується на вимогах до веб-застосунку, необхідності роботи з файлами електронних книг та організації структурованого зберігання даних.

Як серверну платформу доцільно використовувати PHP, оскільки ця мова програмування є добре пристосованою для розробки веб-додатків, має широку підтримку хостингів та велику кількість готових бібліотек. У межах даної системи використання PHP дозволяє ефективно реалізувати бізнес-логіку, роботу з файлами, автентифікацію користувачів та взаємодію з базою даних.

Як основний фреймворк обрано Laravel, який забезпечує структуровану архітектуру застосунку, підтримку шаблону MVC, зручну роботу з маршрутизацією, міграціями бази даних та ORM (Eloquent). Використання Laravel значно прискорює процес розробки та підвищує якість коду завдяки дотриманню сучасних стандартів проєктування.

Для реалізації клієнтської частини використовується серверний рендеринг шаблонів за допомогою Blade. Такий підхід дозволяє створювати простий і зрозумілий інтерфейс без необхідності використання складних JavaScript-фреймворків, що відповідає вимогам мінімальної складності та швидкої реалізації.

Як систему управління базами даних обрано MySQL, яка забезпечує надійне зберігання структурованих даних, підтримку зв’язків між таблицями та високу продуктивність при виконанні запитів. MySQL є оптимальним вибором для даного типу застосунків завдяки своїй стабільності та широкій підтримці.

Для зберігання електронних книг та обкладинок використовується файлове сховище на сервері. Метадані файлів (шлях, назва, тип) зберігаються у базі даних, що дозволяє ефективно організувати доступ до ресурсів та забезпечити їх логічну структуру.

Таким чином, обрана сукупність технічних рішень – PHP, Laravel, MySQL та Blade-шаблони – забезпечує оптимальний баланс між простотою реалізації, продуктивністю та можливістю подальшого розвитку інформаційно-довідкової системи.

2.4 Порівняльний аналіз можливих варіантів реалізації

Під час проєктування інформаційно-довідкової системи «Особиста бібліотека електронних книг» було розглянуто декілька можливих варіантів реалізації як серверної, так і клієнтської частини системи, а також підходів до

зберігання даних (табл. 2.1). Порівняльний аналіз проводиться за такими основними критеріями: складність реалізації, продуктивність, масштабованість, зручність супроводу та наявність готових інструментів.

Таблиця 2.1 – Порівняння розглянутих рішень за критеріями

Критерій	**PHP + Laravel**	**Node.js**	**Django**	**Spring Boot**
Складність	Низька	Середня	Низька	Висока
Швидкість розробки	Висока	Середня	Висока	Середня
Продуктивність	Висока	Висока	Висока	Дуже висока
Супровід	Простий	Середній	Простий	Складний

Як альтернативу PHP та Laravel можна розглядати використання інших серверних технологій, зокрема Node.js (Express), Python (Django або Flask), а також Java (Spring Boot). Node.js забезпечує високу продуктивність при роботі з великою кількістю запитів у реальному часі, однак має більш складну екосистему для класичних CRUD-застосунків. Python (Django) є зручним для швидкої розробки та має потужний ORM, але потребує іншого стеку та менш поширений у класичних веб-хостингах порівняно з PHP. Java (Spring Boot) забезпечує високу надійність і масштабованість, однак є значно складнішим у розробці та надмірним для задачі середнього рівня складності.

У контексті клієнтської частини розглядалися два основні підходи: використання JavaScript-фреймворків (React, Vue.js) або серверного рендерингу шаблонів (Blade). Використання SPA-фреймворків дозволяє створювати більш динамічний інтерфейс, однак значно ускладнює архітектуру системи та потребує додаткової логіки API. У свою чергу Blade-шаблони забезпечують простоту реалізації, швидку розробку та достатній рівень функціональності для даного типу застосунку.

Щодо систем управління базами даних, окрім MySQL, могли бути використані PostgreSQL або NoSQL-рішення, наприклад MongoDB. PostgreSQL має розширені можливості для складних запитів та кращу відповідність стандартам SQL, однак у межах даної задачі його переваги не є критичними. MongoDB дозволяє гнучко працювати з неструктурованими даними, проте для системи з чітко визначеною структурою сутностей (книги, користувачі, колекції) реляційна модель є більш доцільною.

На основі проведеного аналізу було обрано стек PHP + Laravel + MySQL з використанням Blade-шаблонів як найбільш доцільний для реалізації даної інформаційно-довідкової системи. Дане рішення забезпечує оптимальний баланс між простотою розробки, функціональністю та можливістю подальшого розширення системи. Крім того, обрані технології добре підходять для навчального проєкту, відповідають вимогам до дипломної роботи та дозволяють реалізувати весь необхідний функціонал без надмірної складності архітектури.

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ

3.1 Загальна структура та модель системи

Розроблена інформаційно-довідкова система «Особиста бібліотека електронних книг» побудована за багаторівневою архітектурою типу MVC (Model–View–Controller), яка забезпечує чіткий розподіл відповідальності між компонентами системи.

Система призначена для зберігання, обробки та відображення інформації про електронні книги, а також для управління користувацькими колекціями. Основними функціональними можливостями є додавання книг, перегляд бібліотеки, створення колекцій, пошук та відстеження прогресу читання.

Модель системи базується на взаємодії трьох основних рівнів:

- Model (Модель) – відповідає за роботу з базою даних та бізнес-логіку;
- View (Представлення) – забезпечує користувацький інтерфейс;
- Controller (Контролер) – обробляє запити користувача та координує взаємодію між моделлю і представленням.

Структура системи включає такі основні компоненти:

- клієнтська частина (веб-інтерфейс користувача);
- серверна частина (Laravel-додаток);
- база даних (MySQL);
- файлове сховище (зберігання електронних книг та обкладинок).

Взаємодія між компонентами системи реалізується через стандартний механізм HTTP-запитів у фреймворку Laravel, що забезпечує реалізацію архітектури MVC (Model–View–Controller).

Логічна структура обробки запиту включає послідовність взаємодії основних компонентів системи: користувач ініціює запит через веб-інтерфейс, після чого він надходить до маршрутизатора Laravel. Далі відповідний

контролер обробляє запит та викликає необхідні методи бізнес-логіки, реалізовані в моделі. Модель взаємодіє з базою даних MySQL, виконуючи операції збереження, отримання або оновлення даних. Отриманий результат передається назад до контролера та відображається користувачу у вигляді Blade-шаблону.

Таким чином забезпечується чіткий поділ відповідальності між компонентами системи та спрощується її подальше супроводження і масштабування.

3.2 Розробка алгоритмів та програмної реалізації

Розробка програмної частини інформаційно-довідкової системи «Особиста бібліотека електронних книг» здійснювалася з використанням фреймворку **Laravel**, який забезпечує реалізацію архітектури MVC та дозволяє ефективно організувати бізнес-логіку, взаємодію з базою даних і користувацький інтерфейс.

Основною метою програмної реалізації є забезпечення таких функціональних процесів:

- управління електронними книгами (створення, редагування, видалення, перегляд);
- робота з колекціями книг;
- пошук та фільтрація даних;
- збереження прогресу читання користувача;
- робота з файлами (обкладинки та електронні книги).

3.2.1 Алгоритм додавання книги до системи

Одним із ключових алгоритмів системи є алгоритм додавання нової книги до бібліотеки.

Послідовність виконання алгоритму:

- користувач відкриває форму додавання книги;
- завантажує файл книги та обкладинку (за наявності);
- якщо файл книги (наприклад EPUB має метадані, сервер зчитує їх та автоматично заповнює форму);
- вводить основні дані (назва, автор, опис);
- дані передаються до контролера через HTTP-запит;
- контролер виконує валідацію введених даних;
- файли зберігаються у файловому сховищі;
- інформація про книгу записується у базу даних MySQL;
- користувач отримує підтвердження успішного додавання.

3.2.2 Алгоритм завантаження електронних книг у вигляді архіву

У системі передбачено розширений функціонал завантаження електронних книг у вигляді ZIP-архіву, що дозволяє користувачу додавати одразу декілька файлів та автоматично формувати структуру бібліотеки.

Завантаження архіву реалізовано як окремий процес обробки файлів, який включає розпакування вмісту, аналіз структури каталогів та автоматичне створення записів у базі даних.

Користувач завантажує ZIP-архів через інтерфейс системи, після чого контролер приймає файл і виконує первинну перевірку його формату, розміру та цілісності. Далі архів розпаковується у тимчасову директорію на сервері, а система рекурсивно обходить його вміст. Для кожного виявленого файлу книги визначається формат (EPUB, PDF, TXT тощо), формуються метадані (назва, шлях та зв’язок із колекцією), після чого файл зберігається у файловому сховищі системи. На наступному етапі в базі даних створюються відповідні записи в таблицях книг (books), колекцій (collections) та зв’язків між ними (collection_book). Після завершення обробки всі тимчасові файли видаляються, а користувач отримує повідомлення про успішне імпортування книг.

3.2.3 Алгоритм пошуку книг

Для забезпечення швидкого доступу до інформації в системі реалізовано алгоритм пошуку книг. Користувач вводить пошуковий запит через інтерфейс системи, після чого контролер отримує його та передає до моделі для обробки. На основі отриманого запиту виконується SQL-запит до бази даних із використанням операторів LIKE або механізмів повнотекстового пошуку. Система аналізує записи та формує список книг, що відповідають критеріям пошуку. Після завершення обробки результати повертаються користувачу та відображаються в інтерфейсі у вигляді переліку знайдених книг.

3.2.4 Алгоритм збереження прогресу читання

Для відстеження прогресу читання в системі реалізовано механізм збереження поточної позиції користувача в книзі. Під час читання користувач відкриває книгу та переходить між сторінками, а інформація про поточну сторінку передається на сервер. Система перевіряє наявність відповідного запису в таблиці reading_progress. Якщо запис уже існує, його дані оновлюються відповідно до нової позиції читання; якщо запис відсутній, створюється новий. Після цього інформація про прогрес читання зберігається в базі даних, що дозволяє користувачеві продовжити читання з останньої збереженої сторінки.

3.2.5 Програмна реалізація

Програмна реалізація системи базується на використанні наступних технологій:

- **PHP 8+** – основна мова програмування;
- **Laravel Framework** – реалізація MVC та бізнес-логіки;

- **MySQL** – система управління базами даних;
- **Blade Templates** – формування інтерфейсу користувача;
- **HTML/CSS** – оформлення веб-сторінок;
- **Storage API Laravel** – робота з файлами.

Структура програмного коду поділена на:

- контролери (обробка запитів);
- моделі (робота з даними);
- представлення (UI);
- міграції (структура БД).

3.3 Реалізація бази даних та програмних модулів

База даних інформаційно-довідкової системи «Особиста бібліотека електронних книг» реалізована з використанням **MySQL**, що забезпечує надійне зберігання структурованих даних та підтримку зв’язків між сутностями (рис. 3.1).

Проектування бази даних виконано з урахуванням принципів нормалізації, що дозволяє мінімізувати надлишковість даних та забезпечити цілісність інформації.

3.3.1 Основні сутності бази даних

Таблиця users

Таблиця призначена для зберігання інформації про користувачів системи:

- id – унікальний ідентифікатор користувача;
- name – ім’я користувача;
- email – електронна пошта;
- password – пароль доступу;

– role – роль користувача (user/admin).

Дана таблиця є центральною для авторизації та зв’язана з більшістю інших сутностей.

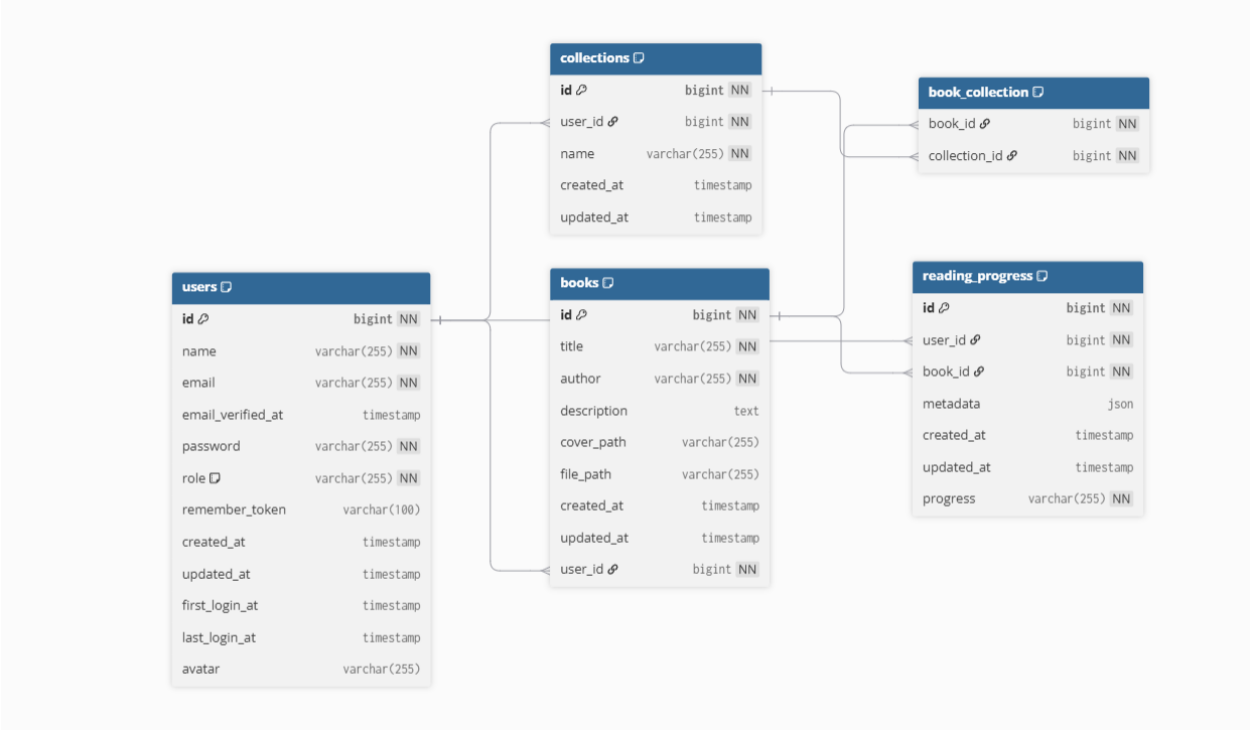


Рисунок 3.1 – Діаграма бази даних

Таблиця books

Основна таблиця системи, що містить інформацію про електронні книги:

– title – назва книги;

– author – автор;

– description – опис;

– cover_path – шлях до обкладинки;

– file_path – шлях до файлу книги;

– user_id – власник книги.

Таблиця реалізує зв’язок типу **one-to-many** з користувачами.

Таблиця collections

Призначена для групування книг у користувацькі колекції:

- name – назва колекції;
- user_id – власник колекції.

book_collection – реалізує зв’язок **many-to-many** між книгами та колекціями.

Додаткові сутності

reading_progress – Зберігає прогрес читання користувача:

- progress – поточний стан читання;
- metadata – додаткові дані у форматі JSON.

3.3.2 Службові таблиці Laravel

У системі також використовуються стандартні таблиці Laravel (рис. 3.2):

- **sessions –** сесії користувачів;
- **cache**, **cache_locks –** кешування;
- **jobs**, **job_batches –** черги завдань;
- **failed_jobs –** обробка помилок задач;
- **personal_access_tokens –** API токени;
- **migrations –** історія міграцій.

3.3.3. Програмні модулі системи

Програмна реалізація системи поділена на логічні модулі відповідно до MVC-архітектури Laravel.

Модуль управління книгами

Відповідає за:

- додавання книг;
- редагування;

– видалення;
– перегляд списку книг;
– завантаження файлів і обкладинок.

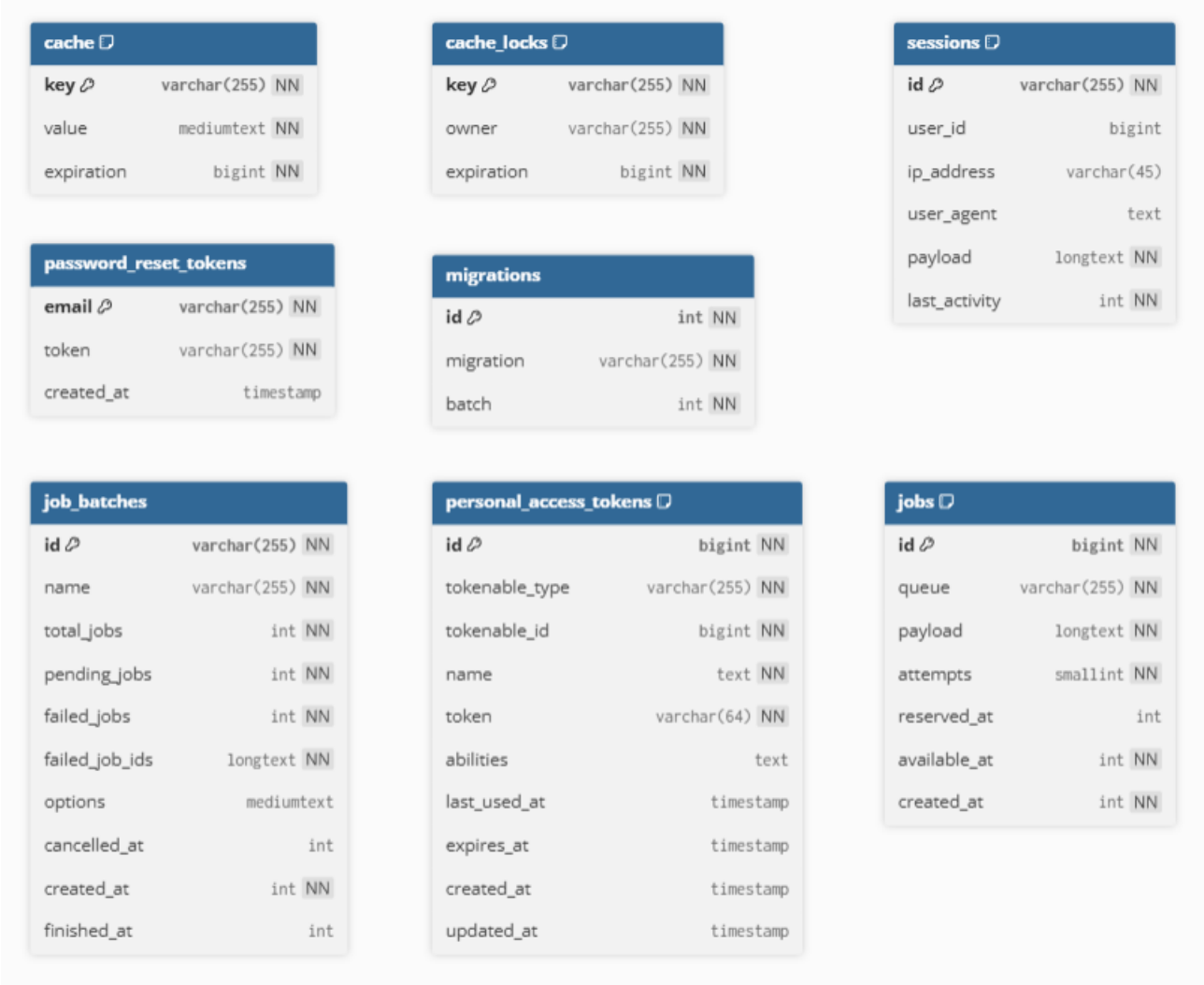


Рисунок 3.2 – Схема стандартних таблиць Laravel

Модуль колекцій

Забезпечує:

– створення колекцій;
– додавання/видалення книг з колекцій;
– відображення структурованих груп книг.

Модуль обробки файлу

Відповідає за зчитування та зберігання метаданих різних типів

Модуль прогресу читання

Відповідає за збереження поточної позиції читання користувача.

Модуль авторизації

Забезпечує:

- реєстрацію користувачів;
- вхід у систему;
- розмежування ролей.

Реалізована структура бази даних забезпечує повну підтримку функціональних можливостей системи та дозволяє ефективно зберігати та обробляти інформацію про електронні книги. Використання Laravel та ORM Eloquent спростило роботу з даними та забезпечило гнучкість і масштабованість програмної реалізації.

3.4 Аналіз результатів та перевірка працездатності системи

Після реалізації інформаційно-довідкової системи «Особиста бібліотека електронних книг» було проведено перевірку її працездатності та тестування основних функціональних модулів. Метою тестування є підтвердження коректності роботи системи, відповідності реалізованого функціоналу поставленим вимогам та стабільності обробки даних.

У процесі тестування використовувалися наступні методи перевірки системи:

- перевірка коректності CRUD-операцій;
- тестування завантаження файлів;
- перевірка роботи зв’язків між таблицями бази даних;
- перевірка користувацького інтерфейсу.

3.4.1 Перевірка основних функцій системи

Реєстрація та авторизація користувача

Було перевірено коректність створення облікового запису та входу в систему (рис. 3.3). Система успішно виконує валідацію даних та забезпечує доступ відповідно до ролей користувачів.

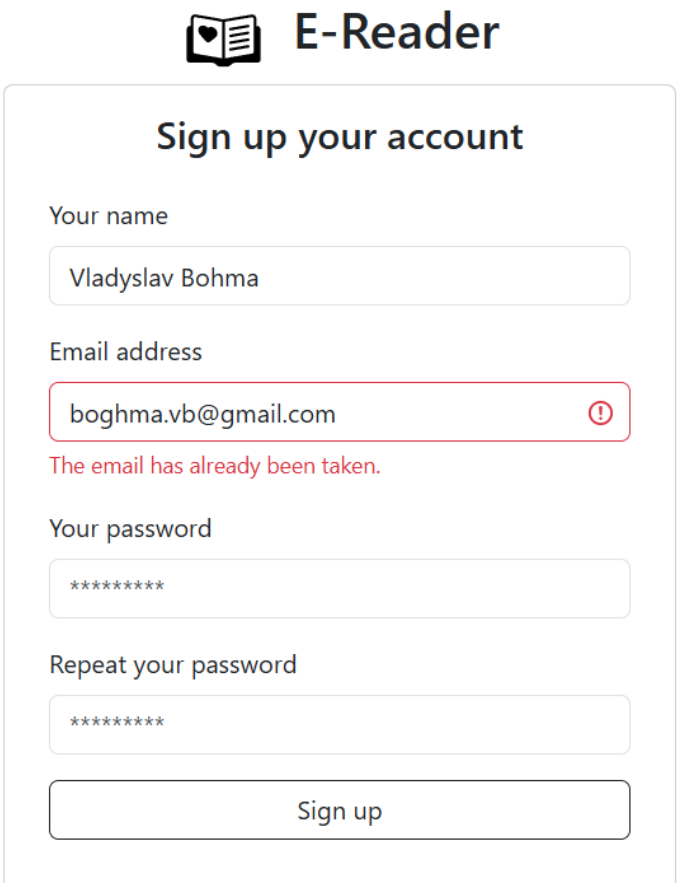


Рисунок 3.3 – Приклад валідації даних сторінки створення акаунту

Додавання та управління книгами

Перевірено функціонал створення, редагування та видалення книг (рис. 3.4) різних форматів (Додаток А). Дані коректно зберігаються у базі даних, а файли книг та обкладинок успішно завантажуються у файлове сховище. Також перевірено функціонал автозаповнення метаданих при додаванні та редагуванні книги (Додаток Б).

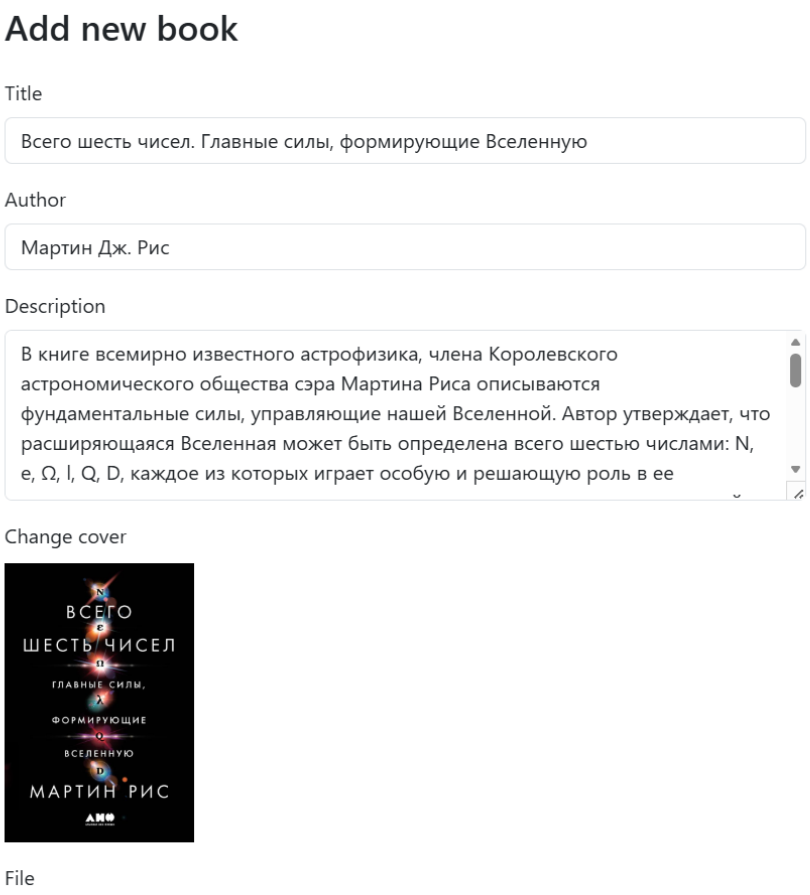


Рисунок 3.4 – Форма додавання книги

Завантаження книг через ZIP-архів

Система коректно обробляє завантаження ZIP-архівів (рис. 3.5–3.6), виконує їх розпакування та автоматично створює записи для кожної книги.

Читання книги

Перевірено читання книг різних форматів та збереження прогресу читання (рис. 3.7). Дані коректно зберігаються в базу даних та відображаються при повторному відкритті книги (рис. 3.8).

Робота з колекціями

Перевірено створення колекцій, додавання до них книг (рис. 3.9–3.10) та завантаження колекції. Зв’язки між сутностями у базі даних працюють коректно (many-to-many).

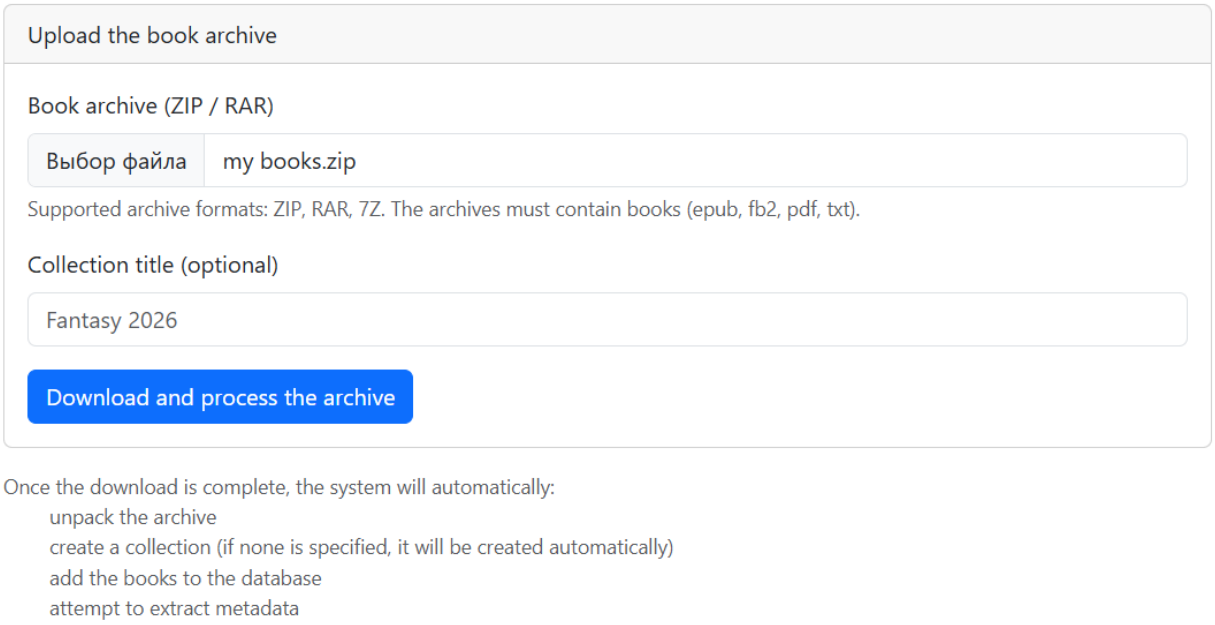


Рисунок 3.5 – Форма завантаження архіву

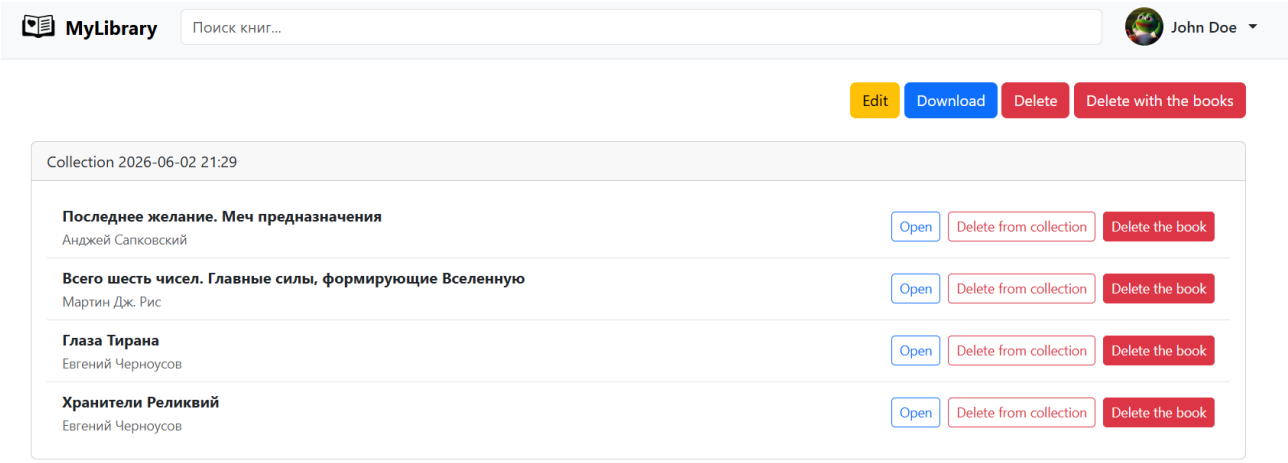


Рисунок 3.6 – Сторінка створення колекції

Пошук

Функція пошуку дозволяє знаходити книги за назвою та автором. Результати відображаються коректно та відповідають введеним критеріям (рис 3.11–3.12).

Результати тестування показали, що всі основні функції системи працюють коректно та відповідають поставленим вимогам. База даних

забезпечує цілісність даних, а програмна реалізація гарантує стабільну взаємодію між компонентами системи.

```
public function saveProgress(Request $request, Book $book)
    {
        $request->validate([
            'progress' => 'required|string'
        ]);

        ReadingProgress::updateOrCreate(
            [
                'user_id' => $request->user()->id,
                'book_id' => $book->id,
            ],
            [
                'progress' => $request->progress,
            ]
        );

        return response()->json([
            'success' => true
        ]);
    }
```

Рисунок 3.7 – Збереження прогресу на сервері

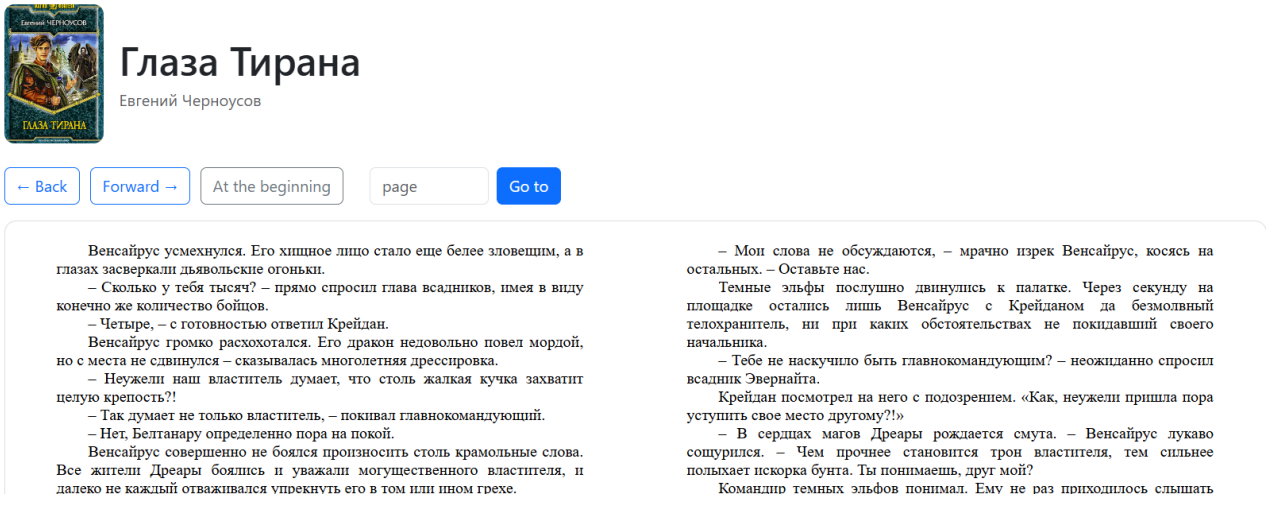


Рисунок 3.8 – Сторінка книги

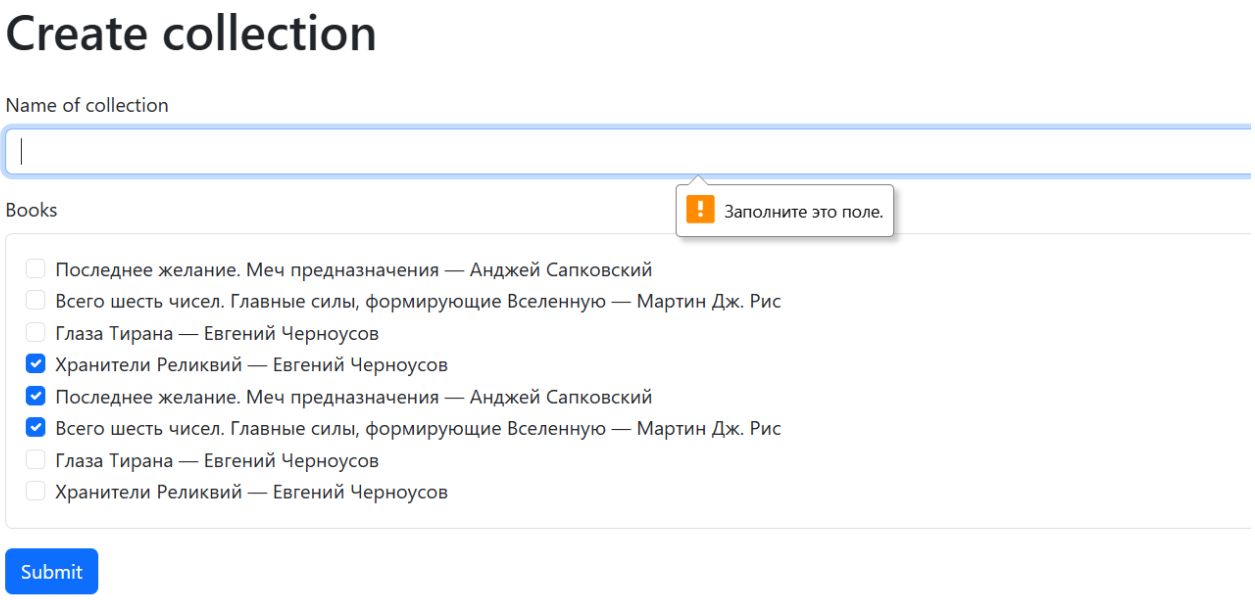


Рисунок 3.9 – сторінка створення колекції

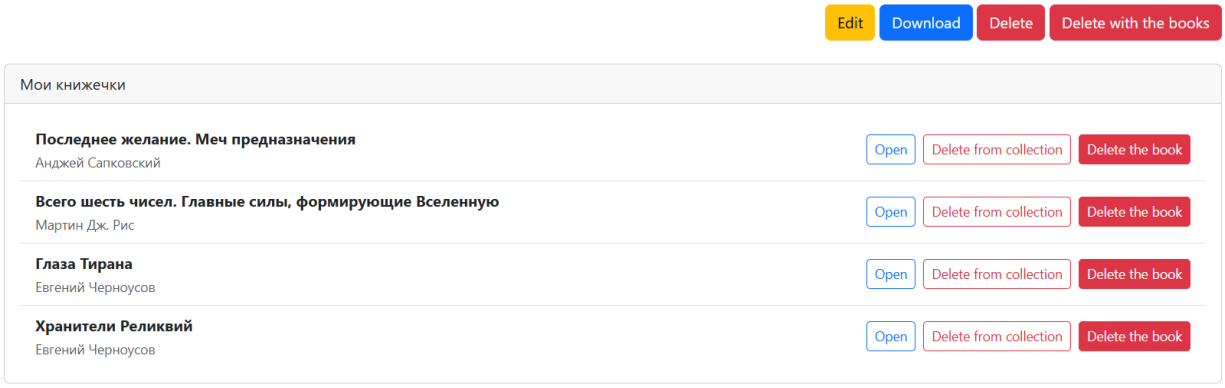


Рисунок 3.10 – сторінка колекції

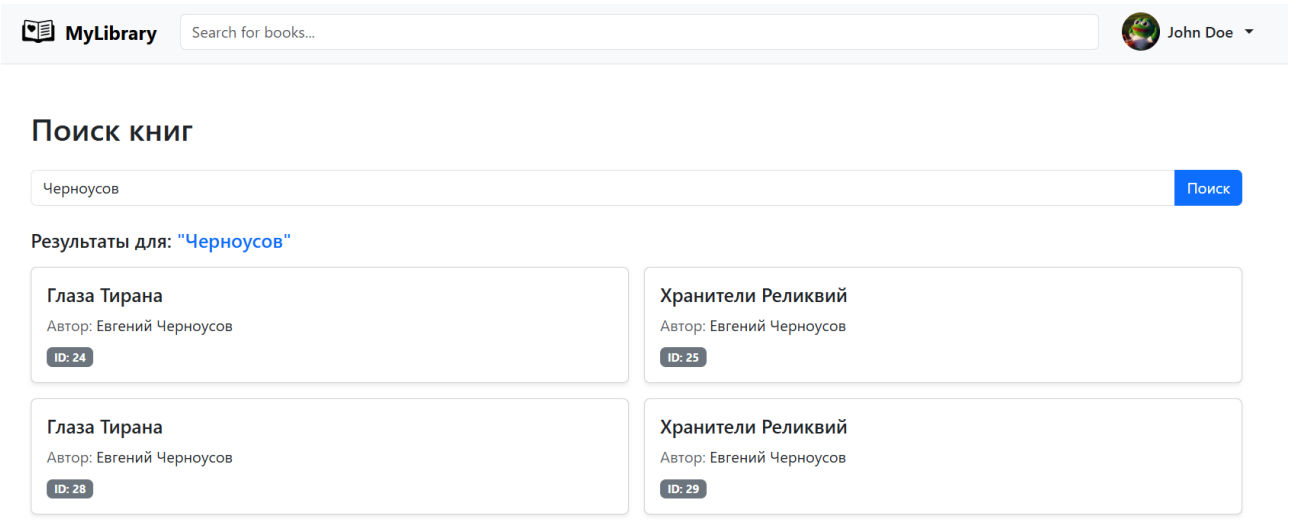


Рисунок 3.11 – Сторінка пошуку

Система демонструє високу стабільність при виконанні операцій з великим обсягом даних, включаючи масове завантаження книг через ZIP-архів.

```
public function search(Request $request)
{
    $search = $request->q;

    $books = Book::query()
        ->when($search, function ($query) use ($search) {

            $terms = explode(' ', $search);

            $query->where(function ($q) use ($terms) {
                foreach ($terms as $term) {
                    $q->orWhere('title', 'LIKE', "%{$term}%")
                    ->orWhere('author', 'LIKE', "%{$term}%");
                }
            });

        })
        ->get();
    return view('books.search', [
        'books' => $books,
        'user' => $request->user(),
    ]);
}
```

Рисунок 3.12 – Метод пошуку

3.5 Практична реалізація та приклади впровадження

Розроблена інформаційно-довідкова система «Особиста бібліотека електронних книг» може бути використана як персональний інструмент для

організації та управління електронними книгами. Система забезпечує зручне зберігання, пошук та структуризацію книжкових ресурсів, а також підтримує роботу з користувацькими колекціями та прогресом читання.

Практична реалізація системи передбачає її використання на локальному сервері або веб-хостингу з доступом через браузер. Завдяки використанню веб-орієнтованої архітектури користувач може працювати з бібліотекою з будь-якого пристрою, що має доступ до інтернету.

Можливості практичного використання

Система може застосовуватися у таких випадках:

- формування особистої електронної бібліотеки користувача;
- систематизація навчальних матеріалів та книг;
- створення тематичних колекцій (наприклад, навчання, художня література, професійна література);
- відстеження прогресу читання.

Перспективи розвитку системи:

- інтеграцію повнотекстового пошуку по вмісту книг;
- додавання мобільної версії або SPA-інтерфейсу;
- впровадження рекомендаційної системи на основі уподобань користувача;
- система відгуків та нотаток;
- синхронізацію з хмарними сховищами;
- підтримку багатокористувацьких бібліотек (спільні колекції).

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено інформаційно-довідкову систему «Особиста бібліотека електронних книг», призначену для зберігання, організації та управління електронними книгами користувача.

У процесі дослідження було проаналізовано існуючі підходи до побудови електронних бібліотек та інформаційно-довідкових систем, що дозволило обґрунтувати вибір сучасних веб-технологій для реалізації програмного продукту.

На етапі проєктування було розроблено архітектуру системи на основі MVC-підходу, визначено основні модулі та взаємодію між ними. Також було спроєктовано структуру бази даних, яка забезпечує зберігання інформації про користувачів, книги, колекції, жанри, відгуки, нотатки та прогрес читання.

У практичній частині роботи реалізовано програмну систему з використанням фреймворку Laravel та СУБД MySQL. Було розроблено основні функціональні модулі, зокрема управління книгами, роботу з колекціями, систему пошуку, відгуків і нотаток, а також механізм збереження прогресу читання. Окремо реалізовано функцію масового завантаження книг через ZIP-архів із автоматичним створенням структур колекцій.

Проведене тестування підтвердило коректність роботи всіх основних функцій системи, стабільність взаємодії між компонентами та цілісність збереження даних. Результати перевірки свідчать про відповідність розробленої системи поставленим вимогам.

Отримана інформаційно-довідкова система може використовуватися як персональна електронна бібліотека, а також як інструмент для систематизації навчальних та літературних матеріалів. Завдяки веб-орієнтованій архітектурі система є зручною у використанні та доступною з різних пристроїв.

Подальший розвиток системи може бути спрямований на розширення функціональності, зокрема впровадження повнотекстового пошуку,

рекомендаційної системи, мобільної версії та інтеграції з хмарними сервісами зберігання даних.

Таким чином, мета дипломної роботи досягнута, а поставлені завдання виконані у повному обсязі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Laravel Documentation. Laravel. URL : https://laravel.com/docs/13.x/ (дата звернення : 15.05.2026).

2. PHP Documentation. PHP. URL : https://www.php.net/docs.php (дата звернення : 15.05.2026).

3. Docker Documentation. Docker. URL : https://docs.docker.com/ (дата звернення : 15.05.2026).

4. Bootstrap 5 Documentation. Bootstrap. URL : https://getbootstrap.com/docs/5.2/getting-started/introduction/ (дата звернення : 15.05.2026).

5. MySQL. MySQL. URL : https://www.mysql.com/ (дата звернення : 15.05.2026).

6. Blade Templates. Laravel Documentation. URL : https://laravel-docs.readthedocs.io/en/latest/blade/ (дата звернення : 15.05.2026).

7. HTML : HyperText Markup Language. MDN Web Docs. URL : https://developer.mozilla.org/en-US/docs/Web/HTML (дата звернення : 15.05.2026).

8. CSS : Cascading Style Sheets. MDN Web Docs. URL : https://developer.mozilla.org/en-US/docs/Web/CSS (дата звернення : 15.05.2026).

9. Bohma Vladyslav. Репозиторій проєкту. GitHub. URL : https://github.com/bohmaVladyslav/dissertation (дата звернення : 15.05.2026).

ДОДАТОК А

Обробка книги різних форматів на прикладі сервісу EPUB

```
class EpubService
{
    protected \ZipArchive $zip;
    protected string $tmpPath;

    public function extract(string $filePath): array
    {
        $this->zip = new \ZipArchive();

        if ($this->zip->open($filePath) !== true) {
            throw new \Exception("Cannot open EPUB file: {$filePath}");
        }

        $opfPath = $this->getOpfPath();
        $opfXml  = $this->getFile($opfPath);

        $opf = new SimpleXMLElement($opfXml);

        $metadata = $this->parseMetadata($opf);
        $manifest = $this->parseManifest($opf);

        $coverPath = $this->extractCover($opfPath, $opf, $manifest, $filePath);

        $this->zip->close();
```

```
    $metadata['cover'] = $coverPath;

    return $metadata;
}

protected function getOpfPath(): string
{
    $containerXml = $this->getFile('META-INF/container.xml');
    $xml = new SimpleXMLElement($containerXml);
    return (string) $xml->rootfiles->rootfile['full-path'];
}
protected function getFile(string $path): string

{
    $content = $this->zip->getFromName($path);
    if ($content === false) {
        throw new \Exception("File not found in EPUB: {$path}");
    }
    return $content;
}

protected function parseMetadata(SimpleXMLElement $opf): array
{
    $ns = $opf->getNamespaces(true);
    $metadata = $opf->metadata;

    $dc = $metadata->children($ns['dc'] ?? null);

    // short description (dc:description)
    $shortDescription = (string) ($dc->description ?? '');
```

```
        // long description (meta property="se:long-description")
        $longDescription = null;


    foreach ($metadata->meta as $meta) {
            $attributes = $meta->attributes();

            if (
                isset($attributes['property']) &&
                (string)$attributes['property'] === 'se:long-description'
            ) {
                $longDescription = (string) $meta;
                break;
            }
        }

        return [
            'title' => (string) ($dc->title ?? 'Unknown'),
            'author' => (string) ($dc->creator ?? 'Unknown'),
            'description' => $longDescription ?: $shortDescription ?: 'Unknown',
        ];
    }

    protected function parseManifest(SimpleXMLElement $opf): array
    {
        $manifest = [];

        foreach ($opf->manifest->item as $item) {
            $manifest[(string)$item['id']] = (string)$item['href'];
        }
```

```
        return $manifest;
    }

    protected function extractCover(
        string $opfPath,
        SimpleXMLElement $opf,
        array $manifest,
        string $epubFile
    ): ?string {
        $coverId = null;


        foreach ($opf->metadata->meta as $meta) {
            if ((string)$meta['name'] === 'cover') {
                $coverId = (string)$meta['content'];
                break;
            }
        }

        if (!$coverId || !isset($manifest[$coverId])) {
            return null;
        }

        $coverRelativePath = $this->resolvePath($opfPath,
$manifest[$coverId]);
        $imageBinary = $this->zip->getFromName($coverRelativePath);

        if (!$imageBinary) {
            return null;
```

```
        }

        $fileName = 'covers/' . Str::random(20) . '.jpg';

        Storage::disk('public')->put($fileName, $imageBinary);

        return $fileName;
    }

    protected function resolvePath(string $opfPath, string $resource): string
    {
        $dir = dirname($opfPath);

        if ($dir === '.') {
            return $resource;
        }

        return $dir . '/' . $resource;
    }
}
```

ДОДАТОК Б

Метод автозаповнення метаданих

```
    protected function extract(string $fullPath)
    {
        $meta = null;
        $fileExtension = strtolower(pathinfo($fullPath, PATHINFO_EXTENSION));

        switch ($fileExtension) {
            case 'txt':
                $meta = [
                    'title' => strtolower(pathinfo($fullPath, PATHINFO_BASENAME)),
                    'author' => null,
                    'description' => null,
                    'cover' => null,
                ];
                break;
            case 'pdf':
                $meta = app(PdfService::class)->extract($fullPath);
                break;

            case 'epub':
                $meta = app(EpubService::class)->extract($fullPath);
                break;
            case 'fb2':
```

```
                $meta = app(Fb2Service::class)->extract($fullPath);
                break;
            default:
                Log::channel('info_file')->info('Unsupported file format: ' .
$fullPath);
                abort(415, 'Unsupported file format');
        }

        return $meta;
    }
```