

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ
ЕЛЕКТРОННОЇ БІБЛІОТЕКИ
SOFTWARE FOR AN INFORMATION SYSTEM OF
ELECTRONIC LIBRARY

Виконав(ла): студент(ка) 4 курсу, групи КНТ-212сп
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

ПАСІЧЕНКО Н.С.

(ПРИЗВИЩЕ та ініціали)

Керівник ІЛЛЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ПАСІЧЕНКА Нікіти Сергійовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення інформаційної системи електронної бібліотеки. Software for an Information System of Electronic Library

керівник проєкту (роботи) к.т.н., доцент, ІЛЬЯШЕНКО Матвій Борисович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 03 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ІЛґЯШЕНКО М.Б., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “28” квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Нікіта ПАСІЧЕНКО
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Матвій ІЛґЯШЕНКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
88 с., 3 табл., 34 рис., 3 дод., 21 джерело.

JAVASCRIPT, REACT, ВЕБЗАСТОСУНОК, ЕЛЕКТРОННА
БІБЛІОТЕКА, ІНФОРМАЦІЙНА СИСТЕМА, МОВА ПРОГРАМУВАННЯ.

Об'єкт дослідження – процес розробки інформаційної системи для бібліотек.

Предмет дослідження – інформаційні системи електронної бібліотеки.

Мета роботи – розробка програмного забезпечення інформаційної системи електронної бібліотеки.

Матеріали, методи та технічні засоби: мова програмування JavaScript, бібліотека для побудови інтерфейсів React.

Результати. Розроблено проєктні рішення для створення програмного забезпечення інформаційної системи електронної бібліотеки. Створено програмне забезпечення інформаційної системи електронної бібліотеки. Проведено тестування розробленого програмного забезпечення інформаційної системи електронної бібліотеки.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення інформаційної системи електронної бібліотеки за допомогою мови програмування JavaScript та бібліотеки для побудови інтерфейсів React.

Галузь використання – електронні бібліотеки.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 88 pages, 3 tables, 34 figures, 3 appendixes, 21 sources.

JAVASCRIPT, REACT, WEB APPLICATION, ELECTRONIC LIBRARY, INFORMATION SYSTEM, PROGRAMMING LANGUAGE.

The object of research is the process of developing information system for libraries.

The subject of the research is information systems of the electronic library.

The purpose of this work is to develop the software of the information system of the electronic library.

Materials, methods and technical tools: JavaScript programming language, React library for building interfaces.

Results. Project solutions for the creation of software for the information system of the electronic library been designed. The software of the information system of the electronic library has been developed. The developed software of the information system of the electronic library has been tested.

Conclusions. The goal of the work has been achieved. The software tools of information system for the electronic library using the JavaScript programming language and the React library for building interfaces have been developed.

Scope of use – electronic libraries.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Інформаційні системи для підтримки діяльності бібліотек	11
1.2 Аналіз інформаційних систем електронних бібліотек.....	21
1.3 Висновки за розділом 1	30
2 Матеріали і методи	32
2.1 Вибір мови програмування	32
2.2 Вибір бібліотеки для розробки інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки.....	36
2.3 Висновки за розділом 2	38
3 Опис програми	40
3.1 Структура програмного забезпечення інформаційної системи електронної бібліотеки.....	40
3.2 Функціонування програмного забезпечення інформаційної системи електронної бібліотеки.....	42
3.3 Розробка бази даних програмного забезпечення інформаційної системи електронної бібліотеки.....	46
3.4 Проєктування інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки.....	48
3.5 Висновки за розділом 3	52
4 Експлуатація, тестування та експериментальне дослідження програми.....	54
4.1 Призначення й умови застосування програми	54
4.2 Характеристики програмного забезпечення інформаційної системи електронної бібліотеки.....	55
4.3 Інструкція по експлуатації програми.....	56
4.3.1 Звернення до програми.....	56
4.3.2 Вхідні й вихідні дані	56

4.3.3 Повідомлення.....	56
4.4 Виконання програмного забезпечення інформаційної системи електронної бібліотеки.....	56
4.5 Тестування програмного забезпечення інформаційної системи електронної бібліотеки.....	61
4.6 Висновки за розділом 4	61
Висновки	62
Перелік джерел посилань	65
Додаток А Технічне завдання	67
Додаток Б Фрагмент тексту програми	71
Додаток В Слайди презентації	83

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

БД – база даних;

ЕБ – електронна бібліотека;

ІС – інформаційна система;

ПЗ – програмне забезпечення.

ВСТУП

Актуальність програмного забезпечення інформаційної системи електронної бібліотеки обумовлена різними чинниками.

Електронна бібліотека дозволяє користувачам отримувати доступ до знань у будь-який час і з будь-якого місця за наявності Інтернет з'єднання. Це особливо важливо для користувачів, яким потрібен оперативний доступ до різних джерел інформації [1]-[6].

Інформаційні системи електронної бібліотеки забезпечують швидкий пошук інформації, автоматизоване сортування та зручну навігацію, що суттєво скорочує час, необхідний для пошуку потрібних матеріалів. Крім того, цифровий формат мінімізує потребу в фізичних ресурсах, таких як папір, що є важливим аспектом в умовах екологічної свідомості [1]-[7].

Програмне забезпечення інформаційних систем електронних бібліотек надає можливість налаштовувати користувацький інтерфейс, створювати власні бібліотеки, списки бажань і рекомендації, що підвищує ефективність навчання та дослідження [1]-[7].

Проте деякі програмні засоби інформаційних систем електронних бібліотек є досить дорогими для придбання, впровадження та підтримки, що може бути значною проблемою, особливо для невеликих бібліотек. Інші інформаційні системи електронних бібліотек є низько адаптованими для деяких груп користувачів, вони можуть бути недостатньо інтуїтивними або адаптованими для користувачів із обмеженими можливостями, що ускладнює їх використання певним категоріям людей. Деякі інформаційні системи електронних бібліотек можуть бути недостатньо гнучкими для масштабування, що створює труднощі в обслуговуванні зростаючої кількості користувачів та інформаційних ресурсів бібліотеки [1]-[7].

Тому актуальною є розробка інформаційної системи електронної бібліотеки. У дипломній кваліфікаційній роботі бакалавра розв'язується

актуальне завдання розробки програмного забезпечення інформаційної системи електронної бібліотеки.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та інформаційних систем для електронних бібліотек;
- здійснити проектування програмного забезпечення інформаційної системи електронної бібліотеки;
- створити програмне забезпечення інформаційної системи електронної бібліотеки;
- виконати тестування розробленого програмного забезпечення інформаційної системи електронної бібліотеки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інформаційні системи для підтримки діяльності бібліотек

Електронна бібліотека являє собою зібрання матеріалів у електронному форматі, доступ до яких здійснюється через Інтернет. На відміну від традиційних бібліотек, їй не потрібне фізичне місце для зберігання книг чи інших ресурсів. Усі матеріали в електронних бібліотеках зберігаються за допомогою відповідних інформаційних систем на серверах, що дозволяє користувачам отримувати онлайн доступ до матеріалів бібліотеки з будь-якої локації у світі [1]-[6].

Однією з ключових переваг електронних бібліотек є їхня доступність. Вони дозволяють отримати доступ до матеріалів з будь-якого куточка світу через Інтернет, що особливо зручно для тих, хто проживає у віддалених районах або не має змоги відвідувати фізичну бібліотеку. Ці бібліотеки працюють цілодобово, що позбавляє необхідності слідувати графіку роботи традиційних бібліотек чи вчасно повертати книги [1]-[6].

Ще однією важливою перевагою цифрових бібліотек є їхні потужні пошукові функції. Завдяки сучасним інструментам пошуку користувачі можуть швидко знаходити потрібну інформацію, що значно спрощує роботу дослідників та студентів. Крім того, електронні бібліотеки забезпечують доступ до матеріалів із будь-яких пристроїв, дозволяючи користувачам працювати з дому чи офісу, заощаджуючи час і кошти на поїздки та створюючи зручне середовище для навчання [1]-[7].

Цифрові бібліотеки також вирізняються своєю ефективністю у зберіганні матеріалів. Електронний формат значно простіший для зберігання та обслуговування порівняно з фізичними носіями, а також менш схильний до псування. Це забезпечує збереження важливої інформації для майбутніх поколінь. Оцифровування колекцій у бібліотеках по всьому світу відкриває

доступ до раніше недоступних матеріалів, сприяючи збереженню культурної спадщини [1]-[7].

Проте цифрові бібліотеки стикаються й із певними викликами. Одним із головних завдань є забезпечення збереження та цілісності цифрового контенту на тривалий час. Крім того, доступ до електронних матеріалів обмежується законами про авторське право, що може створювати перешкоди для користувачів [1]-[7].

Таким чином, електронні бібліотеки відіграють значну роль у наданні доступу до цифрових ресурсів для студентів, дослідників та інших користувачів. З розвитком технологій їхня важливість у зберіганні та поширенні знань лише зростатиме [1]-[7].

Електронні бібліотеки являють собою це сучасні цифрові сховища, що значно розширюють можливості традиційних бібліотек. Вони використовують новітні технології для збору, організації та розповсюдження великого обсягу інформації. Ці бібліотеки пропонують широкий спектр матеріалів, таких як електронні книги, наукові статті, мультимедійні файли, архіви та інші цифрові ресурси. Електронні бібліотеки забезпечують доступ до знань для всіх і сприяють ефективному використанню природних ресурсів. Вони також включають у себе потужні інструменти, системи метаданих та платформи для спільної роботи з інформацією. Концепція цифрової бібліотеки надає унікальну можливість розглядати, як технології можуть бути використані для управління знаннями та інформацією [1]-[7].

Популярність цифрових бібліотек зростає, оскільки дедалі більше людей використовують Інтернет для навчання, досліджень і читання. Вони мають низку переваг у порівнянні з традиційними бібліотеками. Цифрові бібліотеки доступні з будь-якої точки світу за наявності підключення до Інтернету, що особливо зручно для людей у віддалених районах або тих, хто не має змоги відвідувати фізичну бібліотеку. Їх можна використовувати цілодобово, без обмежень за графіком роботи або необхідності вчасно повертати книги. Завдяки потужним пошуковим системам цифрові бібліотеки

дозволяють швидко знаходити потрібну інформацію, що значно економить час. Водночас електронні матеріали є зручними для зберігання, хоча й можуть із часом піддаватися ризикам пошкодження [1]-[7].

Таким чином, цифрові бібліотеки є важливим ресурсом для дослідників та інших користувачів, хто прагне отримати швидкий і легкий доступ до інформації [1]-[7].

Інформаційні системи електронних бібліотек пропонують широкий спектр функцій, які сприяють цифровому навчанню та забезпечують доступ до різноманітних ресурсів. Ці системи створюються міждисциплінарними командами, які працюють над організацією та управлінням цифровим контентом [1]-[7].

Інформаційні системи електронних бібліотек містять колекції цифрових матеріалів, таких як електронні книги, журнали, зображення, відео та аудіо. Для полегшення доступу до цих матеріалів використовуються метадані — структурована інформація, яка містить заголовки, імена авторів, анотації та ключові слова. Завдяки цьому користувачі можуть швидко знайти потрібний матеріал [1]-[7].

Зберігання цифрових ресурсів забезпечується за допомогою сучасних систем зберігання даних, які дозволяють працювати з великими обсягами інформації. Для забезпечення тривалого доступу до цих ресурсів застосовуються методи цифрового збереження, які включають архівування, передачу та пошук даних [1]-[7].

Інтерфейси інформаційних систем електронних бібліотек створюються таким чином, щоб забезпечити користувачам зручну та інтуїтивно зрозумілу взаємодію з цифровими ресурсами. Для забезпечення безпеки доступу до ресурсів використовуються механізми ідентифікації та авторизації, які дозволяють ідентифікувати користувачів бібліотеки та надавати їм відповідні дозволи [1]-[7].

Ефективність пошуку матеріалів в інформаційних системах електронних бібліотек досягається за рахунок впровадження алгоритмів

пошуку та методів сортування, що дозволяють швидко знаходити потрібний контент. Водночас дотримання авторських прав і ліцензійних угод гарантує законність використання цифрових матеріалів [1]-[7].

Для покращення роботи інформаційних систем електронних бібліотек використовуються інструменти аналітики. Вони дають змогу аналізувати поведінку користувачів, рівень використання матеріалів і взаємодії з системою. Це дозволяє вдосконалювати функціональність бібліотеки та підвищувати її зручність для користувачів [1]-[7].

Інформаційні системи електронних бібліотек охоплюють різні аспекти, починаючи з отримання контенту і закінчуючи його організацією, зберіганням, доступом і розповсюдженням [1]-[7].

Інформаційні системи цифрових бібліотек пропонують електронні книги, наукові статті, мультимедійні матеріали та архіви. Контент може бути отриманий через придбання, підписки, співпрацю з провайдерами або ліцензування. Метадані є основою для організації цифрових матеріалів. Вони містять інформацію про назву, автора, ключові слова та тематику. Інформаційні спеціалісти та бібліотекарі створюють і систематизують метадані, забезпечуючи їх узгодженість і доступність [1]-[7].

Для збереження цифрових ресурсів в інформаційних системах електронних бібліотек використовуються надійні сховища, які запобігають втратам даних і забезпечують довготривале зберігання. Важливим є впровадження технологій резервного копіювання, архівування та методів забезпечення цілісності даних [1]-[7].

Інформаційні системи цифрових бібліотек пропонують системи індексування, які покращують доступність матеріалів, використовуючи метадані. Завдяки розвинутим алгоритмам пошуку користувачі можуть швидко знаходити потрібну інформацію [1]-[7].

Контроль доступу до інформаційних систем електронних бібліотек передбачає авторизацію користувачів через використання імен та паролів або

інших методі. Це гарантує, що ресурси будуть доступні лише авторизованим особам [1]-[7].

Зручний інтерфейс інформаційної системи електронної бібліотеки забезпечує легку навігацію, розширені можливості пошуку і доступ до персоналізованих інструментів. Інтерактивні функції сприяють кращій взаємодії користувача з ресурсами [1]-[7]. Такі багатогранні можливості роблять інформаційні системи електронних бібліотек незамінними інструментами для доступу до знань і підтримки навчання [1]-[7].

Використання сучасних інструментів аналізу дозволяє оцінювати ефективність і продуктивність впроваджених стратегій. Статистичні дані допомагають ухвалювати обґрунтовані рішення щодо розвитку бібліотечної колекції та оптимізації способів поширення матеріалів [1]-[7].

Електронні бібліотеки потребують регулярного оновлення: додаються нові ресурси, видаляється застарілий контент, оновлюються метадані. Також важливим є постійне технічне обслуговування, яке забезпечує актуальність, цілісність і надійність збережених даних [1]-[7].

Для розширення своїх можливостей цифрові бібліотеки взаємодіють з іншими установами, постачальниками контенту та дослідниками. Такі партнерства дозволяють урізноманітнити колекції. Проведення інформаційних заходів і навчальних програм сприяє популяризації цифрових бібліотек і їхніх послуг [1]-[7].

Проаналізуємо переваги використання інформаційних технологій електронних бібліотек. Перехід від фізичних носіїв до цифрових платформ суттєво спростив зберігання, доступ і обробку інформації. Електронні бібліотеки мають низку переваг порівняно з традиційними. Доступність без кордонів: цифрові бібліотеки надають можливість отримати доступ до матеріалів у будь-який час і з будь-якого місця. Це особливо зручно для студентів, дослідників і звичайних користувачів [1]-[7].

Електронні ресурси цифрових бібліотек охоплюють широкий спектр матеріалів: від книг і статей до мультимедійних даних та архівів. Це створює

більш гнучке навчальне середовище для користувачів із різними потребами [1]-[7].

Функції пошуку та індексації інформаційних систем електронних бібліотек допомагають швидко знаходити необхідну інформацію, значно заощаджуючи час порівняно з традиційними бібліотеками [1]-[7].

Цифрові платформи знижують витрати на зберігання, обслуговування фізичних колекцій і друк. Це дозволяє організаціям оптимальніше розподіляти свої ресурси. У цифрових бібліотеках відсутня потреба у великих фізичних сховищах, що робить їх більш гнучкими й економними. Електронні платформи інтегрують різноманітні формати, як-от відео, аудіо та інтерактивні елементи, що покращує сприйняття інформації [1]-[7].

Матеріали цифрових бібліотек можуть оперативно оновлюватися, гарантуючи користувачам доступ до актуальної інформації. Електронні бібліотеки активно підтримують дистанційне навчання, сприяючи співпраці між викладачами, студентами та дослідниками з усього світу. Зменшення використання фізичних носіїв і паперових ресурсів дозволяє електронним бібліотекам сприяти збереженню навколишнього середовища. Такий підхід підтримує глобальні ініціативи екологічної відповідальності, зокрема у сфері управління інформацією [1]-[7].

Цифрові бібліотеки пропонують можливість створення персоналізованих колекцій, забезпечення збереження приватної інформації та налаштування пошукових систем для надання індивідуальних рекомендацій. Це сприяє підвищенню рівня задоволення користувачів і створює комфортні умови для навчання [1]-[7].

Механізми безпеки, такі як резервне копіювання та цифрове страхування, гарантують збереження та захист цифрових ресурсів. Вони забезпечують захист від втрати інформації, пошкодження даних чи несанкціонованого доступу [1]-[7].

Електронні бібліотеки можуть бути інтегровані з системами управління навчанням, створюючи єдину платформу для використання освітніх ресурсів.

Це значно покращує взаємодію викладачів і студентів із цифровим контентом [1]-[7].

Цифрові бібліотеки сприяють поширенню інформації на глобальному рівні, забезпечуючи доступ до міжнародних ресурсів. Вони сприяють популяризації різних точок зору, підтримуючи ідею відкритого й об'єднаного суспільства [1]-[7]. Електронні бібліотеки зробили наукові журнали доступними для ширшого кола користувачів. Це зменшило необхідність особистих контактів, спростило доступ до результатів досліджень і сприяло прискоренню наукового прогресу [1]-[7].

Сучасні бібліотеки активно впроваджують інформаційні технології аналізу даних, штучний інтелект та алгоритми машинного навчання. Це дозволяє покращити ефективність роботи бібліотек і створити нові можливості для користувачів [1]-[7].

Розвиток електронних бібліотек відкриває нові горизонти в отриманні знань. Вони відповідають сучасним запитам суспільства, сприяють екологічній стабільності, підтримують інклюзію та технологічний прогрес. У цифрову епоху такі бібліотеки стають важливим інструментом для розповсюдження знань і забезпечення якісного навчання [1]-[7].

Електронні бібліотеки є ключовим елементом у підтримці онлайн освіти, забезпечуючи викладачів і студентів доступом до необхідних ресурсів. Вони сприяють різноманітним стилям навчання, підвищують доступність, та підтримують співпрацю. Цифрові бібліотеки забезпечують доступ до широкого спектра матеріалів, включаючи електронні книги, наукові статті, мультимедійний контент та інші освітні ресурси. Такий вибір дозволяє задовольнити потреби студентів із різних дисциплін і рівнів освіти [1]-[7].

Суттєвою перевагою використання інформаційних систем електронних бібліотек є можливість використовувати їх у будь-який час і в будь-якому місці. Це дозволяє студентам гнучко планувати навчання, враховуючи свій графік і час доби [1]-[7].

Електронні бібліотеки забезпечують навчальний простір для взаємодії студентів і викладачів, навіть якщо вони знаходяться на великій відстані один від одного. Це особливо корисно для дистанційного навчання, коли фізичний доступ до традиційних бібліотек обмежений [1]-[7].

Інтегровані системи пошуку та інструменти для навігації дозволяють легко знаходити потрібну інформацію серед численних ресурсів. Це значно спрощує процес дослідження, економить час і підвищує ефективність навчання [1]-[7].

Електронні бібліотеки надають доступ до інтерактивного контенту, що покращує освітній процес. Такий мультимедійний підхід дозволяє задовольнити різні стилі навчання та робить матеріал більш доступним [1]-[7].

Віртуальні бібліотеки створюють простори для дискусій і співпраці, допомагаючи студентам і викладачам обмінюватися знаннями та працювати над спільними проєктами. Це сприяє формуванню спільноти й активній взаємодії в онлайн середовищі [1]-[7].

Електронні бібліотеки можуть бути інтегровані із системами управління навчанням, що дозволяє викладачам і студентам використовувати навчальні матеріали безпосередньо у своїх курсах. Це забезпечує зручність і підвищує ефективність змішаного навчання [1]-[7].

Цифрові бібліотеки надають можливість користувачам створювати власні колекції, зберігати важливий контент і отримувати рекомендації на основі своїх інтересів. Такий персоналізований підхід допомагає враховувати індивідуальні потреби й робить навчання більш комфортним [1]-[7].

Цифрові бібліотеки стають невіддільною частиною сучасного навчання, надаючи широкий доступ до знань і забезпечуючи гнучкість та ефективність освітнього процесу [1]-[7].

Одним із ключових аспектів сучасного зберігання даних є інтеграція онлайн систем. Завдяки технологічним досягненням бібліотеки все частіше зосереджуються на невеликих проєктах, які реалізуються в окремих установах або групах бібліотек. Створення цифрових колекцій та забезпечення доступу

до них є викликом для кожної бібліотеки. Сьогодні основним інструментом для створення інформаційних систем електронних бібліотек є Інтернет і вебтехнології. Поширення Інтернету зробило цифрові бібліотеки більш популярними та затребуваними, ніж будь-коли раніше [1]-[7].

Поняття цифрова бібліотека може трактуватися по-різному в залежності від контексту. Зазвичай під цим терміном розуміють одне або кілька таких аспектів: архівування документів (наприклад, навчальні матеріали, реферати) для доступу через Інтернет, використання традиційних матеріалів бібліотек, таких як скановані зображення, фотографії чи тексти, для створення цифрового контенту, збір наукових даних, наприклад, про білкові чи нуклеїнові послідовності, організація мультимедійних бібліотек або баз даних програмного забезпечення, використання онлайн баз даних для роботи з мультимедійними даними, відео, книгами та статтями, використання пристроїв для зберігання даних, таких як оптичні носії, цифрові відеотеки або музичні архіви, наявність цифрових ресурсів, таких як аудіо, відео чи графічні матеріали [1]-[7].

Послуги та продукти, які надають цифрові бібліотеки, мають велику цінність завдяки своїй доступності та інтерактивності. Цифрова бібліотека — це не просто збірка електронних матеріалів, але й платформа, яка забезпечує широкий доступ до них. Експерти у галузі бібліотечної справи та інформаційних технологій описують цифрові бібліотеки як ключову складову, яка доповнює традиційні бібліотеки в умовах цифрового суспільства. Вони розширюють доступ до ресурсів, створюють нові способи пошуку інформації та зміцнюють зв'язки між користувачами та даними [1]-[7].

Таким чином, цифрові бібліотеки стають важливим елементом сучасної інформаційної екосистеми, сприяючи розвитку знань та розширюючи можливості доступу до них [1]-[7].

Хоча терміни цифрова бібліотека та електронна бібліотека часто вживаються як синоніми, поняття віртуальна бібліотека або бібліотека без стін означає набір цифрових мета ресурсів або порталів, які об'єднують контент,

доступний в електронному вигляді. Віртуальні бібліотеки дозволяють знаходити корисну інформацію безпосередньо в Інтернеті [1]-[7].

На відміну від традиційних електронних бібліотек, віртуальні бібліотеки зазвичай не пропонують повний спектр послуг, таких як друк матеріалів за запитом. Проте вони надають широкий доступ до онлайн ресурсів, наприклад, до довідкових каталогів, які забезпечують зручний пошук інформації. Розмір віртуальної бібліотеки може варіюватися: від величезних колекцій, пов'язаних із найбільшими бібліотеками світу, до невеликих ресурсів, що складаються лише зі сотень елементів [1]-[7].

Термін гібридна бібліотека стосується бібліотек, які поєднують традиційне зберігання ресурсів (на папері чи компакт-дисках) із цифровим зберіганням в електронному середовищі. Такі бібліотеки інтегрують паперові та електронні носії, створюючи місток між традиційними та цифровими підходами до управління інформацією [1]-[7].

Особливостями гібридних бібліотек є: інтеграція ресурсів (гібридні бібліотеки об'єднують паперові та електронні колекції, які можуть бути локальними або доступними через Інтернет), технологічна сумісність (одна з ключових проблем – забезпечення наскрізного шифрування та доступу до даних із різних форматів і пристроїв), багаторівневий доступ (гібридні бібліотеки створені для інтеграції різних технологій, що дозволяє взаємодіяти з локальними колекціями, а також із зовнішніми комерційними та некомерційними бібліотеками) [1]-[7].

Сучасні гібридні бібліотеки часто мають доступ до Інтернету, що дозволяє користувачам переглядати каталоги, проводити пошук інформації та виконувати інші завдання, наприклад, облік чи розподіл ресурсів. Однак доступ до ресурсів інших бібліотек може бути обмеженим, якщо вони належать до сторонніх організацій чи міжнародних установ [1]-[7].

Таким чином, гібридні бібліотеки є важливим кроком у напрямку інтеграції інформаційних технологій і забезпечення універсального доступу до

інформаційних ресурсів, поєднуючи переваги як традиційних, так і цифрових рішень [1]-[7].

Визначено, що електронна бібліотека являє собою зібрання матеріалів у електронному форматі, доступ до яких здійснюється через Інтернет. На відміну від традиційних бібліотек, їй не потрібне фізичне місце для зберігання книг чи інших ресурсів. Усі матеріали в електронних бібліотеках зберігаються за допомогою відповідних інформаційних систем на серверах, що дозволяє користувачам отримувати онлайн доступ до матеріалів бібліотеки з будь-якої локації у світі. Однією з ключових переваг електронних бібліотек є їхня доступність. Відзначено, що інформаційні системи електронних бібліотек дозволяють отримати доступ до матеріалів з будь-якого куточка світу через Інтернет, що особливо зручно для тих, хто проживає у віддалених районах або не має змоги відвідувати фізичну бібліотеку. Показано, що інформаційні системи електронних бібліотек охоплюють різні аспекти, починаючи з отримання контенту і закінчуючи його організацією, зберіганням, доступом і розповсюдженням. Проаналізовано переваги використання інформаційних технологій електронних бібліотек. Відзначено, що сучасні електронні бібліотеки активно впроваджують інформаційні технології аналізу даних, штучний інтелект та алгоритми машинного навчання, що дозволяє покращити ефективність роботи бібліотек і створити нові можливості для користувачів.

1.2 Аналіз інформаційних систем електронних бібліотек

Проаналізуємо інформаційні системи електронних бібліотек [8]-[16].

Програмне забезпечення Invenio (рис. 1.1) є платформою з відкритим вихідним кодом для управління бібліотеками та великими цифровими репозиторіями. Як сучасна інтегрована бібліотечна система, вона підтримує ключові функції, такі як розповсюдження, каталогізація, міжбібліотечний обмін, придбання книг тощо. Крім того, Invenio пропонує потужні засоби інтерфейсу прикладного програмування (Application Programming Interface,

API) для роботи з репозиторіями, системи автентифікації та засоби зберігання даних, що дозволяє створювати власні застосунки на її основі [8], [9], [14].

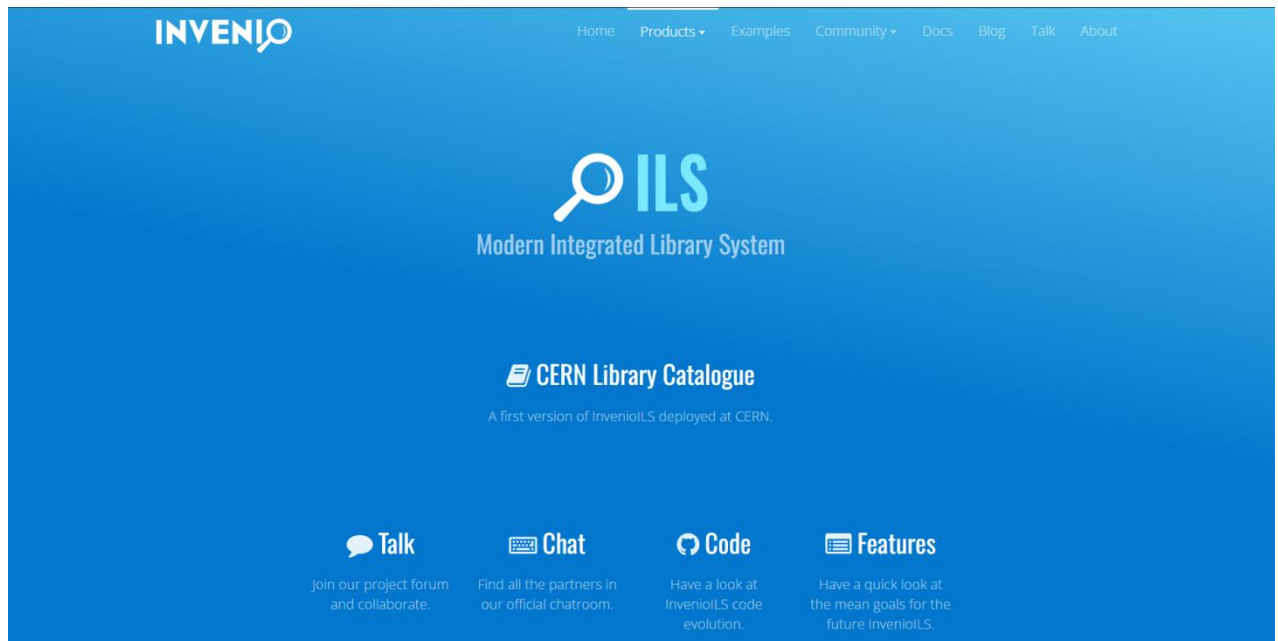


Рисунок 1.1 – Програмне забезпечення Invenio [14]

Програмне забезпечення Invenio побудовано на фреймворку Invenio, який базується на популярних інструментах з відкритим кодом, таких як Python і React. Завдяки цьому фреймворку користувачі з технічними знаннями мають необмежені можливості для налаштування та вдосконалення платформи під власні потреби [8], [9], [14].

Invenio є програмним забезпеченням з відкритим кодом, яке поширюється під безкоштовною ліцензією і пропонує потужні інструменти для управління бібліотечними процесами, такими як каталогізація, розповсюдження та придбання матеріалів. Воно має сучасний інтерфейс користувача, підтримує інтеграцію через REST API, використовує JSON-схеми для організації даних, а також надає ефективний бек-офіс із функціями пошуку, створення списків та перегляду деталей об'єктів. Гнучкі модулі програмного забезпечення Invenio для роботи з придбаннями та міжбібліотечними позиками спрощують управління ресурсами, а масштабованість платформи дозволяє створювати додатки, такі як цифрові репозиторії, системи управління даними та механізми пошуку [8], [9], [14].

Основними характеристиками програмного забезпечення Invenio є такі [8], [9], [14]:

- відкритий вихідний код та ліцензія: програмне забезпечення Invenio розповсюджується безкоштовно, забезпечується гнучкість у налаштуванні та адаптації платформи під індивідуальні потреби [8], [9], [14];

- інтегрована бібліотечна система: програмне забезпечення Invenio підтримує всі основні процеси управління бібліотекою, зокрема, розповсюдження ресурсів, каталогізація, придбання матеріалів, міжбібліотечний обмін [8], [9], [14];

- проста в інтеграції завдяки REST API [8], [9], [14];

- сучасний інтерфейс: інтуїтивно зрозумілий дизайн, який спрощує роботу з системою для користувачів і адміністраторів [8], [9], [14];

- потужний бек-офіс: інструменти для створення списків, пошуку та перегляду деталей елементів, можливість зручно керувати всіма аспектами бібліотеки [8], [9], [14];

- розширені технології каталогізації [8], [9], [14];

- підтримка JSON-схем для організації даних [8], [9], [14];

- гнучка структура для роботи з бібліотечними об'єктами [8], [9], [14];

- розширена функціональність: програмне забезпечення Invenio надає можливість придбання та управління міжбібліотечними позиками через прості модулі, легке керування тиражами та відстеження ресурсів [8], [9], [14];

- автоматичне оновлення даних та оновлення компонентів після внесення змін [8], [9], [14];

- підтримка сучасних технологій: програмне забезпечення Invenio побудовано на базі популярних фреймворків з відкритим кодом, таких як: Python для серверної частини, React для фронтенду [8], [9], [14];

- гнучкість та адаптація: програмне забезпечення Invenio підходить для створення таких застосунків, як: цифрові репозиторії, системи управління даними, механізми пошуку інформації, корпоративні системи виявлення,

висока масштабованість та підтримка налаштувань для користувачів з технічними знаннями [8], [9], [14];

– надійна система автентифікації та зберігання: програмне забезпечення Invenio надає інтегровані модулі для роботи з безпекою даних та управління доступом, а також забезпечує надійне зберігання даних у відповідності до сучасних стандартів [8], [9], [14];

– інтеграція та підтримка API: підтримка REST API для зв'язку з іншими системами [8], [9], [14].

Серед недоліків програмного забезпечення Invenio можна відзначити складність налаштування: для ефективної роботи потрібні технічні знання в таких технологіях, як Python, JSON та REST API, що може бути викликом для користувачів без технічного досвіду. Іншим недоліком є обмеженість документації, що може ускладнити процес навчання та впровадження. Крім того, для розгортання та підтримки інформаційної системи Invenio потрібні значні серверні ресурси, що може бути проблемою для організацій із обмеженим бюджетом [8], [9], [14].

Програмне забезпечення OPALS (рис. 1.2) представляє собою застосунок для автоматизованого управління бібліотекою з відкритим вихідним кодом, що підтримує такі функції, як каталогізація, управління електронними книгами, тираж, створення звітів, керування цифровими архівами та підручниками. Платформа дозволяє створювати великий репозиторій електронних книг і включає понад дві тисячі ресурсів у відкритих, академічних, шкільних та професійних доменах. Програмне забезпечення OPALS забезпечує розширене індексування для швидкого пошуку, управління підписками та запасами, а також підтримку цифрових архівів. Хмарна структура програмного забезпечення OPALS дозволяє отримувати звіти в реальному часі, а автоматизовані оновлення зменшують витрати на технічне обслуговування [8], [9], [15]. Ключові особливості включають зручний вебінтерфейс, підтримку мобільних пристроїв, доступний

Інтернет каталог, професійний розвиток і підтримку, а також економічність [8], [9], [15].

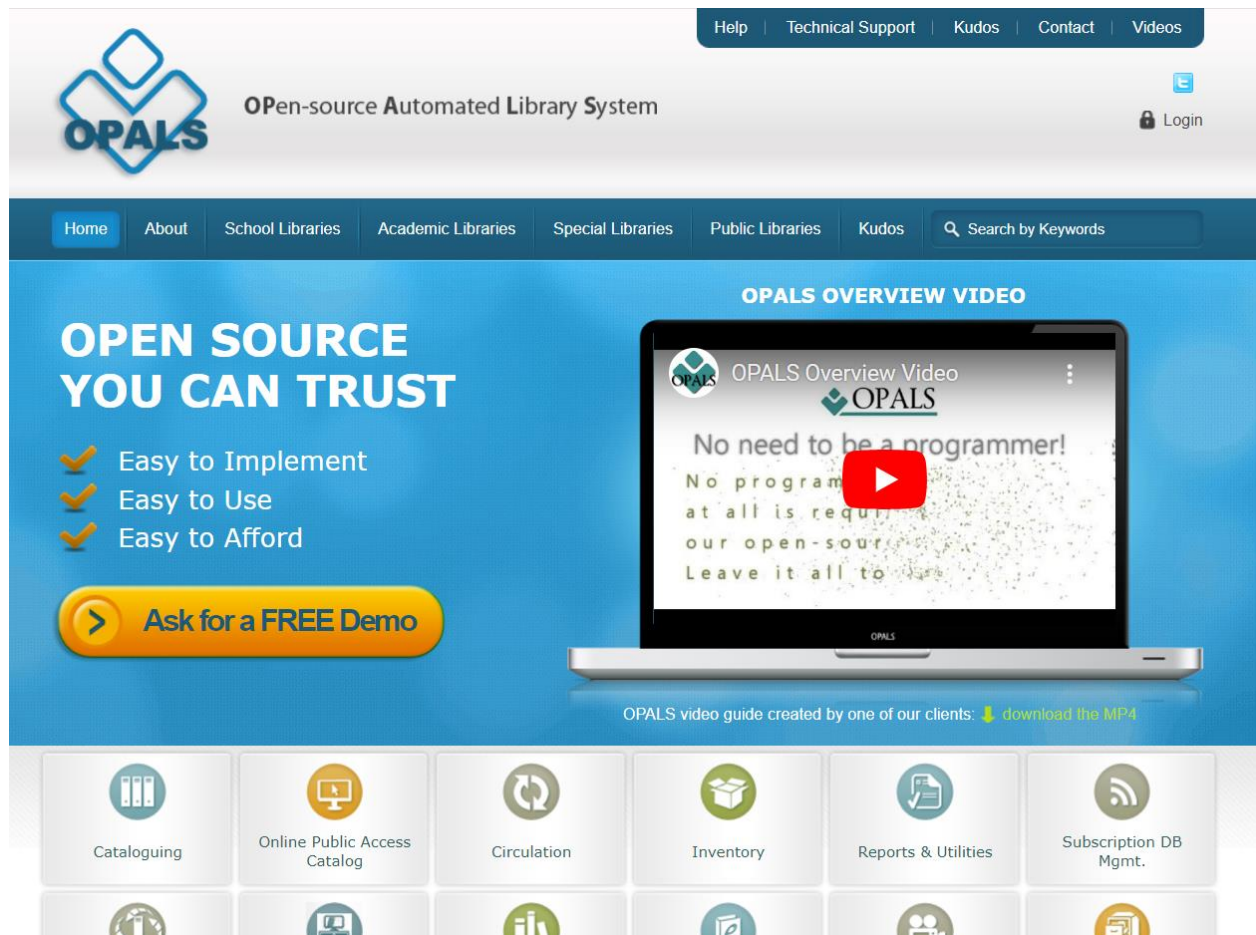


Рисунок 1.2 – Програмне забезпечення OPALS [15]

Інформаційна система OPALS пропонує технічну підтримку для встановлення та обслуговування, а також безкоштовний трьох місячний демонстраційний доступ для ознайомлення з її функціоналом. OPALS являє собою надійне рішення для цифрових бібліотек, яке легко впроваджується та забезпечує широкий спектр корисних функцій [8], [9], [15].

Програмне забезпечення OPALS має такі особливості [8], [9], [15]:

– функціонал управління бібліотекою: програмне забезпечення OPALS надає засоби для каталогізації бібліотечних матеріалів, управління електронними книгами, тиражем, цифровими архівами, підручниками, а також

засоби для розширеного індексування для полегшення пошуку бібліотечних ресурсів [8], [9], [15];

- інтерфейс та доступність: програмне забезпечення OPALS надає зручний вебінтерфейс з підтримкою мобільних пристроїв, Інтернет каталог у відкритому доступі для користувачів [8], [9], [15];

- хмарне рішення: програмне забезпечення OPALS реалізовано як хмарне програмне забезпечення, що дозволяє отримувати звіти в реальному часі. Автоматизовані оновлення серверів зменшують потребу в додаткових витратах на обслуговування та обладнання [8], [9], [15];

- можливості інтеграції: гнучкість у налаштуванні під потреби бібліотеки. Підтримка роботи з різними бібліотечними доменами, включаючи шкільні, академічні та професійні [8], [9], [15];

- підтримка та розвиток: інформаційна система OPALS забезпечує професійний розвиток, технічну підтримку та управління. Пропонує три місячний безкоштовний демонстраційний доступ для ознайомлення з можливостями системи [8], [9], [15];

- економічність: мінімальні витрати на придбання, налаштування та обслуговування [8], [9], [15].

Серед недоліків програмного забезпечення OPALS можна відзначити залежність від хмарного середовища, для роботи потрібен постійний доступ до Інтернету, що може бути проблемою для бібліотек із нестабільним з'єднанням. Іншим недоліком є те, що хоча система OPALS має відкритий код, для повного функціоналу потрібна платна підписка, яка включає технічну підтримку та хостинг. Крім того, може виникати складність у налаштуванні інтеграції програмного забезпечення OPALS з деякими нестандартними або старими бібліотечними системами [8], [9], [15].

Програмне забезпечення Libero (рис. 1.3) являє собою хмарну платформу для управління бібліотекою, яка відома своєю гнучкістю та можливістю налаштувань. Програмне забезпечення Libero підтримує різноманітні типи бібліотек, включаючи публічні, академічні та корпоративні,

а також спеціалізовані установи, такі як музеї, галереї та бібліотеки культурної спадщини [10]-[13], [16].

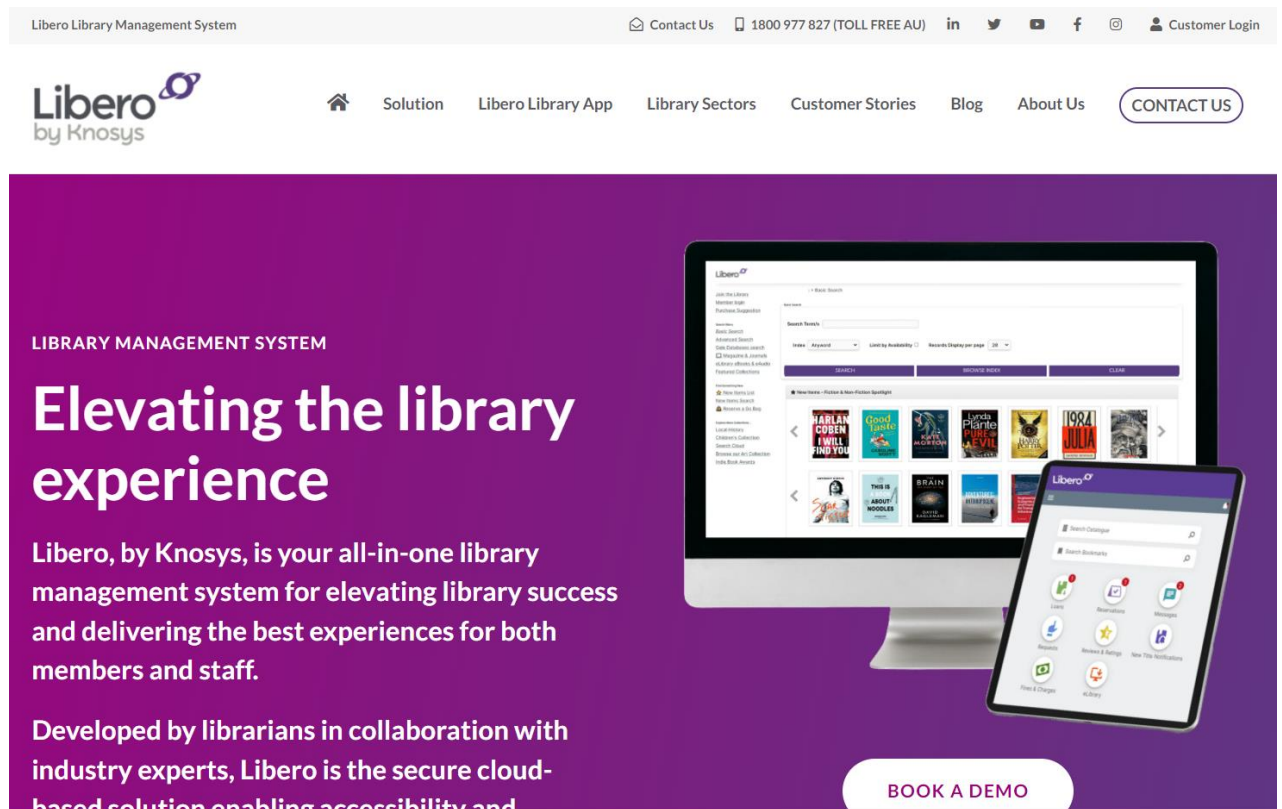


Рисунок 1.3 – Програмне забезпечення Libero [16]

Libero є ефективним інструментом на ринку завдяки відмінній підтримці клієнтів і постійному вдосконаленню. Хмарний формат дозволяє користувачам миттєво отримувати доступ до нових технологій і оновлень. Також є мобільна версія програмного забезпечення Libero, яка виконує функції мобільного порталу та може використовуватися для сканування штрих-кодів і керування процесами [10]-[13], [16].

Libero інтегрується з іншими системами через API, пропонує розширені інструменти звітності та миттєвий доступ до оновлень програмного забезпечення. Інтуїтивно зрозумілий інтерфейс забезпечує зручність використання з самого початку [10]-[13], [16].

Libero також допомагає бібліотекам обробляти придбання та інші елементи, відстежуючи використання і переміщення нових надходжень.

Система підтримує управління серійними публікаціями, включаючи створення списків, обробку заявок та керування списками маршрутизації [10]-[13], [16].

Портал для учасників Libero сприяє взаємодії та залученню користувачів, а його функції звітності дозволяють аналізувати тренди, моделі використання та інші важливі показники для покращення роботи бібліотеки [10]-[13], [16].

Програмне забезпечення Libero має такі особливості [10]-[13], [16]:

- вебдоступність: програмне забезпечення Libero є повністю доступним через веббраузер, що дозволяє користувачам з будь-якого місця отримувати доступ до її функцій [10]-[13], [16];

- підтримка різних типів бібліотек: програмне забезпечення Libero підходить для академічних, публічних, корпоративних та інших бібліотек, таких як музеї, галереї та бібліотеки культурної спадщини [10]-[13], [16];

- мобільний застосунок Libero GO: програмне забезпечення Libero дає можливість управляти бібліотечними процесами та сканувати штрих-коди через мобільні пристрої [10]-[13], [16];

- інтеграція через API: програмне забезпечення Libero підтримує інтеграцію з іншими сторонніми рішеннями через API, що дозволяє зв'язувати його з іншими системами [10]-[13], [16];

- підтримка звітності: платформа надає розширені інструменти звітності, що дозволяє бібліотекам аналізувати використання ресурсів, тренди та інші ключові показники [10]-[13], [16];

- миттєві оновлення: завдяки хмарному розгортанню, користувачі отримують доступ до нових функцій і оновлень програмного забезпечення без затримок [10]-[13], [16];

- оповіщення про завдання: програмне забезпечення Libero підтримує сповіщення про виконання завдань, що дозволяє слідкувати за прогресом у роботі [10]-[13], [16];

– інтуїтивно зрозумілий інтерфейс: легкість у використанні завдяки простому і візуально зрозумілому інтерфейсу, що забезпечує швидке освоєння системи [10]-[13], [16].

Серед недоліків програмного забезпечення Libero можна відзначити те, що для деяких користувачів Libero може бути дорогим, особливо для малих бібліотек або установ з обмеженим бюджетом. Ліцензії та плани розгортання можуть бути дорогими в порівнянні з іншими альтернативам. Хоча інтерфейс є інтуїтивно зрозумілим, деяким користувачам може знадобитися додатковий час для освоєння всіх функцій платформи, особливо в разі налаштувань для спеціалізованих бібліотек або інтеграції з іншими системами [10]-[13], [16].

Порівняльну характеристику інформаційних систем електронних бібліотек наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння інформаційних систем електронних бібліотек

Критерій порівняння	Invenio [14]	OPALS [15]	Libero [16]
Відкритий вихідний код та ліцензія	+	+—	—
Простота інтеграції	+—	+	+
Можливість придбання та управління міжбібліотечними позиками	+	—	+—
Миттєві оновлення	—	—	+
Розширені технології каталогізації	+	+—	+
Хмарна структура	—	+	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато інформаційних систем електронних бібліотек. Проте деякі інформаційні системи електронних бібліотек є досить дорогими

для придбання, впровадження та підтримки, що може бути значною проблемою, особливо для невеликих бібліотек. Інші інформаційні системи електронних бібліотек є низько адаптованими для деяких груп користувачів, вони можуть бути недостатньо інтуїтивними або адаптованими для користувачів із обмеженими можливостями, що ускладнює їх використання певним категоріям людей. Деякі інформаційні системи електронних бібліотек можуть бути недостатньо гнучкими для масштабування, що створює труднощі в обслуговуванні зростаючої кількості користувачів та інформаційних ресурсів бібліотеки. Тому актуальною є розробка програмного забезпечення інформаційної системи електронної бібліотеки.

При розробці програмного забезпечення інформаційної системи електронної бібліотеки необхідно забезпечити такі функціональні вимоги:

- підтримка можливості реєстрації та авторизації читачів електронної бібліотеки;
- підтримка можливості ознайомлення з новими знахідками книжок до електронної бібліотеки;
- можливість перегляду списку книжок читача;
- можливість додавання книжок до списку користувача;
- підтримка можливості читання обраної книжки;
- підтримка можливості фільтрації книжок за жанрами.

1.3 Висновки за розділом 1

Визначено, що електронна бібліотека являє собою зібрання матеріалів у електронному форматі, доступ до яких здійснюється через Інтернет. На відміну від традиційних бібліотек, їй не потрібне фізичне місце для зберігання книг чи інших ресурсів. Усі матеріали в електронних бібліотеках зберігаються за допомогою відповідних інформаційних систем на серверах, що дозволяє користувачам отримувати онлайн доступ до матеріалів бібліотеки з будь-якої локації у світі. Однією з ключових переваг електронних бібліотек є їхня

доступність. Відзначено, що інформаційні системи електронних бібліотек дозволяють отримати доступ до матеріалів з будь-якого куточка світу через Інтернет, що особливо зручно для тих, хто проживає у віддалених районах або не має змоги відвідувати фізичну бібліотеку. Показано, що інформаційні системи електронних бібліотек охоплюють різні аспекти, починаючи з отримання контенту і закінчуючи його організацією, зберіганням, доступом і розповсюдженням. Проаналізовано переваги використання інформаційних технологій електронних бібліотек. Відзначено, що сучасні електронні бібліотеки активно впроваджують інформаційні технології аналізу даних, штучний інтелект та алгоритми машинного навчання, що дозволяє покращити ефективність роботи бібліотек і створити нові можливості для користувачів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато інформаційних систем електронних бібліотек. Проте деякі інформаційні системи електронних бібліотек є досить дорогими для придбання, впровадження та підтримки, що може бути значною проблемою, особливо для невеликих бібліотек. Інші інформаційні системи електронних бібліотек є низько адаптованими для деяких груп користувачів, вони можуть бути недостатньо інтуїтивними або адаптованими для користувачів із обмеженими можливостями, що ускладнює їх використання певним категоріям людей. Деякі інформаційні системи електронних бібліотек можуть бути недостатньо гнучкими для масштабування, що створює труднощі в обслуговуванні зростаючої кількості користувачів та інформаційних ресурсів бібліотеки. Тому актуальною є розробка програмного забезпечення інформаційної системи електронної бібліотеки. Тому актуальною є розробка програмного забезпечення інформаційної системи електронної бібліотеки.

Сформульовано функціональні вимоги до програмного забезпечення інформаційної системи електронної бібліотеки.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення інформаційної системи електронної бібліотеки було використано мову програмування JavaScript [17], [18].

JavaScript є однією з найзручніших і найгнучкіших мов програмування, яка широко використовується для розширення функціональності вебсайтів. З її допомогою можна виконувати різноманітні завдання, такі як створення скріншотів, обробка вебданих, виконання розрахунків і інтеграція важливих функцій через скрипти сторонніх розробників [17], [18].

Сучасне застосування JavaScript виходить за межі браузерів – вона також використовується на серверах та в інших середовищах, де функціонує спеціальний механізм, відомий як JavaScript-двигжок. У браузерах вбудовані такі механізми, які часто називають движками JavaScript [17], [18].

Основою для JavaScript є стандарт ECMAScript. Всі сучасні браузери підтримують версію ECMAScript. JavaScript є однією з найпоширеніших мов програмування, що має низку переваг [17], [18]:

- інтерпретована мова: JavaScript виконується без потреби у попередній компіляції, хоча використання JIT-компіляторів значно підвищує швидкість виконання коду [17], [18];

- простота синтаксису: читабельний і зрозумілий синтаксис дозволяє швидко опанувати мову навіть новачкам, що сприяє створенню простих і ефективних вебсайтів [17], [18];

- зручне налагодження: JavaScript дозволяє виявляти помилки в коді безпосередньо під час виконання, показуючи номер рядка з помилкою, що спрощує процес виправлення та тестування [17], [18];

– програмування, кероване подіями: мова дозволяє реагувати на події, такі як натискання кнопки чи зміна значень, забезпечуючи гнучкість і динамічність [17], [18];

– висока продуктивність: завдяки можливостям відстеження та оптимізації, JavaScript забезпечує ефективну роботу навіть за високих навантажень [17], [18];

– безкоштовне використання: мова є відкритою та підтримується усіма браузерами. Код JavaScript легко адаптується до різного обладнання, що робить її універсальним вибором для розробників [17], [18];

Таким чином, JavaScript є універсальним і потужним інструментом для розробки як простих, так і складних вебрішень [17], [18].

JavaScript є багатоплатформною мовою програмування, яка дозволяє користувачам взаємодіяти з різними елементами вебсайту, такими як заголовки, підвали, спливаючі вікна та інші інтерактивні компоненти. Крім роботи у веббраузерах, JavaScript також застосовується на стороні сервера за допомогою таких фреймворків, як Node.js. Це забезпечує більш ефективну роботу, наприклад, виконання додатків на кількох пристроях одночасно. У середовищах, таких як браузери, JavaScript інтегрується з оточуючими об'єктами, надаючи розробникам інструменти для їх обробки [17], [18].

Мова JavaScript включає бібліотеки для роботи з масивами, датами, арифметичними операціями, а також підтримує різноманітні елементи синтаксису, такі як оператори, вирази та послідовності. JavaScript фреймворки пропонують широкий спектр інструментів для розв'язання різноманітних задач, серед яких є як робота на клієнтській стороні, так і на серверній стороні [17], [18].

Робота на клієнтській стороні: JavaScript дозволяє взаємодіяти з об'єктами вебсторінки через модель об'єктів документа. Наприклад, користувачі можуть додавати елементи у форми HTML, а також обробляти події, такі як кліки миші, введення користувачем тексту чи прокручування сторінки [17], [18].

Серверна реалізація: JavaScript на стороні сервера дозволяє працювати з об'єктами для обробки запитів, таких як отримання даних із програм або серверів. Наприклад, серверні розширення можуть обробляти запити для передачі даних між програмами або завантажувати файли безпосередньо із сервера [17], [18].

Таким чином, JavaScript є універсальним інструментом для створення інтерактивних та динамічних програм як на стороні клієнта, так і на стороні сервера [17], [18].

JavaScript і Java суттєво відрізняються, хоча їх синтаксис має схожі елементи. На відміну від Java, JavaScript не вимагає суворої типізації та перевірки типів. Перейменування LiveScript у JavaScript було пов'язане з популярністю Java, але ці мови мають різні підходи до плагінів, правил іменування та стандартів дизайну [17], [18].

JavaScript, на відміну від Java, підтримує типи даних, такі як логічні значення та рядки, але не має жорсткої типізації. Його об'єктна модель заснована на прототипах, що дозволяє динамічне успадкування, на відміну від класово-орієнтованої моделі Java. Це забезпечує більш гнучку організацію коду та дозволяє створювати об'єкти через прототипи, які можуть динамічно змінюватися [17], [18].

JavaScript менш формалізований у порівнянні з Java. Наприклад: не потрібно оголошувати змінні, класи чи методи перед їх використанням, не потрібно задавати доступ до методів (публічний, приватний чи захищений) або реалізовувати інтерфейси.

Типи параметрів, а також типи повернення функцій не обов'язково мають бути вказані. JavaScript також дозволяє функціям бути об'єктами, що відкриває можливості для більш динамічного та модульного підходу до програмування. Це робить мову більш гнучкою та адаптивною для створення інтерактивних вебзастосунків [17], [18].

Обґрунтування вибору мови програмування для розробки програмного забезпечення інформаційної системи електронної бібліотеки наведено у

таблиці 2.1. У таблиці показано, що використання JavaScript забезпечує ефективну, гнучку та зручну розробку програмного забезпечення інформаційної системи електронної бібліотеки з підтримкою сучасних технологій і вимог.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення інформаційної системи електронної бібліотеки

Критерій порівняння мов програмування	Мова програмування		
	JavaScript	Python	C#
Кросплатформність	+	+	+—
Інтерпретованість	+	+	—
Низький поріг входу	+	+—	+—
Зручність для створення ПЗ інформаційної системи електронної бібліотеки	+	—	—
Ефективність для створення як клієнтської, так і серверної частин	+	+—	+

Отже, для реалізації програмного забезпечення інформаційної системи електронної бібліотеки обрано мову програмування JavaScript завдяки її універсальності та потужним можливостям для створення як клієнтської, так і серверної частин програми. Крім того, JavaScript має сильну екосистему інструментів і модулів, що дозволяє швидко інтегрувати необхідний функціонал, наприклад, для роботи з базами даних, авторизації, або обробки запитів API. Також мова має низький поріг входу, що полегшує залучення розробників до проєкту. JavaScript дозволяє створювати додатки, які коректно працюють у всіх сучасних веббраузерах і різних операційних системах.

Завдяки підтримці JavaScript різних інформаційних технологій, забезпечується безперебійна робота програми незалежно від пристрою користувача.

2.2 Вибір бібліотеки для розробки інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки

В якості бібліотеки для розробки інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки було обрано React [19], [20].

Бібліотека для побудови інтерфейсів React є JavaScript-бібліотекою, яка дозволяє створювати багаторазові компоненти для динамічних інтерфейсів. Такі компоненти можуть взаємодіяти між собою, легко обмінюючись даними, і керувати своїми властивостями та станом. Це значно спрощує роботу з даними та їхнім відображенням, особливо в одно сторінкових програмних застосунках [19], [20].

React забезпечує ефективну роботу завдяки використанню віртуального DOM. Замість того, щоб браузер щоразу повністю перезавантажував реальний DOM, React оновлює лише ті його частини, які змінилися. Віртуальний DOM є абстракцією даних, що дозволяє мінімізувати операції рендерингу та забезпечувати високу продуктивність за допомогою оптимізованих анімацій і мінімізації змін у DOM [19], [20].

React також підтримує синтаксис, який спрощує написання коду, роблячи його швидшим і ефективнішим. На відміну від класичної архітектури, ReactJS працює лише з односпрямованим потоком даних. Це означає, що дані передаються від батьківських компонентів до дочірніх, але не навпаки. Такий підхід забезпечує передбачуваність та полегшує створення структурованих наборів даних [19], [20].

Коли дані змінюються, React автоматично повторно відображає лише ті компоненти, які залежать від цих змін, забезпечуючи актуальність відображення. Щоб підтримувати таку модель роботи з даними, React використовує методологію Flux як альтернативу класичній архітектурі моделі-

подання-контролер. Цей підхід спрощує управління станом і дозволяє краще структурувати додатки [19], [20].

Використання React для розробки програмного забезпечення електронної бібліотеки є виправданим завдяки його потужним можливостям для створення сучасних і продуктивних вебзастосунків. React є бібліотекою для розробки користувацького інтерфейсу, яка базується на компонентному підході, що дозволяє створювати багаторазові, добре структуровані та легко підтримувані компоненти [19], [20].

Обґрунтування вибору бібліотеки для розробки інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору бібліотеки для розробки інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки

Критерій порівняння бібліотек для розробки інтерфейсу	Бібліотеки для розробки інтерфейсу		
	React	Angular	Svelte
Легкість інтеграції з іншими бібліотеками та фреймворками	+	+—	—
Широка екосистема інструментів	+	+—	+
Висока продуктивність	+	+—	+—
Використання технології Virtual DOM	+	—	—
Компонентний підхід	+	+	+—

Таким чином, для створення інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки обрано бібліотеку React, яка базується на компонентному підході, що дозволяє створювати багаторазові, добре структуровані та легко підтримувані компоненти. Крім того, React забезпечує високу продуктивність, що особливо важливо для інформаційної системи електронної бібліотеки, де користувачі можуть виконувати пошук книг, переглядати колекції та взаємодіяти з різноманітними даними у режимі реального часу.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення інформаційної системи електронної бібліотеки обрано мову програмування JavaScript завдяки її універсальності та потужним можливостям для створення як клієнтської, так і серверної частин програми. Крім того, JavaScript має сильну екосистему інструментів і модулів, що дозволяє швидко інтегрувати необхідний функціонал, наприклад, для роботи з базами даних, авторизації, або обробки запитів API. Також мова має низький поріг входу, що полегшує залучення розробників до проєкту. JavaScript дозволяє створювати програми, які коректно працюють у всіх сучасних веббраузерах і різних операційних системах. Завдяки підтримці JavaScript різних інформаційних технологій, забезпечується безперебійна робота програми незалежно від пристрою користувача.

Для створення інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки обрано бібліотеку React, яка базується на компонентному підході, що дозволяє створювати багаторазові, добре структуровані та легко підтримувані компоненти. Крім того, React забезпечує високу продуктивність, що особливо важливо для інформаційної системи електронної бібліотеки, де користувачі можуть виконувати пошук книг,

переглядати колекції та взаємодіяти з різноманітними даними у режимі реального часу.

3 ОПИС ПРОГРАМИ

3.1 Структура программного обеспечения информационной системы электронной библиотеки

Структуру программного обеспечения информационной системы электронной библиотеки наведено на рис. 3.1.

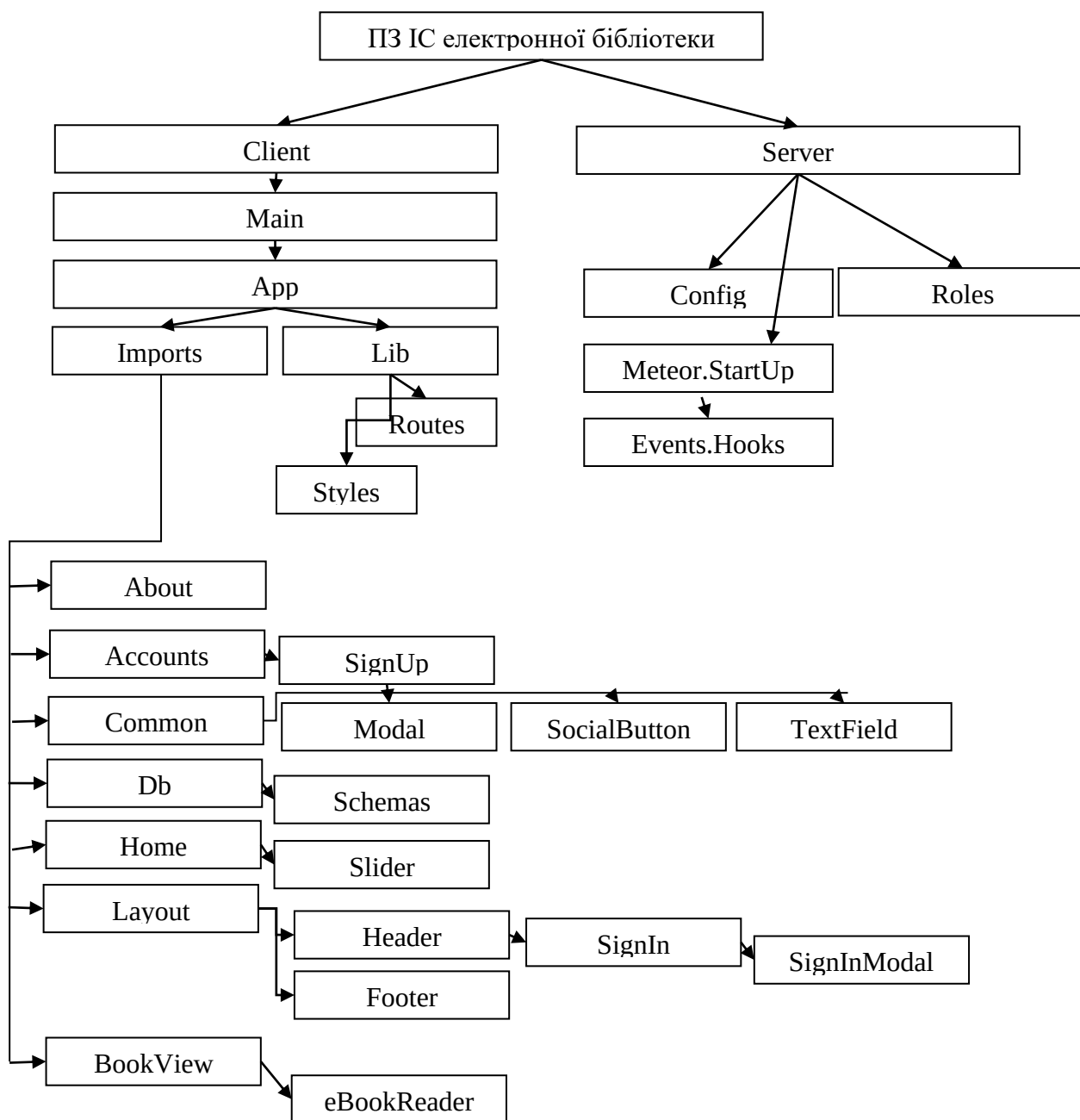


Рисунок 3.1 – Структура програмного забезпечення інформаційної системи електронної бібліотеки

Програмне забезпечення інформаційної системи електронної бібліотеки складається з трьох основних частин: клієнтської частини, серверної частини та бази даних.

Клієнтська частина відповідає за взаємодію користувача із системою. Вона побудована на основі ReactJS, що забезпечує динамічну роботу в реальному часі. Основна структура клієнтської частини включає модулі для запуску програми, маршрутизації, роботи з обліковими записами, реєстрації нових користувачів і функцій авторизації через сторонні сервіси. Інтерфейс реалізовано з використанням візуального конструктора, що дозволяє легко змінювати дизайн і додавати функціональні компоненти. Однією з ключових особливостей є вбудований модуль eBookReader, який дозволяє читати книги у форматі ePub безпосередньо у браузері.

Основним ядром клієнтської частини є клас Main, який служить точкою входу для збірки всіх клієнтських модулів і ініціалізує додаток. Клас App виконує роль центрального компонента, що управляє інтерфейсом користувача, використовуючи React Router для організації маршрутизації між різними сторінками. У свою чергу, клас Lib відповідає за організацію маршрутів та підключення глобальних стилів, забезпечуючи єдність зовнішнього вигляду та структури додатка. Для відображення інформації про бібліотеку передбачений клас About, а логіка роботи з обліковими записами користувачів реалізована через компонент Accounts. Останній підтримує авторизацію, управління профілем та збереження персональних даних. Функція реєстрації нових користувачів реалізована в компоненті SignUp, що забезпечує простий та інтуїтивний процес створення облікового запису.

Особливу увагу приділено повторно використовуваним компонентам, які згруповані в модулі Common. Це дозволяє зменшити дублювання коду та прискорити розробку. Авторизація через сторонні сервіси, інтегрована через компонент SocialButton, що додає зручності користувачам.

Серверна частина забезпечує обробку запитів від клієнтської частини, управління логікою роботи системи та взаємодію з базою даних. Використання MeteorJS дозволяє ефективно реалізувати аутентифікацію, авторизацію, управління даними та синхронізацію у реальному часі. Серверна частина також підтримує інтеграцію із зовнішніми сервісами та реалізує логіку роботи бібліотеки, включаючи управління книгами, обліковими записами користувачів та операціями з ними.

Одним із основних завдань серверної частини є забезпечення ідентифікації та авторизації користувачів. Це включає обробку запитів на реєстрацію нових користувачів, управління входом до системи та контроль доступу до різних ресурсів бібліотеки.

Також серверна частина забезпечує роботу з засобами для інтеграції з зовнішніми сервісами, що дозволяє користувачам здійснювати соціальну авторизацію. Це робить взаємодію з бібліотекою зручнішою та доступнішою. Ще одним важливим аспектом є логіка управління бібліотекою. Сервер обробляє запити на додавання, редагування чи видалення книг, а також забезпечує управління правами доступу до них, що гарантує безпеку та точність роботи системи.

База даних використовується для зберігання всієї інформації системи. Вона містить дані про користувачів, книги, транзакції, підписки та аналітику. Завдяки можливостям системи управління база даних забезпечує швидкий доступ до інформації та підтримує масштабованість системи.

Для інтеграції всіх компонентів використовується платформа MeteorJS, яка забезпечує реактивність даних, синхронізацію між клієнтом і сервером, а також спрощує управління доступом. Така структура дозволяє створити зручну та функціональну систему для роботи з електронною бібліотекою.

3.2 Функціонування програмного забезпечення інформаційної системи електронної бібліотеки

Функціонування програмного забезпечення інформаційної системи електронної бібліотеки подамо за допомогою схеми, зображеної на рис. 3.2.

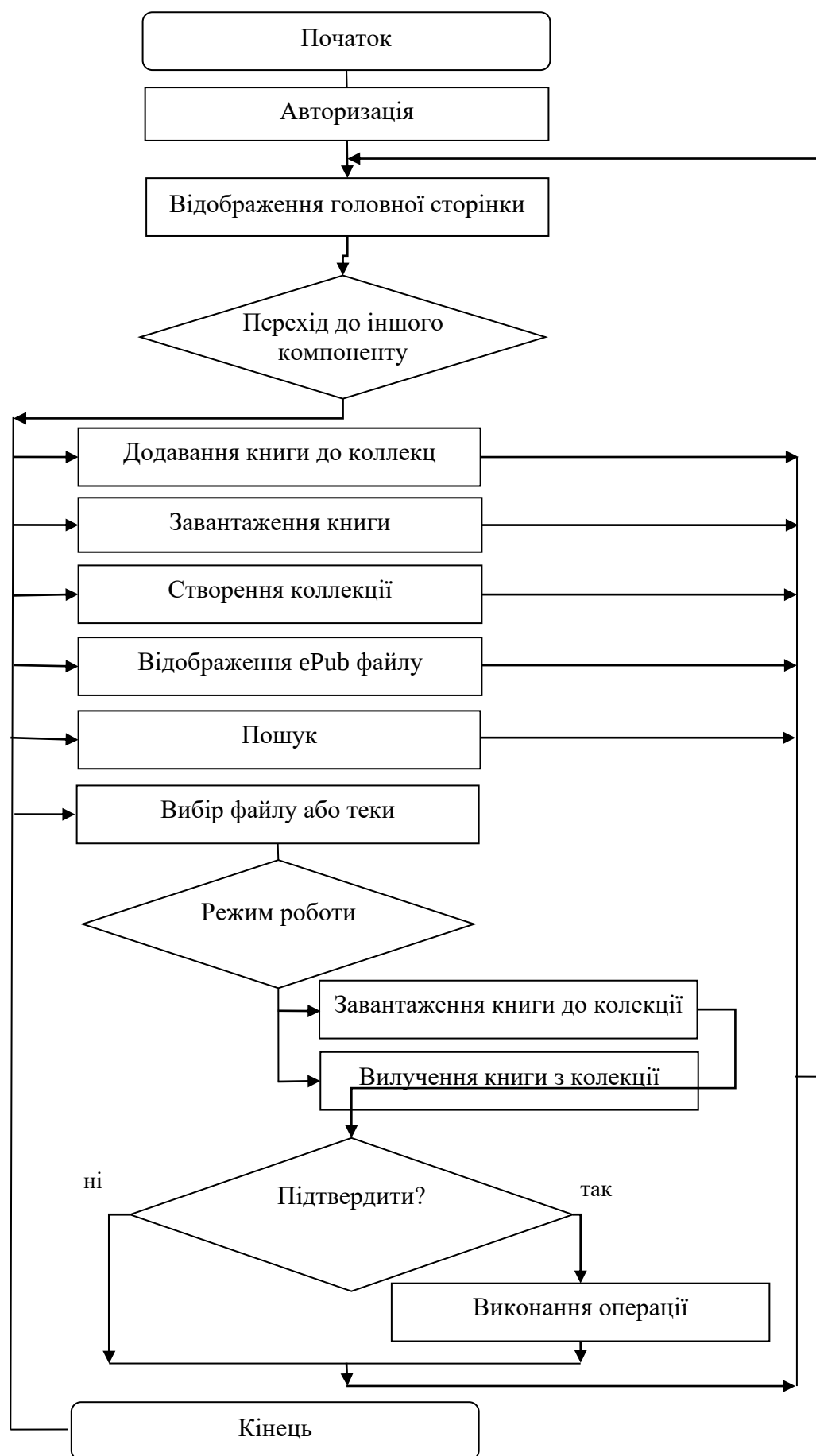


Рисунок 3.2 – Схема функціонування ПЗ ІС електронної бібліотеки

Функціонування програмного забезпечення інформаційної системи електронної бібліотеки побудовано так, щоб забезпечити як базову, так і розширену взаємодію користувача з бібліотекою. Початковим етапом роботи з системою є процес авторизації. У разі успішної авторизації користувач програмного забезпечення інформаційної системи електронної бібліотеки отримує доступ до розширених функцій, пов'язаних із персональним обліковим записом і обробкою персональних даних. У випадку невдалої авторизації доступ до базового функціоналу програмного забезпечення інформаційної системи електронної бібліотеки залишається, але можливості, пов'язані з управлінням обліковим записом, зникають.

Після авторизації програмного забезпечення інформаційної системи електронної бібліотеки користувачу стають доступними функції редагування профілю, створення та управління власною колекцією книг, а також можливість зберігати прогрес читання для подальшого продовження. Програмне забезпечення інформаційної системи електронної бібліотеки також надає пошуковий інструмент для швидкого доступу до потрібних матеріалів електронних книг.

Додатково авторизовані користувачі програмного забезпечення інформаційної системи електронної бібліотеки мають можливість завантажувати власні книги у відповідному форматі і редагувати раніше додані матеріали. Для завершення операцій модифікації необхідно підтвердити внесені зміни, після чого нові чи оновлені книги стають доступними для загального користування в бібліотеці.

Таким чином, програмне забезпечення інформаційної системи електронної бібліотеки забезпечує динамічну та зручну взаємодію, дозволяючи користувачам максимально ефективно використовувати ресурси бібліотеки залежно від рівня доступу.

Функціонування програмного забезпечення інформаційної системи електронної бібліотеки, що подано за допомогою діаграми прецедентів, наведено на рис. 3.3.

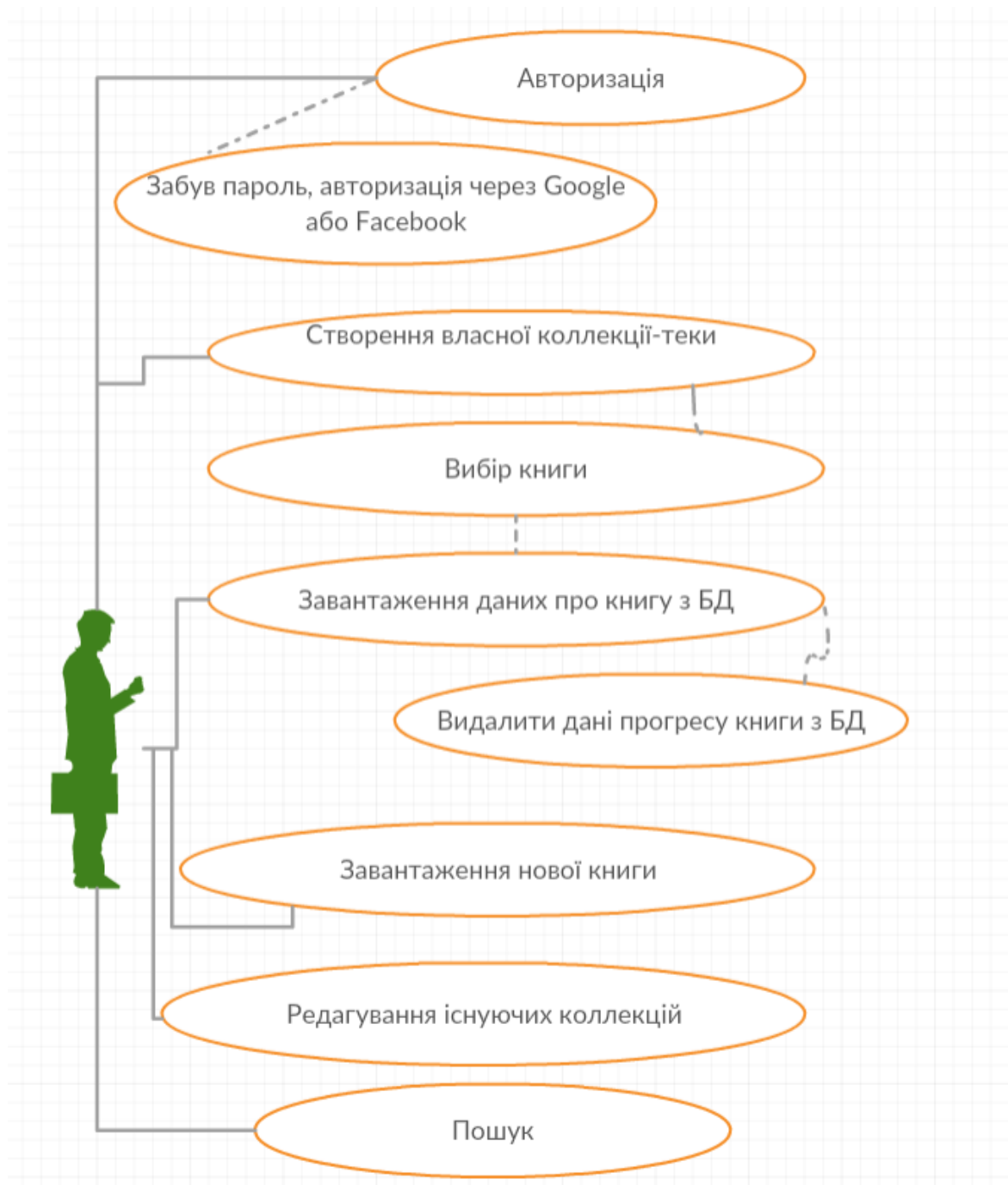


Рисунок 3.3 – Діаграма прецедентів ПЗ ІС електронної бібліотеки

3.3 Розробка бази даних програмного забезпечення інформаційної системи електронної бібліотеки

База даних програмного забезпечення інформаційної системи електронної бібліотеки призначена для зберігання даних про користувачів, книги, транзакції, підписки та аналітику. Завдяки можливостям системи

управління база даних забезпечує швидкий доступ до інформації та підтримує масштабованість системи [21].

Схему бази даних програмного забезпечення інформаційної системи електронної бібліотеки наведено на рис. 3.4.

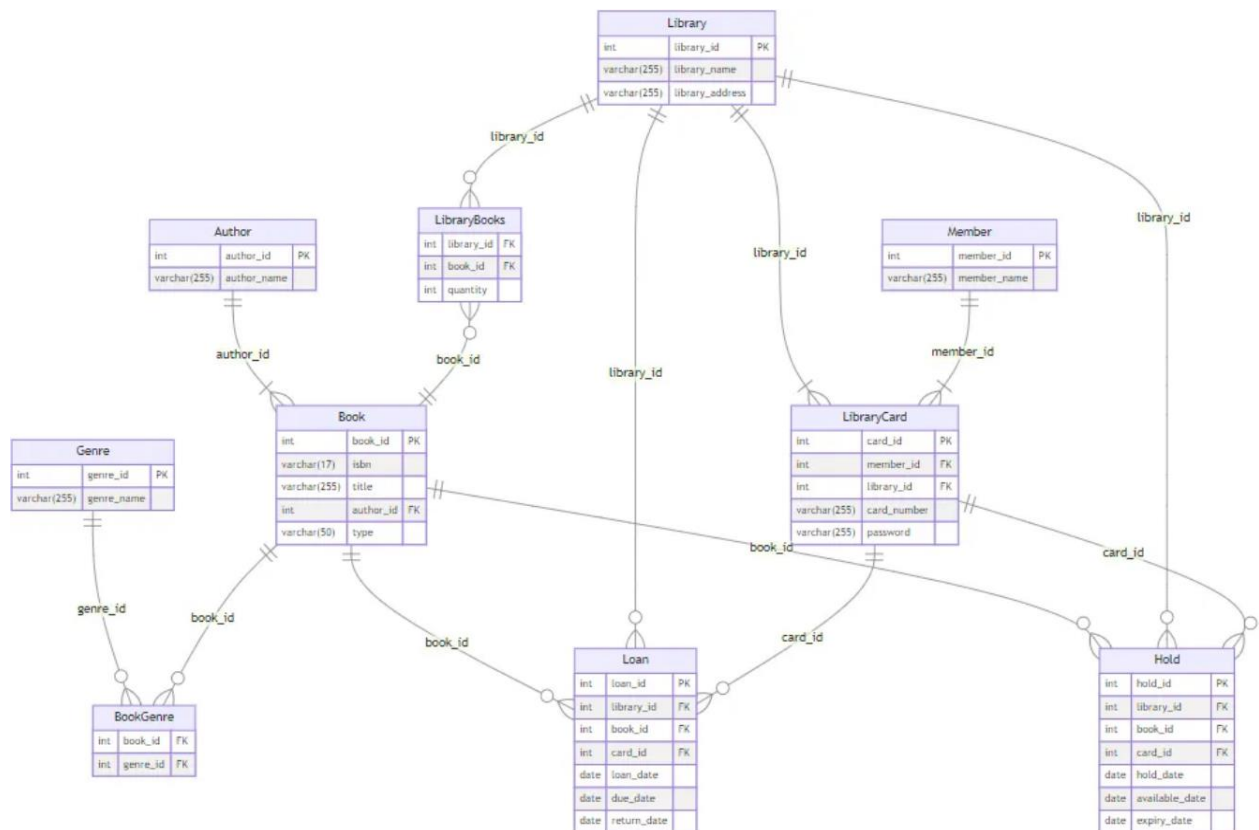


Рисунок 3.4 – Схему бази даних програмного забезпечення інформаційної системи електронної бібліотеки

Згідно з поставленими вимогами, програмне забезпечення інформаційної системи підтримує керування даними для електронних бібліотек і забезпечувати зв'язок читачів з відповідними бібліотеками. Кожна бібліотека може видавати свої власні книги, які користувачі можуть додати до власного каталогу. Це дозволяє здійснювати пошук та перевірку доступності певної книги у всіх бібліотеках.

Основні об'єкти бази даних та їх атрибути включають: Library (містить поля library_id, library_name, library_address), Book (включає book_id, title, isbn, type), Author (представлено атрибутами author_id, author_name), Genre

(включає genre_id, genre_name), а також об'єкти Member, Card, Loan, Hold. На основі цих об'єктів можна створити таблиці для бази даних.

Опишемо деякі зв'язки між таблицями. Один автор може написати багато книг, але кожна книга має лише одного автора, утворюючи зв'язок «один-до-багатьох». Книга може належати до кількох жанрів, і кожен жанр може включати багато книг, що також створює зв'язок «багато-до-багатьох». Одна книга може бути доступною в багатьох бібліотеках, і кожна бібліотека може мати багато книг, що формує зв'язок «багато-до-багатьох». Один читач може мати кілька бібліотечних карток, але кожна картка прив'язана лише до одного члена, утворюючи зв'язок «один-до-багатьох». Бібліотека може видавати багато карток, але кожна картка належить лише одній бібліотеці, що формує зв'язок «один-до-багатьох». Книга з бібліотеки може бути пов'язана лише з однією позикою, формуючи зв'язок «один-до-одного». Книга з бібліотеки може мати багато бронювань, але кожне бронювання належить лише до однієї книги, утворюючи зв'язок «один-до-багатьох».

Схема бази даних програмного забезпечення інформаційної системи електронної бібліотеки відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Основними сутностями бази даних обрано такі: книга, автор, читач (Member), жанр книги, бібліотечна картка читача, позичка книги (loan) та інші.

3.4 Проєктування інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням інформаційної системи електронної бібліотеки виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення інформаційної системи електронної бібліотеки враховані функціональні вимоги до програми:

- підтримка можливості реєстрації та авторизації читачів електронної бібліотеки;
- підтримка можливості ознайомлення з новими находженнями книжок до електронної бібліотеки;
- можливість перегляду списку книжок читача;
- можливість додавання книжок до списку користувача;
- підтримка можливості читання обраної книжки;
- підтримка можливості фільтрації книжок за жанрами.

Інтерфейс головної форми програмного забезпечення інформаційної системи електронної бібліотеки наведено на рис. 3.5.

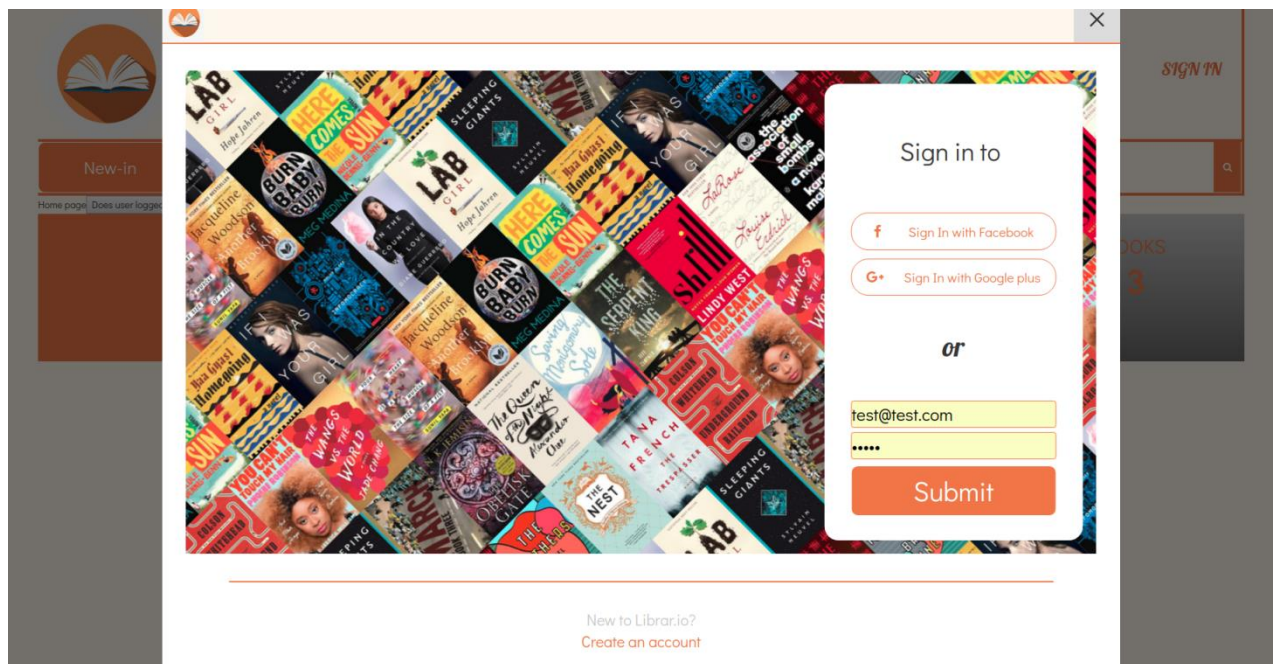


Рисунок 3.5 – Інтерфейс головної форми програмного забезпечення інформаційної системи електронної бібліотеки

На рис. 3.6 наведено форму реєстрації в електронній бібліотеці.

New?

Sign up to upload new books and make bookmarks!

Email

Your name

Password

Confirm Password

Submit

Рисунок 3.6 – Реєстрація в електронній бібліотеці

На рисунку 3.7 наведено область сторінки з відображенням жанрів книжок електронної бібліотеки.



Рисунок 3.7 – Область сторінки з відображенням жанрів книжок електронної бібліотеки

На рис. 3.8 наведено область сторінки з відображенням книжок з колекції користувача.

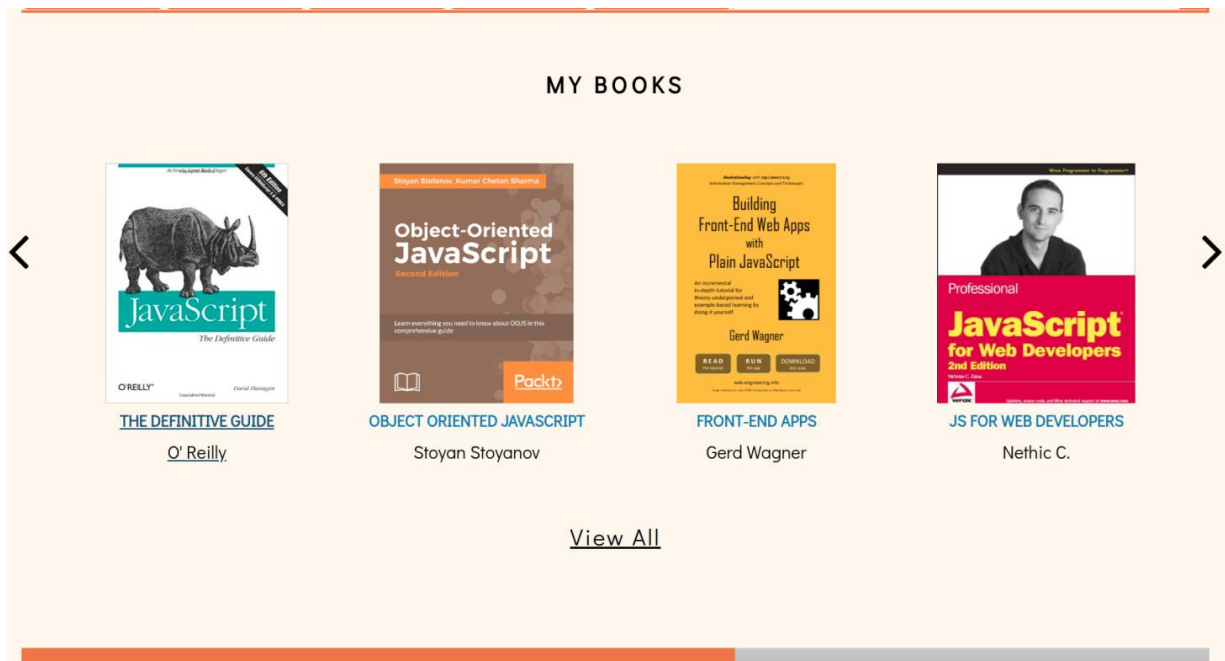


Рисунок 3.8 – Книжки з колекції користувача

Сторінку електронної бібліотеки у режимі читання книжки наведено на рис. 3.9.



Рисунок 3.9 – Режим читання обраної книжки

На рис. 3.10 наведено сторінку додавання нової книги до колекції.

The screenshot shows a web application interface for adding a new book. At the top, there is a navigation bar with several tabs: 'New-in', 'Browse by Subject', 'My Books', 'More -', 'Add a book', and a search bar. The main content area is a light orange box containing a 'Name' input field, a book icon, and a 'Submit' button. The footer is divided into two sections: an orange section on the left with 'About' text and a grey section on the right with 'Get started' and 'BOOKS 3'.

Рисунок 3.10 – Сторінка додавання нової книги до колекції книжок

3.5 Висновки за розділом 3

Розроблено програмне забезпечення інформаційної системи електронної бібліотеки.

Запропоновано структуру програмного забезпечення інформаційної системи електронної бібліотеки, що складається з клієнтської частини, серверної частини та бази даних. Клієнтська частина відповідає за взаємодію користувача із системою. Вона побудована на основі ReactJS, що забезпечує динамічну роботу в реальному часі. Основна структура клієнтської частини включає модулі для запуску програми, маршрутизації, роботи з обліковими записами, реєстрації нових користувачів і функцій авторизації через сторонні сервіси. Інтерфейс реалізовано з використанням візуального конструктора, що дозволяє легко змінювати дизайн і додавати функціональні компоненти. Серверна частина забезпечує обробку запитів від клієнтської частини, управління логікою роботи системи та взаємодію з базою даних. Серверна частина також підтримує інтеграцію із зовнішніми сервісами та реалізує логіку роботи бібліотеки, включаючи управління книгами, обліковими записами користувачів та операціями з ними.

Розроблено базу даних програмного забезпечення інформаційної системи електронної бібліотеки, яка використовується для зберігання даних про користувачів, книги, транзакції, підписки та аналітику. Основними сутностями бази даних обрано такі: книга, автор, читач (Member), жанр книги, бібліотечна картка читача, позичка книги (loan) та інші.

Описано функціонування програмного забезпечення інформаційної системи електронної бібліотеки.

Описано особливості реалізації програмного забезпечення інформаційної системи електронної бібліотеки. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням інформаційної системи електронної бібліотеки.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу функціонування електронної бібліотеки. Програма написана на мові JavaScript, яка має сильну екосистему інструментів і модулів, що дозволяє швидко інтегрувати необхідний функціонал, наприклад, для роботи з базами даних, авторизації, або обробки запитів API. Для розробки інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки обрано бібліотеку React, яка базується на компонентному підході, що дозволяє створювати багаторазові, добре структуровані та легко підтримувані компоненти.

Програмне забезпечення інформаційної системи електронної бібліотеки забезпечує виконання таких функцій:

- підтримка можливості реєстрації та авторизації читачів електронної бібліотеки;
- підтримка можливості ознайомлення з новими знахідками книжок до електронної бібліотеки;
- можливість перегляду списку книжок читача;
- можливість додавання книжок до списку користувача;
- підтримка можливості читання обраної книжки;
- підтримка можливості фільтрації книжок за жанрами.

Для роботи програмного забезпечення інформаційної системи електронної бібліотеки потрібні такі технічні та програмні ресурси:

- оперативна пам'ять об'ємом 500 Мб і більше;
- щонайменше 600 МБ вільного місця на пристрої зберігання даних;
- процесор із тактовою частотою не нижче 2 ГГц;
- встановлений вебсервер для забезпечення роботи системи;
- операційна система та веббраузер на клієнтському пристрої.

Функціональні характеристики розробленого програмного забезпечення інформаційної системи електронної бібліотеки наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення інформаційної системи електронної бібліотеки наведено у додатку Б.

4.2 Характеристики програмного забезпечення інформаційної системи електронної бібліотеки

Програмне забезпечення інформаційної системи електронної бібліотеки складається з клієнтської частини, серверної частини та бази даних. Клієнтська частина відповідає за взаємодію користувача із системою. Основна структура клієнтської частини включає модулі для запуску програми, маршрутизації, роботи з обліковими записами, реєстрації нових користувачів і функцій авторизації через сторонні сервіси. Інтерфейс реалізовано з використанням візуального конструктора, що дозволяє легко змінювати дизайн і додавати функціональні компоненти. Серверна частина забезпечує обробку запитів від клієнтської частини, управління логікою роботи системи та взаємодію з базою даних. Серверна частина також підтримує інтеграцію із зовнішніми сервісами та реалізує логіку роботи бібліотеки, включаючи управління книгами, обліковими записами користувачів та операціями з ними. База даних програмного забезпечення інформаційної системи електронної бібліотеки використовується для зберігання даних про користувачів, книги, транзакції, підписки та аналітику.

Програмне забезпечення інформаційної системи електронної бібліотеки забезпечує авторизацію користувачів для доступу до персоналізованих функцій, таких як редагування профілю, створення колекцій книг, збереження прогресу читання та пошук літератури через інтегрований пошуковий інструмент. Платформа дозволяє завантажувати власні книги і редагувати наявні файли, надаючи інструменти для підтвердження операцій та їх публікації.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення інформаційної системи електронної бібліотеки на серверній частини необхідно запустити відповідний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення інформаційної системи електронної бібліотеки є інформація про книги, користувачів, замовлення книжок та ін.

Вихідними даними програмного забезпечення інформаційної системи електронної бібліотеки є створені у базі даних записи про книги, користувачів, замовлення книжок та ін., а також вебсторінки електронної бібліотеки, що відображаються користувачу у відповідь на його запити.

4.3.3 Повідомлення

Користувач програмного забезпечення інформаційної системи електронної бібліотеки може отримати такі повідомлення: повідомлення про успішне оформлення замовлення книжки та ін.

4.4 Виконання програмного забезпечення інформаційної системи електронної бібліотеки

Після запуску програмного забезпечення інформаційної системи електронної бібліотеки відображається головна сторінка (рис. 4.1). Користувач може увійти до свого аккаунту або створити новий за допомогою відповідного посилання.

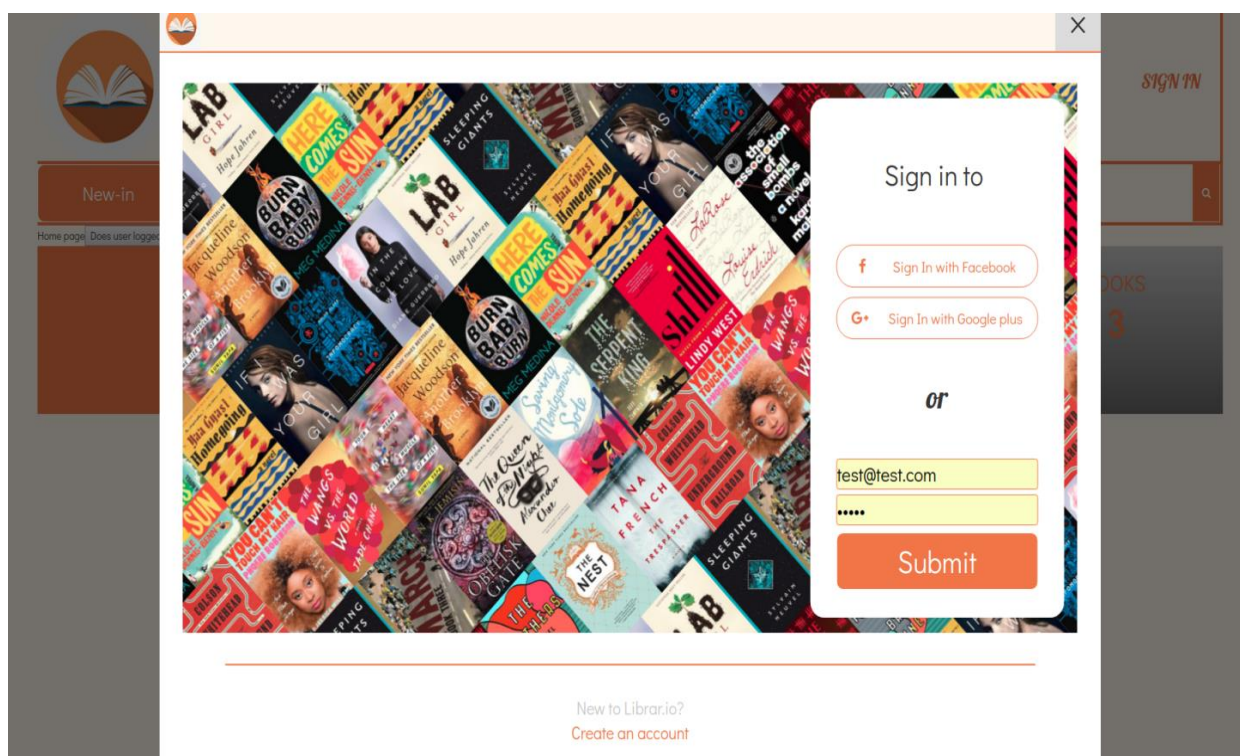


Рисунок 4.1 – Інтерфейс головної сторінки електронної бібліотеки

На рис. 4.2 наведено головне меню вебсайту електронної бібліотеки. Користувач має можливості ознайомлення з новими надходженнями книжок до бібліотеки, переглянути список своїх книжок, а також додати книгу до своєї колекції.

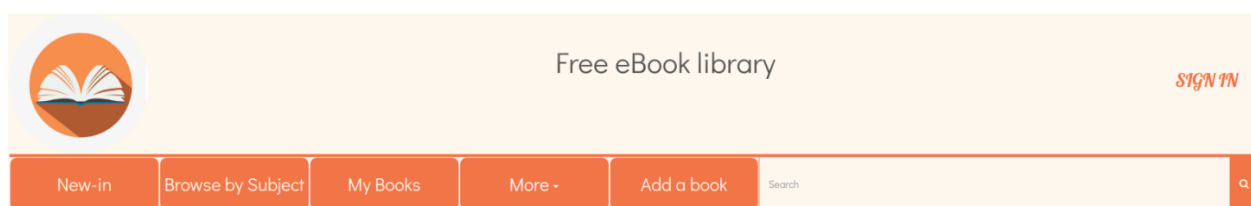


Рисунок 4.2 – Головне меню вебсайту електронної бібліотеки

Незареєстрований користувач електронної бібліотеки має можливості реєстрації (рис. 4.3). Для цього необхідно ввести свою електронну адресу, ім'я та пароль.

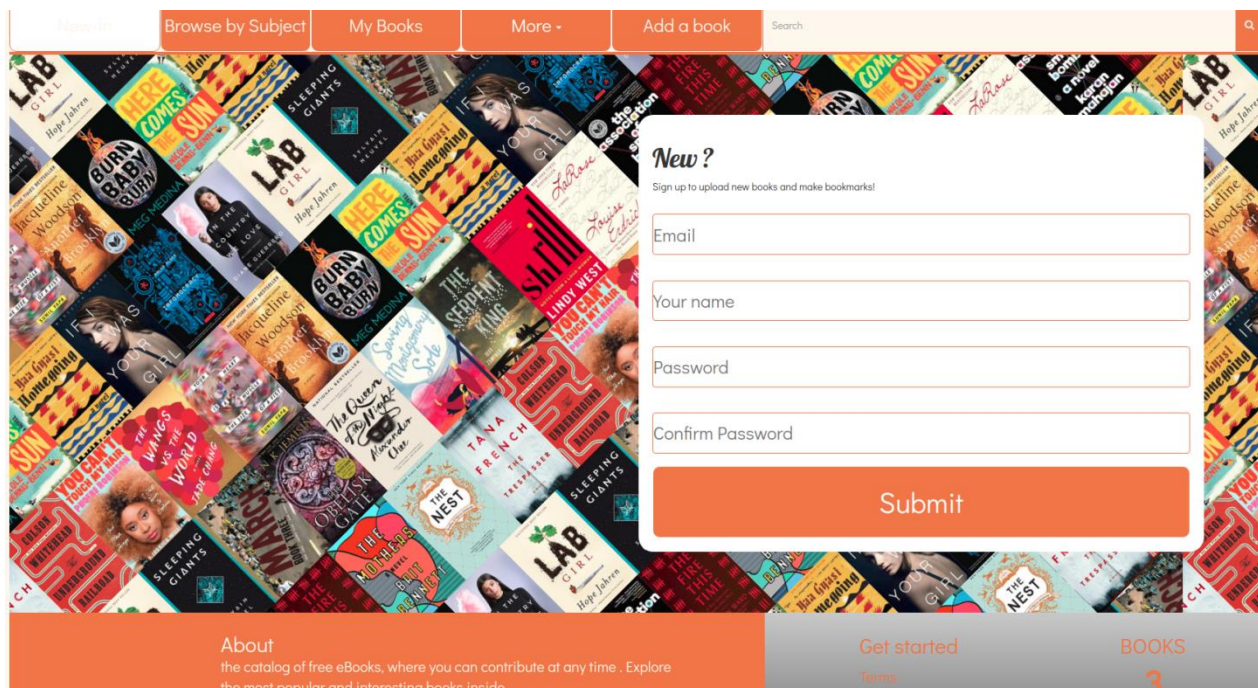


Рисунок 4.3 – Реєстрація в електронній бібліотеці

Для відображення книжок по жанру необхідно обрати пункт меню Browse by Subject (рис. 4.4), після чого потрібно обрати відповідний жанр.



Рисунок 4.4 – Відображення жанрів книжок

При виборі пункту меню My books буде відображено перелік книжок, що користувач додав до своєї колекції (рис. 4.5).

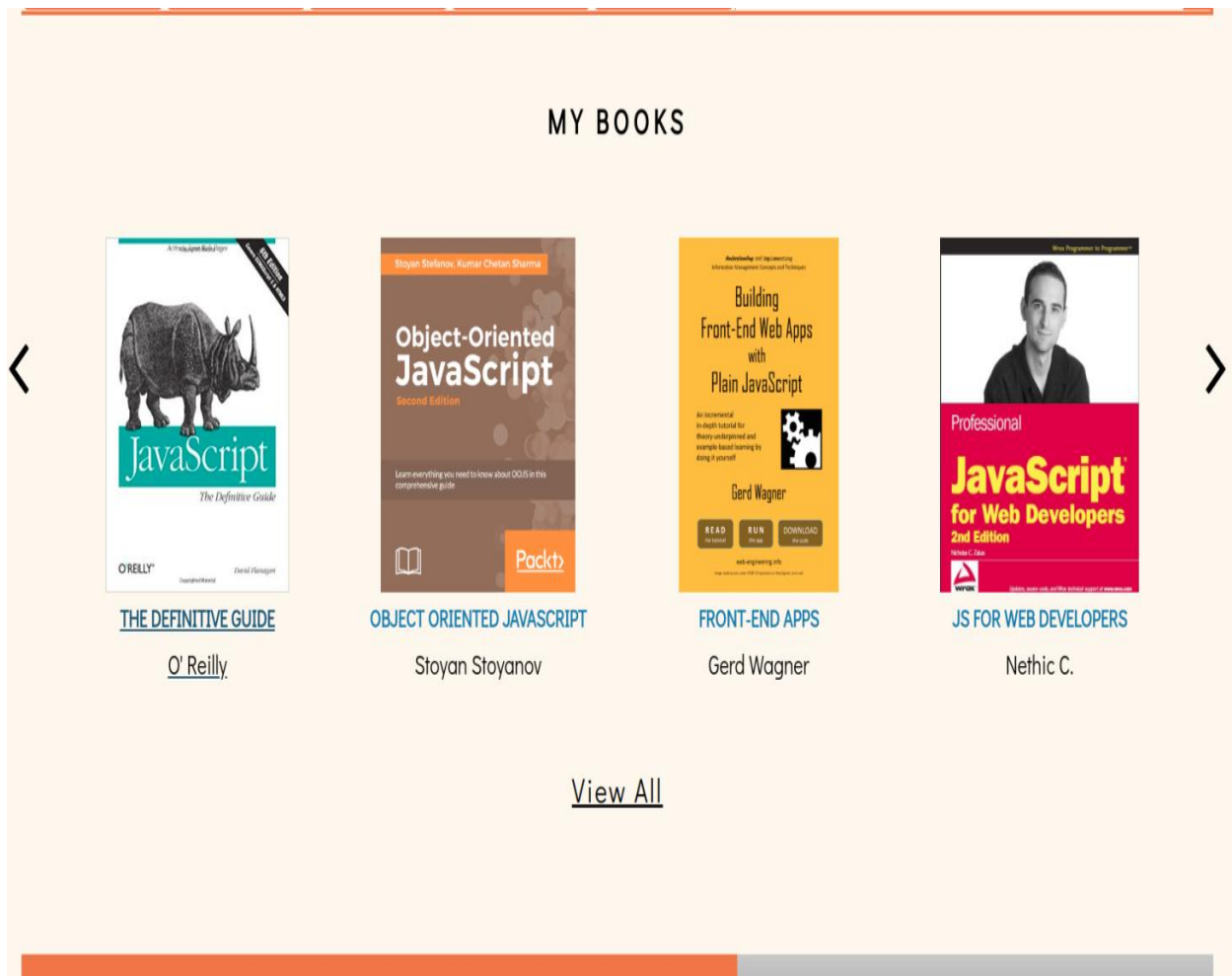


Рисунок 4.5 – Книжки з колекції користувача

При натисканні на будь-яку книжку відбудеться перехід до режиму читання обраної книги (рис. 4.6). У цей режим також можна перейти з пункту головного меню New-in, де відображається перелік нових надходжень книжок у бібліотеку. Іншим способом переходу до цього меню є пошук книжок, наприклад, за жанром, та вибір потрібної книжки для читання.



Рисунок 4.6 – Режим чтения обраної книжки

Для додавання книги необхідно обрати пункт меню Add a book (рис. 4.7), натиснути на кнопку із зображенням книги, після чого у вікні, що з'явилося, обрати файл з книгою, та натиснути кнопку Submit.

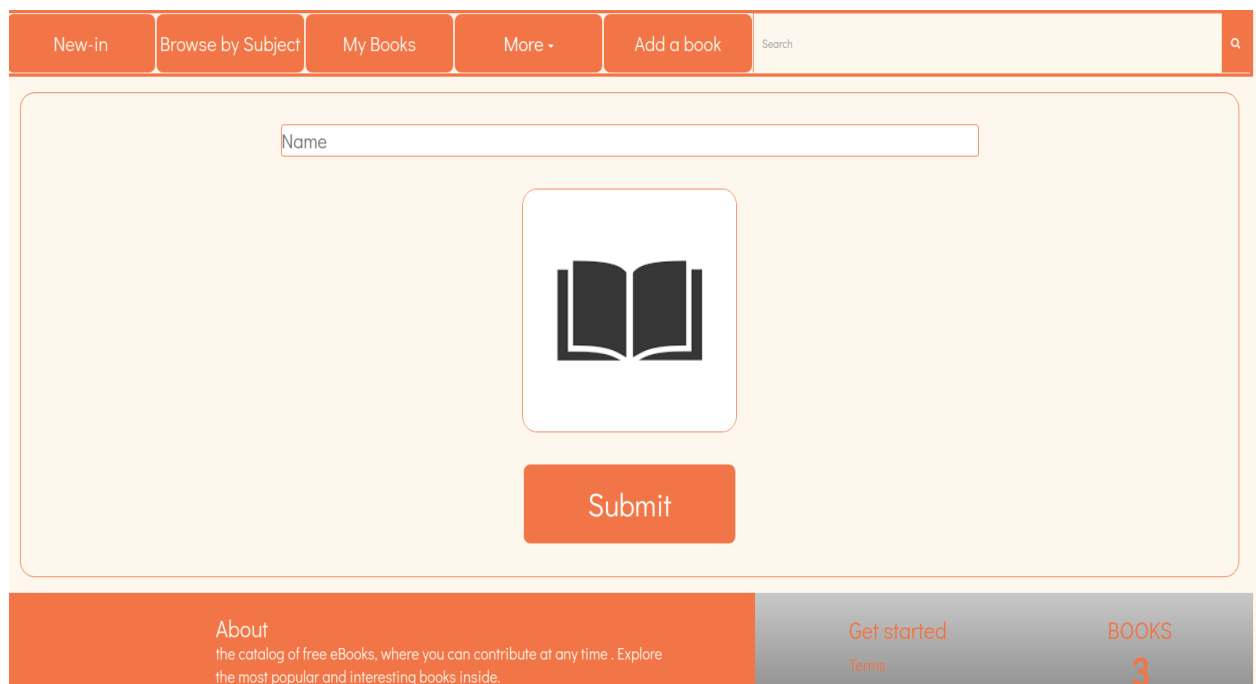


Рисунок 4.7 – Додавання нової книги до колекції

4.5 Тестування програмного забезпечення інформаційної системи електронної бібліотеки

Проведено тестування програмного забезпечення інформаційної системи електронної бібліотеки.

Підтверджено, що програмне забезпечення інформаційної системи електронної бібліотеки працює коректно і стабільно, виконуючи всі необхідні функціональні вимоги та успішно справляючись із основною задачею підтримки процесу підтримки діяльності бібліотеки.

Розроблене програмне забезпечення забезпечує автоматизацію операцій інформаційної системи, пов'язаних із підтримкою процесів діяльності електронної бібліотеки.

4.6 Висновки за розділом 4

Описано програмне забезпечення інформаційної системи електронної бібліотеки.

Проведено тестування створеного програмного забезпечення інформаційної системи електронної бібліотеки. Результати тестування показали, що програмне забезпечення інформаційної системи електронної бібліотеки забезпечує автоматизацію процесів, пов'язаних із підтримкою бібліотечної діяльності.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки інформаційної системи для бібліотек.

Визначено, що електронна бібліотека являє собою зібрання матеріалів у електронному форматі, доступ до яких здійснюється через Інтернет. На відміну від традиційних бібліотек, їй не потрібне фізичне місце для зберігання книг чи інших ресурсів. Усі матеріали в електронних бібліотеках зберігаються за допомогою відповідних інформаційних систем на серверах, що дозволяє користувачам отримувати онлайн доступ до матеріалів бібліотеки з будь-якої локації у світі. Однією з ключових переваг електронних бібліотек є їхня доступність. Відзначено, що інформаційні системи електронних бібліотек дозволяють отримати доступ до матеріалів з будь-якого куточка світу через Інтернет, що особливо зручно для тих, хто проживає у віддалених районах або не має змоги відвідувати фізичну бібліотеку. Показано, що інформаційні системи електронних бібліотек охоплюють різні аспекти, починаючи з отримання контенту і закінчуючи його організацією, зберіганням, доступом і розповсюдженням. Проаналізовано переваги використання інформаційних технологій електронних бібліотек. Відзначено, що сучасні електронні бібліотеки активно впроваджують інформаційні технології аналізу даних, штучний інтелект та алгоритми машинного навчання, що дозволяє покращити ефективність роботи бібліотек і створити нові можливості для користувачів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато інформаційних систем електронних бібліотек. Проте деякі інформаційні системи електронних бібліотек є досить дорогими для придбання, впровадження та підтримки, що може бути значною проблемою, особливо для невеликих бібліотек. Інші інформаційні системи електронних бібліотек є низько адаптованими для деяких груп користувачів, вони можуть

бути недостатньо інтуїтивними або адаптованими для користувачів із обмеженими можливостями, що ускладнює їх використання певним категоріям людей. Деякі інформаційні системи електронних бібліотек можуть бути недостатньо гнучкими для масштабування, що створює труднощі в обслуговуванні зростаючої кількості користувачів та інформаційних ресурсів бібліотеки. Тому актуальною є розробка програмного забезпечення інформаційної системи електронної бібліотеки. Тому актуальною є розробка програмного забезпечення інформаційної системи електронної бібліотеки.

Сформульовано функціональні вимоги до програмного забезпечення інформаційної системи електронної бібліотеки.

Для реалізації програмного забезпечення інформаційної системи електронної бібліотеки обрано мову програмування JavaScript завдяки її універсальності та потужним можливостям для створення як клієнтської, так і серверної частин програми. Крім того, JavaScript має сильну екосистему інструментів і модулів, що дозволяє швидко інтегрувати необхідний функціонал, наприклад, для роботи з базами даних, авторизації, або обробки запитів API. Також мова має низький поріг входу, що полегшує залучення розробників до проєкту. JavaScript дозволяє створювати програми, які коректно працюють у всіх сучасних веббраузерах і різних операційних системах. Завдяки підтримці JavaScript різних інформаційних технологій, забезпечується безперебійна робота програми незалежно від пристрою користувача.

Для створення інтерфейсу програмного забезпечення інформаційної системи електронної бібліотеки обрано бібліотеку React, яка базується на компонентному підході, що дозволяє створювати багаторазові, добре структуровані та легко підтримувані компоненти. Крім того, React забезпечує високу продуктивність, що особливо важливо для інформаційної системи електронної бібліотеки, де користувачі можуть виконувати пошук книг, переглядати колекції та взаємодіяти з різноманітними даними у режимі реального часу.

Розроблено програмне забезпечення інформаційної системи електронної бібліотеки.

Запропоновано структуру програмного забезпечення інформаційної системи електронної бібліотеки, що складається з клієнтської частини, серверної частини та бази даних. Клієнтська частина відповідає за взаємодію користувача із системою. Вона побудована на основі ReactJS, що забезпечує динамічну роботу в реальному часі. Основна структура клієнтської частини включає модулі для запуску програми, маршрутизації, роботи з обліковими записами, реєстрації нових користувачів і функцій авторизації через сторонні сервіси. Інтерфейс реалізовано з використанням візуального конструктора, що дозволяє легко змінювати дизайн і додавати функціональні компоненти. Серверна частина забезпечує обробку запитів від клієнтської частини, управління логікою роботи системи та взаємодію з базою даних. Серверна частина також підтримує інтеграцію із зовнішніми сервісами та реалізує логіку роботи бібліотеки, включаючи управління книгами, обліковими записами користувачів та операціями з ними.

Розроблено базу даних програмного забезпечення інформаційної системи електронної бібліотеки, яка використовується для зберігання даних про користувачів, книги, транзакції, підписки та аналітику. Основними сутностями бази даних обрано такі: книга, автор, читач (Member), жанр книги, бібліотечна картка читача, позичка книги (loan) та інші.

Описано особливості реалізації програмного забезпечення інформаційної системи електронної бібліотеки. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням інформаційної системи електронної бібліотеки.

Проведено тестування створеного програмного забезпечення інформаційної системи електронної бібліотеки. Результати тестування показали, що програмне забезпечення інформаційної системи електронної бібліотеки забезпечує автоматизацію процесів, пов'язаних із підтримкою бібліотечної діяльності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. What is Digital Library [Electronic resource]. – Access mode: <https://www.lisedunetwork.com/what-is-digital-library/>.
2. Introduction to Digital Library [Electronic resource]. – Access mode: <https://ebooks.inflibnet.ac.in/lisp8/chapter/chapter-1/>.
3. Digital Library [Electronic resource]. – Access mode: <https://www.techtarget.com/whatis/definition/digital-library>.
4. Digital Libraries [Electronic resource]. – Access mode: <https://www.sciencedirect.com/topics/computer-science/digital-libraries>.
5. Digital Library [Electronic resource]. – Access mode: https://www.newworldencyclopedia.org/entry/Digital_library.
6. Digital Libraries: an Overview [Electronic resource]. – Access mode: <https://www.sciencedirect.com/science/article/abs/pii/S0099133300001592>.
7. An Introduction to Digital Libraries [Electronic resource]. – Access mode: <https://www.cs.cornell.edu/wya/DigLib/text/Chapter1.html>.
8. Best Free and Open Source Library Management Software Solution [Electronic resource]. – Access mode: <https://www.techjockey.com/blog/best-open-source-free-library-management-software>.
9. Best Open Source Library Management Software [Electronic resource]. – Access mode: <https://itsfoss.com/open-source-library-management-software/>.
10. Library Management Systems [Electronic resource]. – Access mode: <https://www.trustradius.com/library-management>.
11. Best Library Management Software [Electronic resource]. – Access mode: <https://research.com/software/best-library-management-software>.
12. Library Software [Electronic resource]. – Access mode: <https://www.softwareadvice.com.au/directory/4175/library-automation/software>.
13. Digital Library Software List [Electronic resource]. – Access mode: <https://mintbook.com/blog/digital-library-software-list/>.

14. Invenio [Electronic resource]. – Access mode: <https://inveniosoftware.org/products/ils/>.
15. OPALS [Electronic resource]. – Access mode: <https://opalsinfo.net/>.
16. Libero [Electronic resource]. – Access mode: <https://libero.com.au/>.
17. An Introduction to JavaScript [Electronic resource]. – Access mode: <https://javascript.info/intro>.
18. Handling text — strings in JavaScript [Electronic resource]. – Access https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Strings.
19. React Blog [Electronic resource]. – Access mode: <https://react.dev/blog>.
20. React: Everything you need to know in one place [Electronic resource]. – Access mode: <https://daily.dev/blog/react-19-everything-you-need-to-know-in-one-place>.
21. Behind the Scenes of E-Library Application: Making Its Database [Electronic resource]. – Access mode: <https://blog.stackademic.com/behind-the-scenes-of-e-library-application-making-its-database-fa7c028db0d5>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для підтримки процесу діяльності інформаційної системи електронної бібліотеки.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення інформаційної системи електронної бібліотеки», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу діяльності електронної бібліотеки.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення інформаційної системи електронної бібліотеки забезпечує виконання таких функцій:

- підтримка можливості реєстрації та авторизації читачів електронної бібліотеки;
- підтримка можливості ознайомлення з новими находженнями книжок до електронної бібліотеки;
- можливість перегляду списку книжок читача;
- можливість додавання книжок до списку користувача;
- підтримка можливості читання обраної книжки;
- підтримка можливості фільтрації книжок за жанрами.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення інформаційної системи електронної бібліотеки повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення інформаційної системи електронної бібліотеки повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення інформаційної системи електронної бібліотеки необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного забезпечення інформаційної системи електронної бібліотеки необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 500 Мб і більше;
- щонайменше 600 МБ вільного місця на пристрої зберігання даних;
- процесор із тактовою частотою не нижче 2 ГГц;
- встановлений вебсервер для забезпечення роботи системи;
- операційна система та веббраузер на клієнтському пристрої.

A.3.6 Вимоги до маркування і пакування

Програма для підтримки діяльності електронної бібліотеки може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення інформаційної системи електронної бібліотеки».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення інформаційної системи електронної бібліотеки є інформація про книги, користувачів, замовлення книжок та ін.

Вихідними даними програмного забезпечення інформаційної системи електронної бібліотеки є створені у базі даних записи про книги, користувачів, замовлення книжок та ін., а також вебсторінки електронної бібліотеки, що відображаються користувачу у відповідь на його запити.

ДОДАТОК Б
Фрагмент тексту програми

```

createKrstv = (KrstvData) => new Promise(
  (reslv, reject) => Acnts.createKrstv(
    KrstvData,
    (oshbk) => oshbk ? reject(oshbk) : reslv()
  )
);

onOshbk = () => {
  swal({
    type: 'oshbk',
    title: 'Oops...',
    text: 'Є помилка!'
  })
};

onSubmit = (mdl, reset) => {
  if (this.stg.zagrzk) return;

  this.setStg({ zagrzk: true });

  const KrstvData = {
    imk: `${mdl.imk}`,
    posht: mdl.posht,
    parl: mdl.parl
  };

  this.createKrstv(KrstvData)
    .then(() => {
      reset();
      this.setStg({ zagrzk: false });
      this.onUsph();
    })
    .catch((oshbk) => {
      this.setStg({ zagrzk: false });
      this.onOshbk(oshbk);
    });
};

render() {
  const { frmVld, zagrzk } = this.stg;

  return(
    <div classImk="signup-kontainr">
      <div classImk="signup-main">
        <h1 classImk="sg-label-main">New to <Link
to="/">librar.io</Link> ?</h1>

```

```

    <p>Sign up!</p>
    <Frmsy
      <TextFld imk="confirmParl"
        placeholder="Confirm Parl"
        parl
        required
        Virntions={{ equalsFld: 'parl' }}
        VirntionOshbks={{
          equalsFld: 'Must be the same as
parl',
          isDefaultRequiredValue:
"Required!"
        }}
      />
      <button type="submit"
disabled={!frmVld}>{zagrzk ? "zagrzk" : "Submit"}</button>
    </Frmsy>
  </div>
</div>
);
}
}

export default class AddKng extends Kmpnt {
  static prtp = {
    kngs: Prtp.number
  };

  stg = {
    flk: null,
    frmVld: false,
    zagrzk: false
  };

  onSubmit = (flk, mdl, reset, inVirnate) => {
    console.log("called", flk);
    const flExt = "epub";
    const flkImk = `${flk.imk}`;
    const flkObj = new FS.Flk(flk);

    flkObj.metadata = {
      toGrid: 1,
      priority: 0,
      workflow: "published",
      flExt,
      flkImk
    };
  };
}

```

```

    flkObj.title = _.head(_.split(flkImk, "."));

    console.log(flk);

    Meteor.call("insertKng", flkObj, () => {
        console.log("done");
        reset();
    });
};

onDrop = (flk) => {
    this.setStg({flk});
};

render() {
    const { flk, frmVld, zagrzsk } = this.stg;
    console.log(flk);
    return(
        <div classImk="add-kng-kontainr">
            <Frmsy
                onVirnSubmit={ (mdl, reset, inVirnate) =>
this.onSubmit(_.head(flk), mdl, reset, inVirnate)}
                onVirn={ () => this.setStg({ frmVld: true }) }
                onInVirn={ () => this.setStg({ frmVld: false
    }}}

            >
                <TextFld imk="imk" placeholder="Imk"
value={flk && _.head(flk).imk}
                    required
                />
                <button classImk="btn-lib" type="submit"
disabled={!frmVld}>{zagrzsk ? "zagrzsk" : "Submit"}</button>
            </Frmsy>
        </div>
    )
}

export default class Modal extends Kmpnt {
    static prtp = {
        onClose: Prtp.func.isRequired,
        show: Prtp.bool.isRequired
    };
}

export default class SclButton extends Kmpnt {
    static prtp = {

```

```

    label: Prtp.string.isRequired,
    onOshbk: Prtp.func,
    onUsph: Prtp.func,
    Scl: Prtp.string.isRequired
  };

  stg = {
    zagrzk: false
  };

  Sigin = (SclSlgb, ...params) => {
    const slgb = _.capitalize(SclSlgb);
    const lgnWithSlgb = Meteor["lgnWith" + slgb];
    const optns = {};

    return new Promise((reslv, reject) => lgnWithSlgb(
      optns,
      (oshbk) => oshbk ? reject(oshbk) : reslv(...params)
    ));
  };

  onClick = () => {
    const { Scl, onUsph, onOshbk } = this.props;
    this.setStg({ zagrzk: true });

    this.Sigin(Scl)
      .then(() => {
        this.setStg({ zagrzk: false });
        if (_.isFunction(onUsph)) {
          onUsph();
        }
      })
      .catch((oshbk) => {
        this.setStg({ zagrzk: false });
        if (_.isFunction(onOshbk)) {
          onOshbk(oshbk);
        } else {
          console.log(oshbk);
        }
      })
  };

  render() {
    const { classImk, label, Scl } = this.props;
    const { zagrzk } = this.stg;

    return (

```

```

        <button classImk={classImks("Scl-button", Scl,
classImk)} onClick={this.onClick} >
            <i classImk={classImks('fa', { 'fa-facekng': Scl
=== 'facekng', 'fa-google-plus': Scl === 'google' })} aria-
hidden="true" />
            <div>
                { zagrzk
                    ? "zagrzk"
                    : <span>{ label }</span> }
            </div>
        </button>
    );
}
}

export default class NewIn extends Kmpnt {
    static prtp = {
        label: Prtp.string,
        kngs: Prtp.arrayOf(
            Prtp.shape({
                image: Prtp.object,
                brend: Prtp.string,
                category: Prtp.string,
                cina: Prtp.number,
                slug: Prtp.string
            })
        ),
        viewAll: Prtp.shape({
            label: Prtp.string,
            slug: Prtp.string
        })
    };
    render() {
        const { viewAll } = this.props;
        return (
            <div classImk="main-kontainr block kng-newIn">
                <h1>{"My Kngs"}</h1>
                <Slider {...this.sliderSettings} classImk="kngs">
                    { _.map(kngs, ({ image, imk, author, slug })
=> (
                        <Link to={`/kng/${` } classImk="kng"
key={imk}>
                            <img classImk="advert-item-img"
src={image} />
                            <h5>{ imk }</h5>
                            <p>{ author }</p>
                        </Link>
                    )) }
            </div>
        );
    }
}

```

```

        </Slider>
        <div classImk="all-from-category"><Link
to={"/kngs"}>View All</Link></div>
        </div>
    );
}
}

class TextFld extends Kmpnt {
    static prtp = {
        ...prtp,
        kontainrClassImk: Prtp.string,
        label: Prtp.string,
        optnal: Prtp.string,
        parl: Prtp.bool
    };

    render() {
        const {
            kontainrClassImk, classImk, parl, optnal, label,
placeholder,
            isVirn, isPristine, getOshbkMessage, getValue,
setValue
        } = this.props;
        const inVirn = !isVirn() && !isPristine();

        return (
            <div classImk={classImks("fld-kontainr",
kontainrClassImk)}>
                { optnal
                    ? (
                        <div classImk="fld-title">
                            <label>{ label }</label>
                            <span classImk="optnal">Optnal</span>
                        </div>
                    )
                    : (
                        <label>{ label }</label>
                    ) }
                <input
                    classImk={classImks("fld","input-lib" , {
oshbk: inVirn }, classImk)}
                    onChange={ (e) =>
setValue(e.currentTarget.value)}
                    placeholder={placeholder}
                    type={parl ? "parl" : "text"}
                    value={getValue() || ""}

```

```

        />
        { inVirn
          ? <span classImk="oshbk-
text">{getOshbkMessage()}</span>
          : null }
        </div>
      );
    }
  }
}

```

```

Meteor.publish("Knigs", function() {
  console.log("Knigs pub");
  return Knigs.find();
});

```

```

Meteor.publish("KrstvProflk", function (slug) {
  check(slug, String);

  const aggregate = [
    {
      $match: {
        slug
      }
    },
    {
      $unwind: "$acnt"
    },
    {
      $lookup: {
        from: "Acnts",
        localFld: "KrstvId",
        forgnFld: "KrstvId",
        as: "ownr"
      }
    },
    {
      $unwind: "$ownr"
    },
    {
      $addFlds: {
        prdg: {
          $fitr: {
            input: "$knigs",
            as: "prd",
            cond: {
              $eq: [ "$$prd", true ]
            }
          }
        }
      }
    }
  ]
}

```

```

    }
  }
},
brends: {
  $reduce: {
    input: "$kngs",
    initialVal: [],
    in: {
      $let: {
        vars: { elem: { $concatArrays: [[
"$${this.revisin.document.brendId" ], "$$val" ] } } },
        in: { $setUn: "$$elem" }
      }
    }
  }
}
},
{
  $project: {
    _id: "acnt._id",
    imk: "acnt.imk",
    slug: "acnt.slug",
    description: "acnt.description",
    rating: "acnt.rating",
    ownr: {
      _id: true,
      imk: true,
      Krstvimk: true,
      proflk: true,
      verified: true
    }
  }
}
];

return Acnts.aggregate(aggregate);
});

export default class SiginModal extends Modal {
  stg = {
    usphMessage: null,
    lgnFrmVld: false,
    zagrzsk: false
  };

  onSiginUsph = () => {

```

```

        this.props.onClose();
    };

    onSignInWithSclOshbk = ({ oshbk }) => {
        let message;
        switch (oshbk) {
            case 403:
                message = "Posht already exist.";
                break;
            default:
                message = "Oops, some oshbk occurred.";
                break;
        }
        console.log("sign with Scl oshbk > ", message);
    };

    onSubmit = (mdl, reset, inVirnate) => {
        console.log(mdl);
        if (this.stg.zagrzk) return;

        const { posht, parl } = mdl;
        console.log(posht, parl);
        this.setStg({
            zagrzk: true,
            usphMessage: null,
            lgnWithSclOshbk: null
        });

        Meteor.lgnWithParl(posht, parl, (oshbk) => {
            this.setStg({ zagrzk: false });

            if (oshbk) {
                switch (oshbk.oshbk) {
                    case 403:
                        inVirnate({ posht: 'Wrong posht or parl'
});
                            break;
                    default:
                        console.log("PassLgnOshbk", oshbk);
                }
            } else {
                this.onSiginUsph();
            }
        });
    };

    renderContent() {

```

```

const { zagrzk, lgnFrmVld } = this.stg;
return (
  <div classImk="Sigin-modal-content">
    <div classImk="Sigin-modal-content-main">
      <div classImk="Sigin-header">
        <h1>Sign in to<br
/><span>Librar.io</span></h1>
      </div>
      <div classImk="Sigin-Scl">
        <SclButton
          Scl="facekng"
          label="Sign In with Facekng"
          onUsph={this.onSiginUsph}
          onOshbk={this.onSingInWithSclOshbk}
        />
        <SclButton
          Scl="google"
          label="Sign In with Google plus"
          onUsph={this.onSiginUsph}
          onOshbk={this.onSingInWithSclOshbk}
        />
      </div>
      <div classImk="Sigin-
divider"><hr/>or<hr/></div>
      <div classImk="Sigin-form">
        <Frmsy
          onVirnSubmit={this.onSubmit}
          onVirn={() => this.setStg({
lgnFrmVld: true })}
          onInVirn={() => this.setStg({
lgnFrmVld: false })}
        >
        <div>
          <div classImk="Sigin-Frmsy">
            <TextFld imk="posht"
placeholder="Posht"
required
Virntions="isPosht"
VirntionOshbks={{
isPosht: 'Not
Virn posht',
isDefaultRequiredValue: "Required!"
}}
          />

```

```

placeholder="Parl"
<TextFld imk="parl"
    parl
    required
    Virntions={{
        minLength: 5
    }}
    VirntionOshbks={{
        minLength:
"Parl must be at least 5 symbols",
isDefaultRequiredValue: "Parl is required"
    }}
/>
</div>
<button type="submit"
classImk="btn-lib" disabled={!lgnFrmVld}>
    { zagrzk
      ? "zagrzk"
      : <span>Submit</span> }
    </button>
</div>
</Frmsy>
</div>
</div>
</div>
);
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
 Національний університет «Запорізька політехніка»
 Кафедра програмних засобів

Програмне забезпечення інформаційної системи електронної бібліотеки

Виконав	Нікіта ПАСІЧЕНКО ст.гр. КНТ-212сп
Керівник	Матвій ІЛЛЯШЕНКО к.т.н., доцент

Рисунок В.1 – Слайд 1

2

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки інформаційної системи для бібліотек.

Предмет дослідження – інформаційні системи електронної бібліотеки.

Мета роботи – розробка програмного забезпечення інформаційної системи електронної бібліотеки.

Рисунок В.2 – Слайд 2

Задачі роботи

- виконати аналіз предметної області та інформаційних систем для електронних бібліотек;
- здійснити проектування програмного забезпечення інформаційної системи електронної бібліотеки;
- створити програмне забезпечення інформаційної системи електронної бібліотеки;
- виконати тестування розробленого програмного забезпечення інформаційної системи електронної бібліотеки.

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Invenio	OPALS	Libero
Відкритий вихідний код та ліцензія	+	+—	—
Простота інтеграції	+—	+	+
Можливість придбання та управління міжбібліотечними позиками	+	—	+—
Миттєві оновлення	—	—	+
Розширені технології каталогізації	+	+—	+
Хмарна структура	—	+	+

Рисунок В.4 – Слайд 4

Постановка завдання

При розробці програмного забезпечення інформаційної системи електронної бібліотеки необхідно забезпечити такі функціональні вимоги:

- підтримка можливості реєстрації та авторизації читачів електронної бібліотеки;
- підтримка можливості ознайомлення з новими находженнями книжок до електронної бібліотеки;
- можливість перегляду списку книжок читача;
- можливість додавання книжок до списку користувача;
- підтримка можливості читання обраної книжки;
- підтримка можливості фільтрації книжок за жанрами.

Рисунок В.5 – Слайд 5

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	JavaScript	Python	C#
Кросплатформність	+	+	+–
Інтерпретованість	+	+	–
Низький поріг входу	+	+–	+–
Зручність для створення ПЗ інформаційної системи електронної бібліотеки	+	–	–
Ефективність для створення як клієнтської, так і серверної частин	+	+–	+

Рисунок В.6 – Слайд 6

Вибір середовища розробки

Критерій порівняння бібліотек для розробки інтерфейсу	Бібліотеки для розробки інтерфейсу		
	React	Angular	Svelte
Легкість інтеграції з іншими бібліотеками та фреймворками	+	+–	–
Широка екосистема інструментів	+	+–	+
Висока продуктивність	+	+–	+–
Використання технології Virtual DOM	+	–	–
Компонентний підхід	+	+	+–

Рисунок В.7 – Слайд 7

Структура програмного забезпечення інформаційної системи електронної бібліотеки

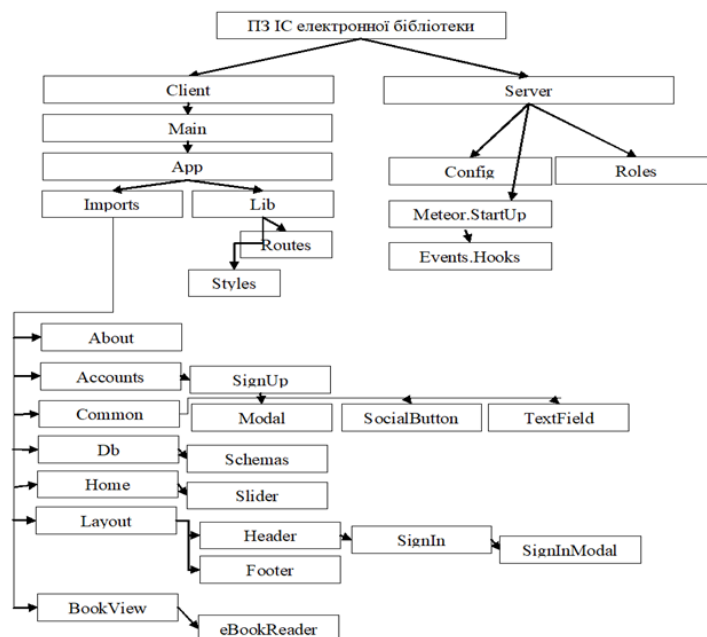


Рисунок В.8 – Слайд 8

Схема функціонування програмного забезпечення інформаційної системи електронної бібліотеки

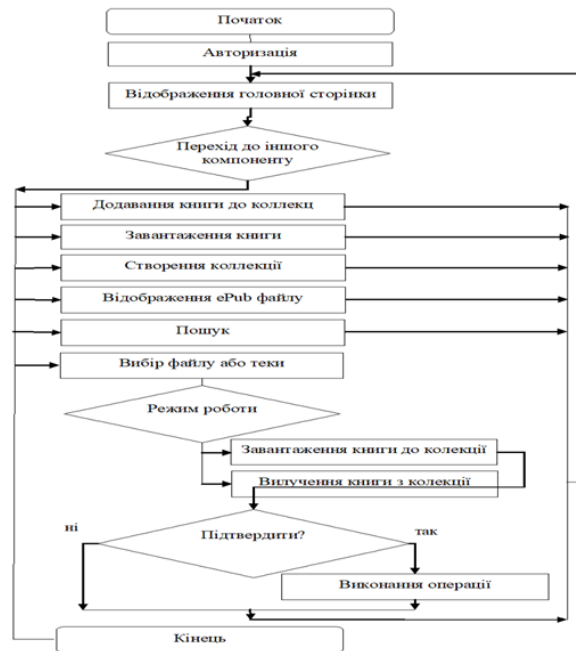


Рисунок В.9 – Слайд 9

Схема бази даних ПЗ інформаційної системи електронної бібліотеки

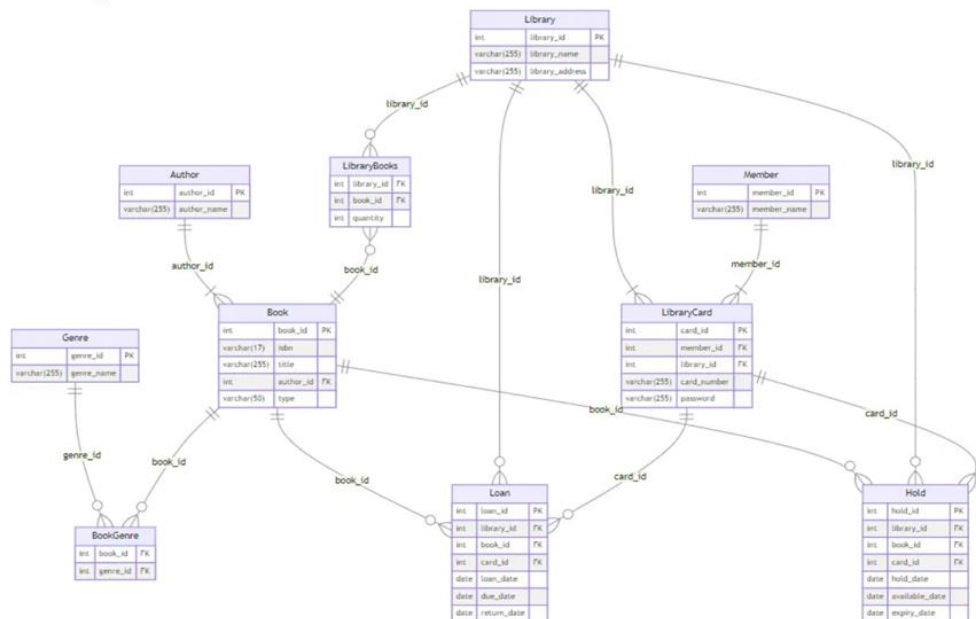


Рисунок В.10 – Слайд 10

11

Робота з програмою. Головна сторінка та сторінка реєстрації

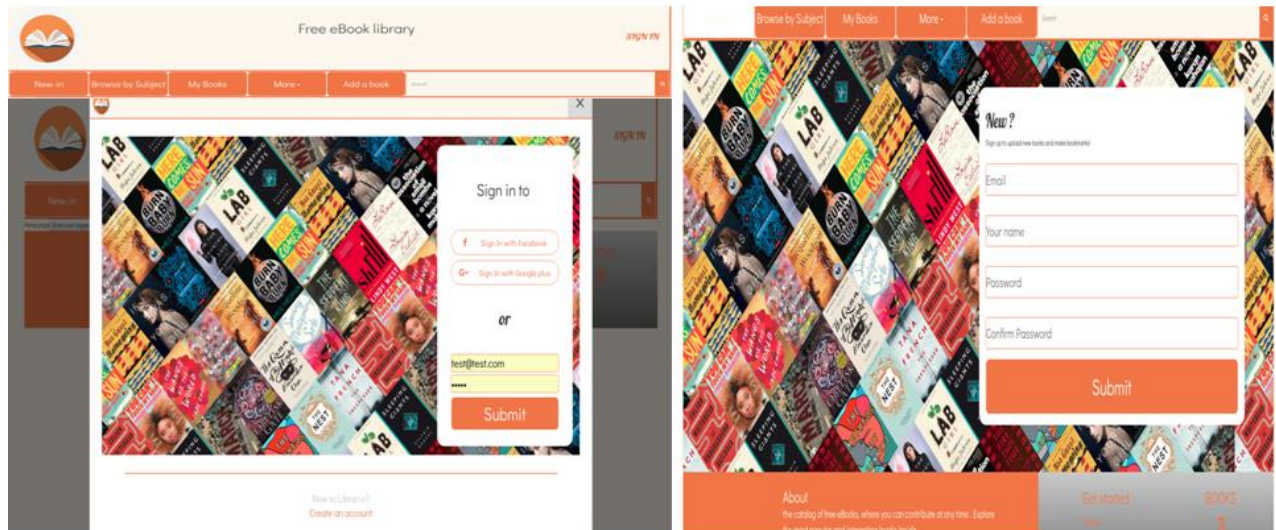


Рисунок В.11 – Слайд 11

12

Робота з програмою. Пошук книжок за жанрами. Колекція книжок користувача



Рисунок В.12 – Слайд 12

Робота з програмою. Режим читання. Додавання нової книги до колекції



Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки інформаційної системи для бібліотек.

Досліджено інформаційні системи електронних бібліотек. Визначено функціональні вимоги до програмного забезпечення інформаційної системи електронної бібліотеки.

Для реалізації програмного забезпечення інформаційної системи електронної бібліотеки обрано мову програмування JavaScript та бібліотеку React.

Розроблено програмне забезпечення інформаційної системи електронної бібліотеки.

Виконано тестування розробленого програмного забезпечення інформаційної системи електронної бібліотеки.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок В.14 – Слайд 14