

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ЧИТАЦЬКОГО КЛУБУ
ЛНТУ**

INFORMATION SYSTEM FOR THE LNTU READING CLUB

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-41
Ярмолюк Василь Русланович

(підпис)

Керівник:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Кваліфікаційну роботу
допущено до захисту
« _____ » червня _____ 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ проф. Н.Черняшук

« _____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Ярмолюку Василю Руслановичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Інформаційна система для читацького клубу ЛНТУ

Керівник роботи к.т.н., доцент Лавренчук Світлана Василівна

затвержені наказом закладу вищої освіти від «28» грудня 2022 року № 982/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 01.06.2023р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Опис інформаційної системи для читацького клубу ЛНТУ

Вимоги до розроблюваної системи

Опис програмної реалізації

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

АНОТАЦІЯ

Ярмолюк В.Р. Інформаційна система для читацького клубу ЛНТУ. Рукопис. Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, трьох додатків.

Перший розділ присвячено огляду предметної області, тут розглядаються основні поняття про веб представлення читацького клубу, їх приклади реалізації та сфери їх використання, наведено багато практичних прикладів. Також в цьому розділі здійснено огляд програм-аналогів (Goodreads, Book Riot, The Rumpus Book Club).

В другому розділі здійснено вибір та обґрунтування засобів розробки. Обрано засоби: HTML, CSS, ReactJS, PostgreSQL, ExpressJS.

Третій розділ присвячено опису розробленої програми її компонентів та бази даних, а також окремих сторінок: блогу, оголошень, адміна, реєстрації та авторизації.

Об'єкт – особливості створення інформаційних систем, зокрема для читацького клубу.

Предмет – інформаційна система для читацького клубу ЛНТУ.

Метою роботи було створення інформаційної системи для читацького клубу ЛНТУ, що надає можливість покращити комунікацію між студентами та викладачами, які мають спільні інтереси та погляди на книжки.

ЧИТАЦЬКИЙ КЛУБ, БАЗА ДАНИХ, ООП, БРАУЗЕР, АВТОРИЗАЦІЯ, АДМІН, КОМЕНТАРІ.

ANNOTATION

Yarmolyuk V.R. Information system for the LNTU reading club. Manuscript. Bachelor's qualifying thesis of the OP "Computer Engineering" majors 123 Computer engineering. Lutsk National Technical University. Lutsk, 2023.

The qualification work consists of an introduction, three sections, conclusions, a list of used sources, and three appendices.

The first section is dedicated to the review of the subject area, here the basic concepts of the web presentation of the reading club, their implementation examples and areas of their use are considered, many practical examples are given. This section also reviews similar programs (Goodreads, Book Riot, The Rumpus Book Club).

In the second section, the selection and testing of development tools was carried out. Selected tools: HTML, CSS, ReactJS, PostgreSQL, ExpressJS.

The third section is devoted to the description of the developed program, its components and database and individual pages: blog, announcements, admin, registration and authorization.

The object is the features of creating an information system in particular for a reading club

The subject is information system for LNTU reading club.

The goal of the work was to create an information system for the LNTU reading club that provides an opportunity to improve communication between students and teachers who have common interests and views on books.

READER CLUB, DATABASE, OOP, BROWSER, AUTHORIZATION, ADMIN, COMMENTS.

ВСТУП

Актуальність теми. У сучасному швидкому світі, де люди часто не мають достатньо часу, книжкові клуби стали популярним способом для індивідуального зв'язку з іншими людьми, котрі люблять літературу. Однак зі зростаючою залежністю від технологій, багато книжкових клубів переходять в Інтернет, створюючи потребу в веб-сайтах книжкових клубів.

Робота спрямована на дослідження розвитку і практичного значення веб-сайтів книжкових клубів, які є платформою для віртуальних зустрічей людей та обговорення книг. Особливість цього дослідження полягає в аналізі ролі технологій у формуванні майбутнього книжкових клубів та в тому, як веб-сайти можуть покращити читацький досвід.

Ця робота є значущою, оскільки вона вносить свій вклад використання технологій у літературному світі. Досліджуючи рівень вивчення та розвитку веб-сайтів книжкових клубів, це дослідження має на меті підкреслити їх потенціал як інструменту для літературної спільноти.

Мета і завдання – здобути практичний досвід створення тематичних веб-сайтів, вибрати та застосувати інструменти для створення та підтримання успішного веб-сайту книжкового клубу.

Об'єкт розробки – особливості створення інформаційних систем, зокрема для читацького клубу.

Предмет – інформаційна система для читацького клубу ЛНТУ.

Завдання кваліфікаційної роботи:

- проаналізувати аналоги читацького клубу;
- здійснити огляд та вибір засобів для створення читацького клубу;
- спроектувати архітектуру системи;
- розробити програмне забезпечення для читацького клубу;
- здійснити опис розробленого веб-застосунку.

РОЗДІЛ 1

ОПИС ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЧИТАЦЬКОГО КЛУБУ ЛНТУ

1.1 Аналіз сфери застосування

Сфера застосування веб-сайту книжкового клубу – об'єднання та створення спільноти книголюбів. Такий веб-сайт надає людям платформу для обговорення книг, обміну досвідом читання та встановлення зв'язків з іншими, хто має подібні інтереси. Веб-сайти книжкових клубів розраховані на широку аудиторію, від не регулярних читачів до більш серйозних книжкових ентузіастів, і можуть охоплювати різні жанри та теми.

Одне з основних завдань веб-сайту книжкового клубу – допомогти людям відкрити для себе нові книги та авторів. Користувачі можуть переглядати рекомендовані книги, бачити, що читають інші, і досліджувати нові назви на основі своїх інтересів. Веб-сайт також може надавати каталог книг, що дозволяє користувачам шукати заголовки та авторів, а також читати резюме та огляди.

Інформаційний потік має бути двонаправленим, що дозволяє користувачам легко спілкуватися та взаємодіяти один з одним. Система також повинна мати можливість сповіщати користувачів про оновлення, нові події або випуски книг.

Веб-сайт читацького клубу повинен містити основні елементи:

- реєстрація користувачів;
- каталогізація книг;
- дискусійні форуми;
- календарі подій;
- системи обміну повідомленнями.

Доцільністю створення книжкового веб-сайту може являтися стимулювання та зміцнення спільноти книголюбів в різних куточках світу.

Користувачі можуть взаємодіяти з іншими людьми, знаходити нових друзів і отримувати задоволення від обміну думками з людьми, які поділяють їхні інтереси. Таким чином, створення книжкового веб-сайту може допомогти підтримувати культуру читання і розвивати спільноту книголюбів.

1.2 Огляд існуючих програмних рішень

Для проектування книжкового веб-сайту потрібно провести порівняльний аналіз існуючих аналогічних програм та систем, що вже використовуються та мають схожий функціонал. Такий аналіз дозволяє з'ясувати переваги та недоліки існуючих систем, а також вибрати оптимальні методи рішення тих проблем, які вже були вирішені іншими розробниками[**Ошибка! Источник ссылки не найден.**].

1.2.1 Застосунок Goodreads

Goodreads [5] – це популярний веб-сайт, який дозволяє користувачам знаходити, оцінювати та рецензувати книги. Його було засновано в 2007 році та придбано компанією Amazon у 2013 році. Goodreads може похвалитися великою базою даних книг із понад 120 мільйонами найменувань на сайті. Користувачі можуть шукати книги за назвою, автором або ISBN, а також можуть переглядати за жанром або темою (рисунок 1.1).

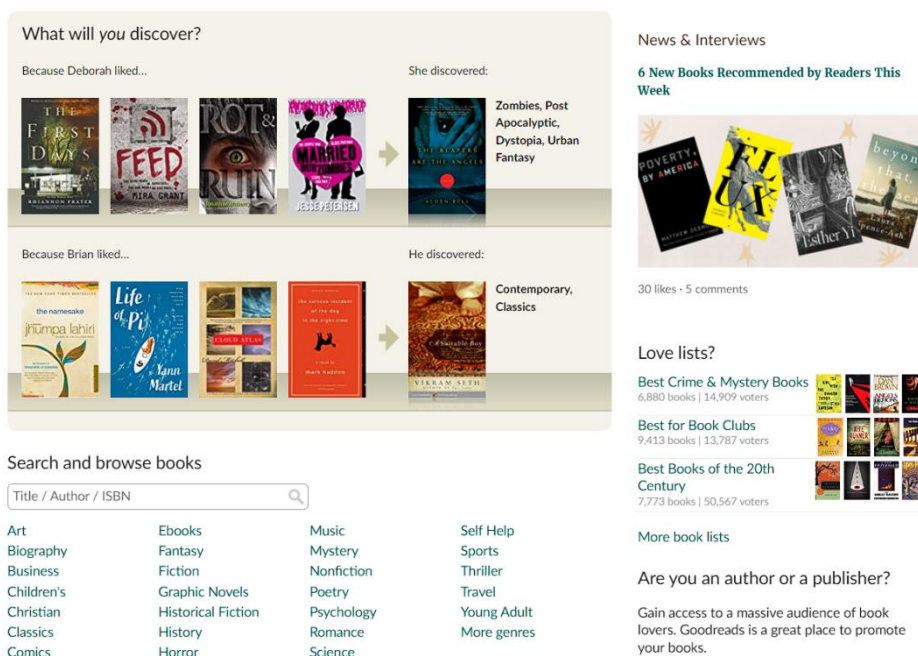


Рисунок 1.1 – Зразок інтерфейсу сайту Goodreads [13]

Однією з найпривабливіших особливостей Goodreads є контент, створений користувачами. Користувачі можуть оцінювати книги за шкалою від однієї до п'яти зірок і залишати письмові відгуки. Ці огляди часто дають цінну інформацію та можуть допомогти користувачам вирішити, читати чи ні певну книгу. Користувачі також можуть створювати полиці для впорядкування своїх книг і відстежувати свій прогрес під час читання.

Ще однією ключовою особливістю Goodreads є його соціальна мережа. Користувачі можуть спілкуватися з друзями, приєднуватися до книжкових клубів і брати участь в обговореннях. Книжкові клуби на Goodreads можуть бути публічними або приватними та організовуватися навколо певного жанру чи теми. Goodreads також надає персоналізовані рекомендації книг на основі історії читання користувача та книг, які він високо оцінив.

Окрім веб-сайту, Goodreads також пропонує мобільний додаток для пристроїв iOS та Android. Додаток дозволяє користувачам отримувати доступ до своїх полиць, оглядів і рекомендацій у дорозі.

Серел недоліків можу відзначити:

– може здатися перенасиченим через велику базу користувачів і великий обсяг даних;

– деякі користувачі можуть вважати соціальний аспект відволікаючим або непривабливим.

Загалом Goodreads є цінним ресурсом для любителів книг. Велика база даних книг і контенту, створеного користувачами, робить його чудовим місцем для відкриття нових книг і спілкування з іншими читачами. Його соціальна мережа та персоналізовані рекомендації також роблять його чудовим інструментом для організації та участі в книжкових клубах.

1.2.2 Застосунок Book Riot

Book Riot – це популярний веб-сайт, який пропонує огляди книг, новини та коментарі на різноманітні літературні теми [10]. Заснований у 2011 році, Book Riot став одним із найбільших і найповажніших книжкових веб-сайтів в Інтернеті (рисунок 1.2).

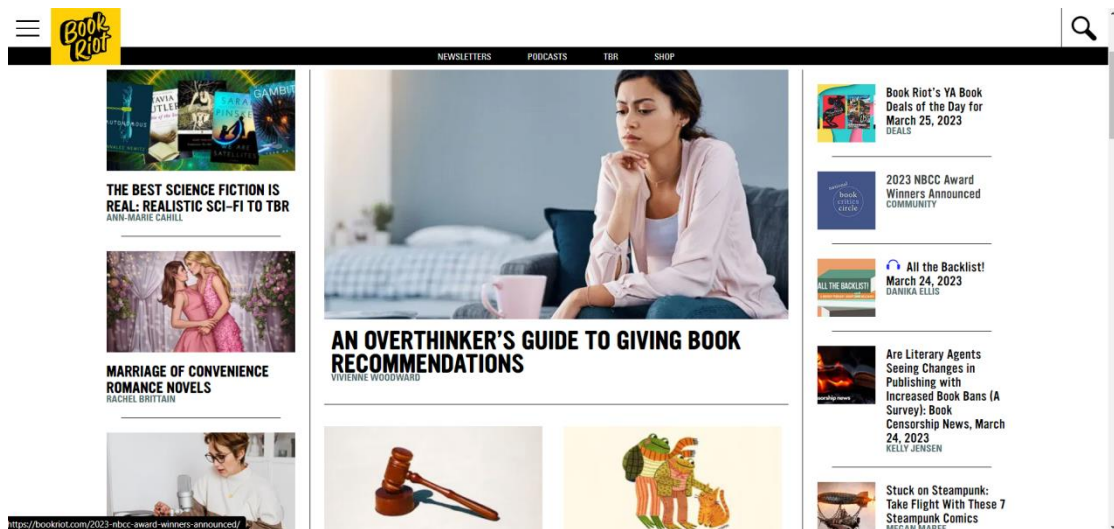


Рисунок 1.2 – Зразок інтерфейсу сайту Book Riot [9]

Однією з видатних особливостей Book Riot є команда з понад 50 учасників, кожен зі своїм унікальним поглядом на літературу. Веб-сайт охоплює широкий спектр жанрів і форматів, від класичної літератури до сучасної фантастики, графічних романів і аудіокниг. Book Riot також містить регулярні колонки, такі як «Рекомендації Riot», де учасники рекомендують книги на певну тему чи тематику.

Окрім оглядів книг і новин, Book Riot також пропонує ряд ресурсів для читачів і письменників. Веб-сайт містить підказки для написання, посібники

для обговорення книжкового клубу та поради для письменників-початківців. Book Riot також містить кілька подкастів, зокрема «Get Booked», де слухачі можуть задавати запитання та отримувати персоналізовані рекомендації книг.

Окрім веб-сайту, Book Riot також пропонує ряд інших послуг і продуктів. На веб-сайті є інтернет-магазин, де читачі можуть придбати книги, книжкову продукцію та передплатні коробки. Book Riot також пропонує кілька онлайн-курсів і семінарів, які охоплюють такі теми, як писання, публікація та літературний аналіз. Веб-сайт активно шукає та наймає письменників із різним досвідом, гарантуючи, що в його статтях та оглядах представлено різноманітні точки зору.

Ще однією перевагою Book Riot є взаємодія з аудиторією. Веб-сайт заохочує читачів брати участь в обговореннях, надсилати рецензії на книги та ділитися своїм досвідом читання. Book Riot також має активну присутність у соціальних мережах, лише в Твіттері нараховує понад 1 мільйон підписників. Цей наголос на розбудові спільноти та залученні допомагає зробити Book Riot яскравим і динамічним місцем для книголюбів.

Незважаючи на численні переваги, Book Riot не позбавлений критики. Деякі читачі відзначили, що веб-сайт може бути величезним, із великим обсягом статей і вмісту. Інші висловлюють занепокоєння з приводу того, що веб-сайт час від часу покладається на заголовки та статті, що містять клік-бейт. Однак ця критика відносно незначна в контексті багатьох позитивних якостей Book Riot.

Book Riot – чудовий ресурс для тих, хто цікавиться книгами та літературою. Його широке охоплення великого діапазону жанрів і форматів робить його цінним джерелом інформації та натхнення як для читачів, так і для письменників.

1.2.3 Застосунок The Rumpus Book Club

The Rumpus Book Club [25] – це книжковий клуб на основі підписки, який щомісяця пропонує своїм учасникам підбрану добірку книг. Заснований у 2009

році письменником Стівеном Елліоттом, The Rumpus Book Club з тих пір став улюбленим місцем для літературної спільноти.

Щомісяця книжковий клуб Rumpus обирає нову книгу для читання та обговорення своїх членів. Добірка часто складається з сучасних творів художньої чи наукової літератури, зосереджених на нових голосах і недостатньо представлених перспективах. Окрім самої книги, учасники також отримують ексклюзивний доступ до інтерв'ю з авторами, дискусійних форумів та іншого пов'язаного вмісту.

Однією з унікальних особливостей The Rumpus Book Club є його наголос на розбудові спільноти та обговоренні. Учасників заохочують спілкуватися один з одним через онлайн-форуми та живі події, включаючи щомісячні телефонні конференції з автором вибраної книги. Такий наголос на спільноті та залученні допомагає створити яскравий і динамічний простір для спілкування читачів і письменників і обміну думками та думками (рисуюнок 1.3).



Рисуюнок 1.3 – Зразок інтерфейсу сайту The Rumpus Book Club [20]

Ще однією сильною стороною The Rumpus Book Club є його прагнення підтримувати незалежних видавців і книготорговців. Багато книг, відібраних для клубу, видаються невеликими видавництвами чи незалежними видавцями,

а The Rumpus Book Club активно просуває та підтримує незалежних книготорговців через партнерство з мережею IndieBound.

На додаток до щомісячної добірки книг, The Rumpus Book Club також пропонує своїм членам низку додаткових функцій і переваг. Це включає доступ до ексклюзивних інтерв'ю з авторами та оглядів книг, знижки на інші продукти та заходи Rumpus, а також можливість брати участь у регулярних письменницьких семінарах та інших літературних заходах.

Однією з найпопулярніших функцій The Rumpus Book Club є його дискусійні онлайн-форуми, де учасники можуть ділитися своїми думками та думками щодо обраної книги та брати участь у живих дебатах і бесідах. Форуми модерують досвідчені співробітники Rumpus, які гарантують, що обговорення залишаються поважними та конструктивними.

Книжковий клуб Rumpus також робить значний наголос на різноманітності та включеності, вибираючи книги, які представляють широкий спектр точок зору та досвіду. Сюди входять книги письменників із різним походженням, а також твори, які стосуються важливих соціальних і політичних питань.

1.2.4 Аналіз архітектури розглянутих програмних рішень

Goodreads, Book Riot і The Rumpus Book Club використовують PostgreSQL як свою систему керування базами даних. Цей вибір, ймовірно, пов'язаний з тим, що ці веб-сайти вимагають високого ступеня масштабованості та гнучкості з точки зору структур даних. Наприклад, Goodreads дозволяє користувачам відстежувати свій прогрес у читанні та створювати власні полиці для своїх книг, для чого потрібна гнучка та потужна система керування базами даних. Книжковий клуб Rumpus також дозволяє користувачам відстежувати свій прогрес у читанні та брати участь в онлайн-обговореннях, для чого потрібна система, яка може обробляти складні запити та великий обсяг контенту, створеного користувачами.

Що стосується методів вирішення, ці веб-сайти використовують комбінацію серверних мов сценаріїв, таких як Ruby та JavaScript, і

інтерфейсних фреймворків, таких як React і Angular. Це дозволяє створювати динамічні та інтерактивні користувальницькі інтерфейси, які покращують взаємодію з користувачем. Крім того, ці веб-сайти часто використовують такі механізми кешування, як Redis, щоб покращити продуктивність і зменшити навантаження на сервер.

Підсумовуючи, під час розробки веб-сайту книжкового клубу вибір системи керування базами даних і методів вирішення залежатиме від вимог веб-сайту щодо масштабованості, гнучкості та продуктивності. MySQL є хорошим вибором для малих і середніх веб-сайтів із простими структурами даних, тоді як PostgreSQL є кращим вибором для великих веб-сайтів зі складними структурами даних. Вибір методів вирішення повинен віддавати перевагу створенню динамічного та інтерактивного інтерфейсу користувача, одночасно оптимізуючи продуктивність і зменшуючи навантаження на сервер.

1.3 Здійснення аналізу та вибору засобів об'єкта проектування

1.3.1 React js

Як зазначено в офіційному слогані, React – це бібліотека для створення інтерфейсів користувача. React не є фреймворком – він навіть не є ексклюзивним для web-розробки. Він використовується з іншими бібліотеками для відтворення в певних середовищах. Наприклад, React Native можна використовувати для створення мобільних додатків [23].

Щоб створювати для web-сторінки, розробники використовують React у тандемі з ReactDOM. React і ReactDOM часто застосовуються в тих самих місцях і для вирішення тих самих проблем, що й інші справжні фреймворки веб-розробки.

React виявився простою бібліотекою, і сьогодні він вважається відмінним навиком розробника. Ми зазначимо кілька переваг роботи з React:

– Спочатку React вирізнявся своєю популярністю, але зараз є інші автономні реалізації віртуального DOM, які можна додати. Це також включає віртуальні моделі DOM, реалізовані на інших платформах JS.

– Візуалізація на стороні сервера поставляється з React, що спрощує керування та відтворення DOM і коду.

– Одна з найкращих порад для будь-якого початківця – якщо розробник працює з React вперше і випадково робить помилку, система підкаже, де помилка в коді та як її виправити.

– React дозволяє розробникам створювати власні події, які потім оновлюються в сховищі даних. Існує багато бібліотек, які допомагають розробникам легко перенести існуючий код із зворотних викликів до більш декларативної моделі програмування.

– React замінює імперативний API на декларативний, що спрощує створення інтерактивних інтерфейсів користувача. React ефективно оновлює та відображає необхідні компоненти для зміни даних.

– React дозволяє розробникам відкривати будь-який файл JavaScript за допомогою компонента React і розуміти можливі стани цього компонента та його поведінку, просто дивлячись на цей файл. React.js також має велику та активну спільноту, що означає, що розробники можуть легко знайти підтримку та ресурси для створення своїх програм. Існує багато доступних бібліотек і інструментів сторонніх розробників, які можна використовувати для покращення та розширення функціональності React.js [4].

ReactJS має багато переваг, він також має деякі недоліки, про які розробники повинні знати, коли вирішують використовувати його для своїх проектів веб-розробки.

Одним із недоліків ReactJS є те, що він має круту криву навчання для початківців. Бібліотека використовує унікальний синтаксис і структуру, для вивчення яких може знадобитися деякий час, особливо для розробників, які не знайомі з веб-розробкою. Однак, як тільки розробники ознайомляться з ReactJS,

вони зможуть використовувати його переваги для створення складних і динамічних веб-додатків [7].

Іншим недоліком ReactJS є те, що його може бути важко налагодити, коли виникають помилки. Бібліотека використовує компонентну архітектуру, що може ускладнити відстеження джерела помилок, коли вони виникають. Однак це можна пом'якшити за допомогою вбудованих інструментів налагодження, доступних у популярних браузерях, або інструментів сторонніх розробників, таких як React Developer Tools [4].

Іншим потенційним недоліком ReactJS є відсутність вбудованої функції для обробки рендерингу на стороні сервера. Хоча за допомогою ReactJS можна досягти відтворення на стороні сервера, для цього може знадобитися додаткове налаштування, що займе багато часу у розробників [21].

1.3.2 Середовище розробки WebStorm

WebStorm – це інтегроване середовище розробки для JavaScript і суміжних технологій. Як і інші IDE JetBrains, воно спрямоване на те, щоб зробити процес розробки приємнішим, автоматизуючи рутинну роботу та допомагаючи користувачам вирішувати складні завдання [6]. Зразок інтерфейсу користувача WebStorm представлений на рисунку 1.4.

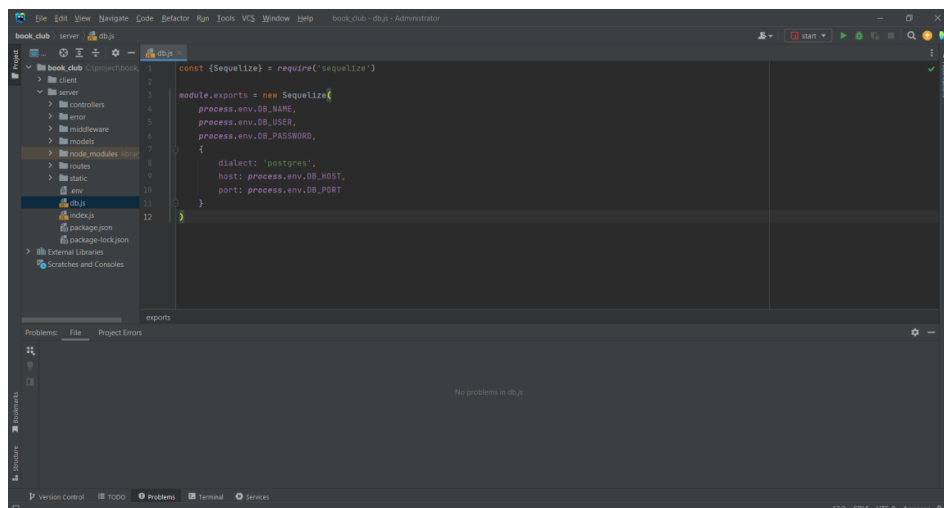


Рисунок 1.4 – Зразок інтерфейсу сайту середовища розробки WebStorm

Однією з головних переваг WebStorm є інтелектуальний редактор коду. Він забезпечує підсвічування синтаксису, автозавершення та перевірку помилок, що полегшує розробникам написання та налагодження коду. Він також містить такі функції, як рефакторинг коду, форматування коду та генерацію коду, які можуть допомогти пришвидшити час розробки та забезпечити якість коду [11].

Ще однією перевагою WebStorm є його інтеграція з різними технологіями веб-розробки, такими як React, Angular, Vue.js і Node.js. Це дозволяє розробникам працювати з обраним технологічним стеком в одному середовищі, полегшуючи керування проектами.

WebStorm також має вбудований налагоджувач, який дозволяє розробникам легко знаходити та виправляти помилки у своєму коді. Він надає широкий спектр інструментів налагодження, таких як точки зупинки, годинники та стеки викликів, які можуть допомогти пришвидшити процес налагодження та покращити загальну якість коду.

Крім того, WebStorm підтримує такі системи контролю версій, як Git, Mercurial і Subversion. Це дозволяє розробникам легко керувати та відстежувати зміни у своєму коді, співпрацювати з членами команди та повертатися до попередніх версій, якщо необхідно [11].

WebStorm являється потужною і багатофункціональною IDE, яка може значно підвищити ефективність і якість проектів веб-розробки. Його інтелектуальний редактор коду, інтеграція з різними технологіями веб-розробки, вбудований налагоджувач і підтримка систем контролю версій роблять його ідеальним інструментом для веб-розробників.

Недоліки Webstorm:

– Вартість: WebStorm є платним IDE, що може бути недоліком для тих, хто шукає безкоштовне або IDE з відкритим кодом.

– Витратність ресурсів: WebStorm вимагає багато ресурсів для безперебійної роботи, що може бути недоліком для тих, хто має старіші чи менш потужні комп'ютери.

- Крута крива навчання: WebStorm може бути важко освоїти, особливо для початківців, які не знайомі з функціями та інструментами IDE.
- Обмежена підтримка мов: Хоча WebStorm підтримує багато мов програмування, він може підтримувати не всі мови, які потрібні для проєкту.
- Обмежене налаштування: WebStorm має обмежені можливості налаштування, що може бути недоліком для тих, хто хоче налаштувати свою IDE відповідно до своїх конкретних потреб .

1.3.3 Система керування базами даних PostgreSQL

PostgreSQL є найпрофесійнішою з реляційних баз даних з відкритим кодом і кілька разів отримувала нагороду «Система баз даних року». Це високонадійна, стабільна, масштабована та безпечна система, яка існує вже більше двох десятиліть. Таким чином, вона стала головним гравцем у світі баз даних з відкритим кодом і кидає виклик великим гравцям, таким як Oracle, Sybase та IBM. PostgreSQL – це професійно розроблене та підтримуване програмне забезпечення, здатне запускати складні програми, які керуються даними [18].

Переваги PostgreSQL величезні. Головною перевагою PostgreSQL є його надійність. Він має перевірену історію стабільності та відомий своєю здатністю обробляти великі обсяги даних без шкоди для продуктивності. Він також надає розширені функції, такі як контроль транзакцій, паралелізм і реплікація, які можуть допомогти забезпечити цілісність даних і запобігти втраті даних.

Ще однією перевагою PostgreSQL є його масштабованість. Його можна використовувати для керування базами даних будь-якого розміру, від невеликих програм до великих систем корпоративного рівня. Він підтримує широкий діапазон типів даних і надає розширені можливості індексування та запитів, які можуть допомогти покращити продуктивність великих баз даних.

PostgreSQL також відомий своїми функціями безпеки. Він забезпечує вбудовані механізми автентифікації та шифрування, а також підтримку шифрування SSL/TLS, яке може допомогти захистити конфіденційні дані від несанкціонованого доступу.

Крім того, PostgreSQL має велику та активну спільноту, що означає, що розробники можуть легко знайти підтримку та ресурси для використання та оптимізації бази даних. Існує багато інструментів і плагінів сторонніх розробників, які можна використовувати для розширення та покращення функціональності PostgreSQL [17].

PostgreSQL – це надійна, масштабована та безпечна реляційна СУБД, яка добре підходить для керування великими обсягами даних. Його розширені функції, підтримка широкого діапазону типів даних і активне співтовариство роблять його популярним вибором серед розробників для створення веб-додатків, корпоративних систем та інших проектів, що містять інтенсивні дані.

Виходячи з вище перерахованих переваг і недоліків в своєму проекті я вирішив використати PostgreSQL, так як розробляв веб застосунок з можливістю легкого розширення функціоналу і бази даних. Виходячи з того що PostgreSQL є кращим вибором для веб-сайтів зі складними структурами даних та зручними засобами для використання їх з React та Java Script.

1.3.4 Мова програмування Java Script

JavaScript, поширене скорочення JS, – це високорівнева інтерпретована мова програмування, яка широко використовується для веб-розробки. Вперше вона була представлена в 1995 році Бренданом Айхом в компанії Netscape Communications, і з тих пір стала однією з найпопулярніших мов для створення динамічних та інтерактивних веб-сторінок [16].

JS є універсальною мовою, яку можна використовувати як на фронтенді, так і на бекенді веб-розробки. На фронтенді вона використовується для створення динамічних та інтерактивних користувацьких інтерфейсів, маніпулюючи об'єктною моделлю документа (DOM), легко інтегрується з HTML та CSS. JS можна використовувати для створення анімації, перевірки даних, введених користувачем, і динамічного оновлення вмісту сторінки на основі дій користувача.

На серверній стороні JS можна використовувати для написання сценаріїв за допомогою популярних фреймворків, таких як Node.js. Це дозволяє

розробникам використовувати єдину мову як для фронтенду, так і для бекенду веб-розробки, що полегшує створення безшовних веб-додатків.

Деякі популярні бібліотеки та фреймворки, побудовані на основі JS, включають React, Vue, Angular та jQuery. Ці бібліотеки та фреймворки надають додаткову функціональність і спрощують виконання таких поширених завдань, як керування станом, HTTP-запити та створення користувацьких інтерфейсів.

JS також має яскраву екосистему пакетів та інструментів, доступних через npm, менеджер пакетів Node.js. Ці пакети можуть допомогти розробникам виконувати такі завдання, як форматування дат, маніпуляції із зображеннями та шифрування та багато інших функцій [24].

Хоча JS є потужною та популярною мовою, вона також має свої проблеми. Оскільки це інтерпретована мова, JS іноді може працювати повільніше, ніж скомпільовані мови, такі як Java або C++. Крім того, може бути складно підтримувати великі бази коду JS без належної організації та інструментів.

1.3.5 JavaScript бібліотека Axios

«Axios – це клієнт HTTP на основі Promise для node.js та браузера. Він може працювати у браузері та nodejs з тією ж базою кодів. На стороні сервера він використовує рідний http-модуль node.js, тоді як на клієнті (браузер) він використовує XMLHttpRequests» [2].

Axios можна використовувати як у браузерних, так і в серверних JavaScript-додатках. Він підтримує широкий спектр методів HTTP-запитів, включаючи GET, POST, PUT, DELETE і PATCH. Axios також дозволяє надсилати дані в різних форматах, таких як JSON, URL-кодовані та FormData.

Однією з головних переваг використання Axios є підтримка HTTP-запитів на основі Promise. Це означає, що можна писати код, який чекає відповіді від сервера, перш ніж виконувати подальші інструкції. Axios також підтримує перехоплювачі, які дозволяють змінювати або скасовувати HTTP-запити до того, як вони будуть відправлені, або відповіді до того, як вони будуть оброблені.

Axios також має вбудовану функцію для обробки міжсайтових атак на скриптинг (XSS) і підробку міжсайтових запитів (CSRF). Він може автоматично додавати токен CSRF до заголовків запитів і перевіряти токен у відповіді сервера.

1.3.6 Механізм на основі HTTP-заголовка CORS

CORS (Cross-Origin Resource Sharing) – це функція безпеки, реалізована веб-браузерами для обмеження веб-сторінок надсилати запити до домену, відмінного від того, який обслуговував оригінальну веб-сторінку [15]. Ця функція безпеки має важливе значення для запобігання доступу зловмисних веб-сайтів до конфіденційних даних або виконання неавторизованих дій від імені користувача. Однак інколи нам потрібно робити перехресні запити з наших веб-додатків, і саме тут на допомогу приходить пакет CORS NPM. CORS NPM – це проміжне програмне забезпечення, яке можна використовувати з Node.js для забезпечення спільного використання ресурсів між джерелами для веб-додатків. Це дозволяє нам вказати, яким доменам дозволено доступ до ресурсів нашої веб-програми та які методи HTTP дозволені для кожного запиту.

Щоб використовувати пакет CORS NPM, нам спочатку потрібно встановити його за допомогою Node Package Manager (NPM). Після встановлення ми можемо імпортувати пакет у наш додаток Node.js і використовувати його як проміжне програмне забезпечення. Ми можемо налаштувати проміжне програмне забезпечення CORS, щоб дозволити певним доменам отримувати доступ до наших ресурсів, вказавши джерело, методи та заголовки в конфігурації проміжного програмного забезпечення.

Пакет CORS NPM забезпечує простий і ефективний спосіб увімкнути спільний доступ до ресурсів між джерелами для програм Node.js. Однак дуже важливо використовувати його обережно та дозволяти доступ лише довіреним доменам, щоб запобігти вразливостям системи безпеки.

1.3.7 Веб-фреймворк Express.js

Express JS – це фреймворк Node.js, призначений для швидкого створення кросплатформних мобільних додатків, веб-додатків, API та полегшення використання node js [1]. Однією з ключових переваг використання Express.js є його простота та гнучкість. Це дозволяє розробникам легко створювати веб-додатки, використовуючи мінімалістичний підхід, водночас надаючи потужні функції для більш складних додатків. Крім того, у Express.js є велика та активна спільнота, що дозволяє легко знайти підтримку та ресурси для вирішення будь-яких проблем або питань, що виникають.

Деякі поширені проблеми безпеки під час використання Express.js включають атаки міжсайтових сценаріїв (XSS), атаки підробки міжсайтових запитів (CSRF) і атаки впровадження SQL. Їх можна пом'якшити шляхом впровадження таких заходів, як перевірка введених даних, використання безпечних файлів cookie, впровадження токенів CSRF і використання підготовлених операторів під час доступу до баз даних.

1.3.8 Node js сервер

Node.js – це кросплатформне середовище виконання JavaScript із відкритим кодом, яке дозволяє розробникам створювати масштабовані та високопродуктивні веб-додатки. Завдяки керованій подіями архітектурі та неблокуючій моделі вводу-виводу Node.js добре підходить для створення веб-серверів, які можуть обробляти велику кількість одночасних з'єднань [22].

Веб-сервери Node.js пропонують ряд переваг порівняно з традиційними веб-серверами, створеними з використанням інших мов або фреймворків. Деякі з ключових переваг використання Node.js для веб-серверів включають:

- Висока продуктивність: Node.js побудовано на неблокуючій моделі вводу-виводу, що дозволяє обробляти велику кількість одночасних з'єднань, не блокуючи цикл подій і не викликаючи проблем з продуктивністю.

- Масштабованість: завдяки своїй керованій подіями архітектурі Node.js має високу масштабованість і може обробляти велику кількість запитів, не споживаючи багато ресурсів.

– Простий у використанні: Node.js створено за допомогою JavaScript, мови, з якою вже знайомі багато розробників. Це полегшує початок роботи з Node.js і швидке створення веб-серверів.

– Велика екосистема: Node.js має велику та динамічну екосистему пакетів і бібліотек, які можна використовувати для додавання додаткових функцій до нашого веб-сервера.

1.4 Постановка завдання на кваліфікаційну роботу бакалавра

Мета цієї кваліфікаційної роботи бакалавра – розробити зручний веб-сайт книжкового клубу, який відповідає потребам і вподобанням його користувачів. Веб-сайт буде розроблено, щоб сприяти обговоренню книг та сприяти єдності книголюбів. Сайт повинен складатись з взаємозалежних частин, а функції кожної частини повинні бути чітко розділені.

Розробка повинна мати наступні можливості:

– Рекомендації щодо книг. Веб-сайт має містити список рекомендованих книг або персоналізовану систему рекомендацій книг, яка пропонує книги на основі історії читання, уподобань і оцінок користувача.

– Книжкові огляди: веб-сайт книжкового клубу також має надати користувачам можливість читати книжкові огляди.

– Блог.

– Оголошення.

– Авторизація.

– Адміністративна панель.

Сайт повинен складатись з двох частин, а саме адміністративної і користувацької.

Адміністративна частина сайту надає можливість додавання книг, постів блогу.

Користувацький інтерфейс веб-сайту повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної на ньому інформації та швидко та логічно переходити до розділів та сторінок.

Для досягнення мети кваліфікаційної роботи потрібно вирішити такі задачі:

- Провести аналіз сайтів-аналогів для розроблення технічного завдання.
- Обрати та налаштувати інструменти реалізації.
- Визначити структуру даних та спроектувати веб-сайт.
- Реалізувати website та розробити інструкції користувача.

РОЗДІЛ 2

ВИМОГИ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Обґрунтування технологій і засобів

Виходячи з всіх переваг і недоліків, на мою думку, обрана схема розробки веб-сайту книжкового клубу з використанням React JS, ExpressJS, PostgreSQL, CORS та JavaScript є найбільш ефективною, зручною і легкою у розробці порівняно з іншими схемами.

Виходячи з вище перерахованих переваг та недоліків, які були перераховані в першому розділі, React JS є найкращим засобом для реалізації такого роду завдання. React JS забезпечує ефективний і масштабований спосіб створення інтерфейсів користувача. Це дозволяє створювати повторно використовувані компоненти, що полегшує підтримку та зміну веб-сайту в майбутньому. React також підтримує рендеринг на стороні сервера, що може покращити продуктивність веб-сайту та SEO.

Виходячи з того що для створення сайту я вирішив використати React JS, на мою думку, найбільш легким у написанні серверної частини сайту є ExpressJS. Він спрощує розробку серверних програм і API, надаючи мінімалістичну, гнучку та масштабовану структуру. Це дозволяє створювати кінцеві точки, які можуть обробляти вхідні запити, обробляти їх і надсилати потрібні відповіді. ExpressJS також підтримує функції проміжного програмного забезпечення, що дозволяє реалізувати такі функції безпеки, як автентифікація та авторизація.

Виходячи з великої розповсюженості та завдяки засобам, що надає PostgreSQL, вирішення такого роду проблем як наприклад розширення та ускладнення бази даних не є складною у реалізації. PostgreSQL – потужна та надійна система керування реляційними базами даних. Він забезпечує підтримку транзакцій ACID, забезпечуючи цілісність і послідовність даних. PostgreSQL також підтримує широкий спектр типів даних, дозволяючи

зберігати різні типи даних, пов'язаних із книгами, користувачами та іншими аспектами веб-сайту.

Виходячи з того що для реалізації клієнтської та серверної частини сайту не має бути необхідності вивчати дві мови програмування, я вирішив використати JavaScript, так як цю мову можна використовувати не тільки на стороні сервера, а і як мову сценаріїв на стороні клієнта, що означає, що він працює в браузері користувача. Це дозволяє створювати інтерактивні та динамічні функції на веб-сайті, такі як анімація, перевірка форм і взаємодія з користувачем. Якщо порівнювати з PHP, який в основному використовується для сценаріїв на стороні сервера і не може забезпечити такий же рівень інтерактивності, JavaScript стає все популярнішим. Це важливо для розробки функцій на стороні клієнта, таких як перевірка форми, взаємодія з користувачем і динамічне завантаження вмісту, серед іншого.

З технічної точки зору прийнята схема розробки з використанням React JS, ExpressJS і PostgreSQL – це широко використовувані та визнані технології, які були ретельно перевірені та оптимізовані для веб-розробки. Використання цих технологій може забезпечити масштабованість, надійність і безпеку веб-сайту.

Крім того, використання архітектури клієнт-сервер і використання JavaScript для функцій на стороні клієнта може покращити продуктивність веб-сайту та взаємодію з користувачем. Використання CORS також може підвищити безпеку веб-сайту, запобігаючи несанкціонованому доступу до ресурсів.

Загалом, обрана схема розробки веб-сайту книжкового клубу з використанням React JS, ExpressJS, PostgreSQL, CORS і JavaScript є вмотивованим і науково обґрунтованим вибором. Він може надати потужну та надійну платформу для взаємодії користувачів із вмістом книжкового клубу, забезпечуючи приємний досвід.

2.1.1 Функціонально-структурна схема проєкту

Даний веб-сайт був створений та протестований за допомогою WebStorm – середовища розробки програмного забезпечення. Крім того, дана система використовує клієнт-серверну архітектуру (рисунок 2.1), яка складається з серверу, веб-клієнта та бази даних.

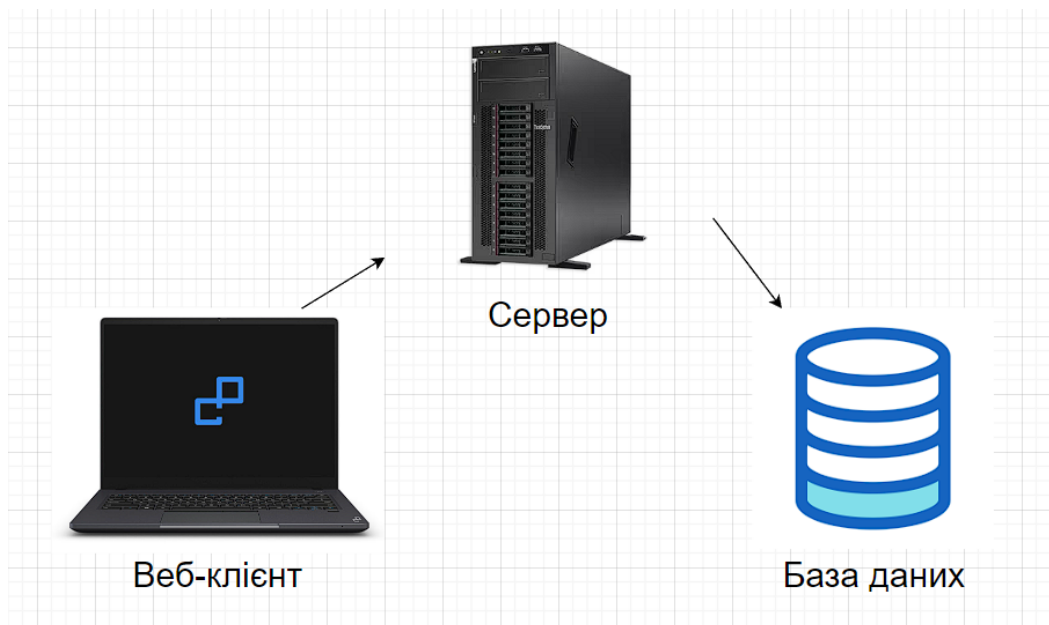


Рисунок 2.1 – Архітектура системи

Дана система складається з таких частин:

- веб-клієнт React.js;
- сервер Express.js;
- база даних MySQL.

2.1.2 Опис функціональності системи

Система «Інформаційна система для читацького клубу ЛНТУ» – це онлайн-ресурс, який надає можливість користувачам читати про книги та обговорювати їх у спільноті. На рисунку 2.2 представлена діаграма, яка описує можливості читача у системі.

Користувачеві надаються такі можливості:

- перегляд книг;

- перегляд постів;
- авторизація в системі;
- додання коментарів;
- перегляд оголошень.

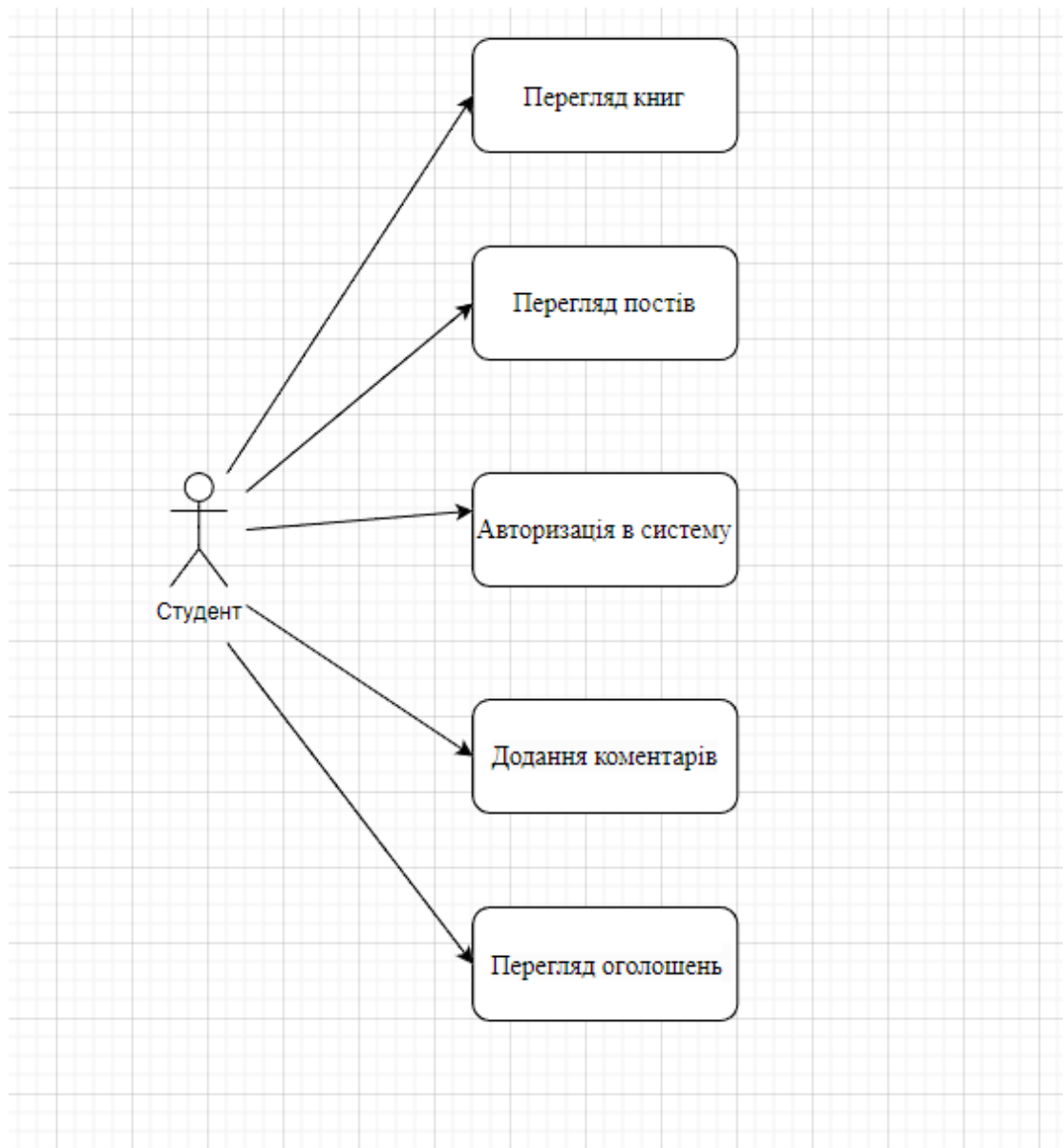


Рисунок 2.2 – Діаграма можливостей веб-застосунку

2.1.3 База даних

База даних побудова на СКБД PostgreSQL. Архітектура бази повністю відповідає тематиці роботи. На рисунку 2.3 представлена схема бази даних.

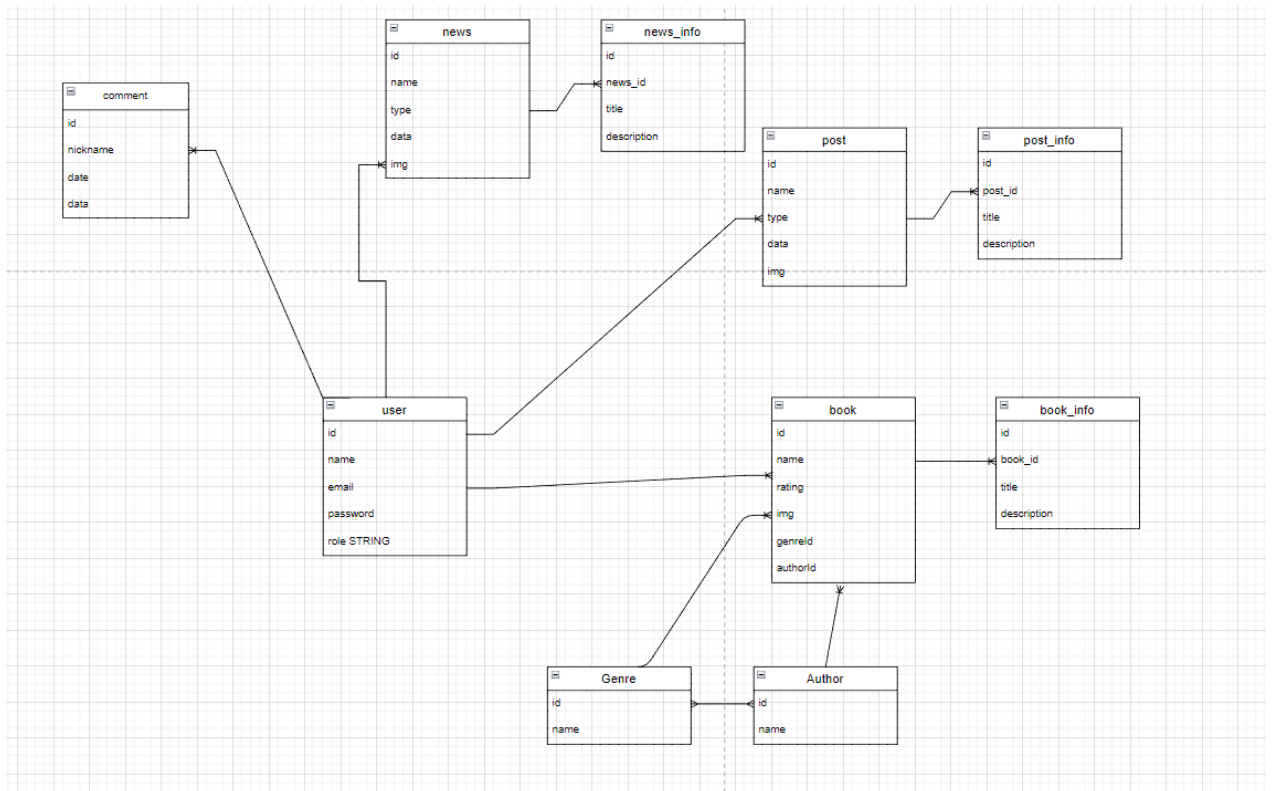


Рисунок 2.3 – Діаграма бази даних

2.1.4 Створення веб-застосунку за допомогою React.js

Для зручності роботи з даними розроблено веб-додаток на мові програмування JavaScript. Це вирішує проблему міжплатформної сумісності та дозволяє запускати програму на будь-якому пристрої з доступом до Інтернету.

Щоб створити проект, необхідно встановити бібліотеку create-react-app, яка спрощує створення коду програми та бібліотеки, а також виконання різноманітних завдань розробки, таких як тестування, компіляція та розгортання [21].

Щоб створити новий проект, потрібно виконати команду «create-react-app project_name», і новий проект буде створено в поточному каталозі з усіма необхідними залежностями та сервером, що полегшить створення та підтримку нашої програми локально. Команда «npm start» запускає сервер, відстежує файли та перебудовує програму, коли до цих файлів вносяться зміни. Параметр --open (або просто -o) автоматично відкриває браузер за адресою <http://localhost:3000/>.

Основними будівельними блоками React є компоненти, які використовуються як складові загальної системи. Компоненти можуть приймати різні параметри, які називаються атрибутами.

Назви та значення стилів зазвичай відповідають тому, як CSS працює у веб-додатках, за винятком угоди про іменування – React використовує стиль верблюжого регістру, наприклад, `backgroundColor` замість `background-color`. Те саме стосується назв подій.

Папка «`components`» містить основні компоненти програми, папка «`configs`» містить налаштування. папка «`Components`» містить сторінки програм, папка «`routes`» містить маршрутизацію, папка «`store`» керує системним станом, а папка «`utils`» містить допоміжні функції (рисунок 2.4).

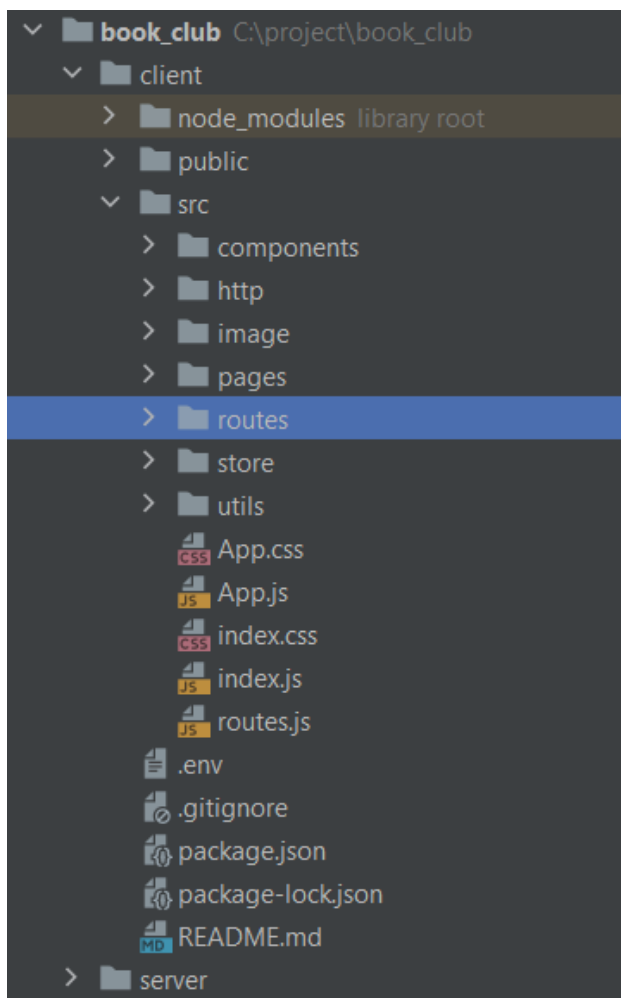


Рисунок 2.4 – Представлена файлова структура веб-клієнта

Для управління станом системи використовується `useContext`. `useContext` – це хук React, який дозволяє читати контекст із нашого компонента та підписуватися на нього.

Було сконфігуровано `store` за допомогою `createContext` хуку `useContext`, в яку прокидаються параметри використовуючи `set` і `get`. Спочатку необхідно створити контекст за допомогою функції `createContext()` з бібліотеки `react`.

Було створено компонент-постачальник (`provider component`), який надає доступ до `store` всім нащадкам цього компонента.

`Store` (рисунок 2.5) складається з `BlogStore`, `BookStore`, `NewsStore` і `UserStore`, котрі в свою чергу складаються з `get` і `set`, які надсилають і приймають дані з програми в `store`, є також єдиним джерелом інформації для `store`.

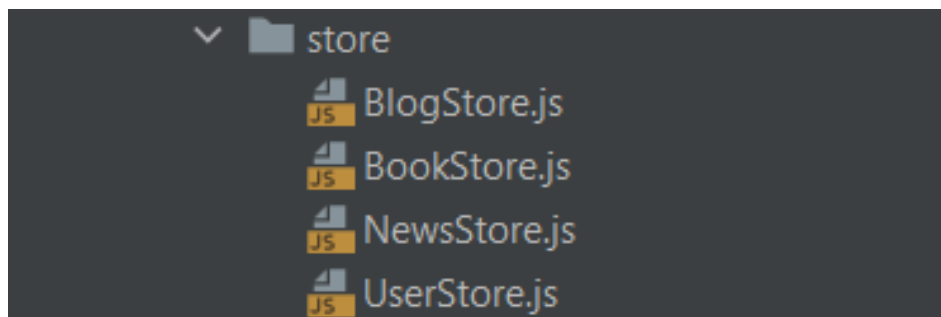


Рисунок 2.5 – Файлова структура системи керування станом системи

Як механізм маршрутизації використовувався компонент `Router` з бібліотеки `React Router`. Компонент маршрутизатора очікує лише один дочірній елемент. Він приймає властивість `history`, яка дозволяє легко контролювати історію сеансів програми. Компонент програми було створено для відтворення всієї програми.

Для стилізації використовувалася бібліотека `styled-components`, а також файли `CSS`, що дозволяють використовувати різні теми зі стилями.

Щоб опрацювати сповіщення, компоненти проміжного програмного забезпечення були написані так, щоб видавати помилки, коли введено неправильний пароль або електронну адресу.

2.1.5 Створення серверної частини за допомогою Express.js

Платформа Node.js була використана при створенні сервера, що полегшує розробку служб REST API за допомогою мови програмування JavaScript.

Щоб запустити проект, необхідно встановити Express, сформувати екземпляр програми та вказати порт для запуску програми.

Щоб обробляти HTTP-запити email в Express.js версії 4 і вище, необхідно встановити модуль проміжного програмного забезпечення, відомий як `body-parser`.

Модуль `body-parser` витягує повний розділ тіла вхідного потоку запитів і призначає його `req.body` [1].

Це проміжне програмне забезпечення колись було частиною Express.js, але тепер його потрібно інсталивати окремо.

Модуль `body-parser` аналізує дані, передані у форматі JSON, буфер, рядок і кодування URL-адреси через HTTP-запит.

Щоб дозволити запити з іншого домену на сервер, потрібно встановити модуль `cors`.

Щоб вилловлювати помилки, повинні бути написані функції проміжного програмного забезпечення (функція проміжної обробки), яка перехоплює помилки та передає їх у результати. Функція проміжної обробки `express.static` використовується для доставки статичних файлів. Щоб дозволити доступ до файлів, необхідно вказати каталог, де знаходяться статичні файли.

Щоб автоматично перезапускати сервер щоразу, коли файл було змінено, потрібно встановити модуль `nodemon`.

Щоб запустити сервер, слід запустити команду `nodemon app.js`.

Сервер виконує різноманітні функції, зокрема:

- спілкування з базою даних;
- завантаження файлів;
- розсилка електронних листів;
- використання HTTP-запитів для взаємодії зі стороною клієнта.

Папка `routes` містить модуль маршрутизації та системні модулі, а модуль бази даних знаходиться в файлі `db`. Папка `public` містить статичні файли. Папка `node_modules` містить залежності, встановлені для розробки сервера. Файл `package.json` містить список підключених модулів для конкретного проєкту за допомогою `npm`, а також команди, необхідні для запуску проєкту.

Нижче перераховані модулі в `package.json`:

- версія аналізатора тіла: 1.18.3;
- версія `cors`: 2.8.5;
- експрес-версія: 4.16.4;
- версія `multer`: 1.4.1;
- версія `mysql`: 2.17.1;
- версія `nodemailer`: 6.1.1;
- версія `nodemon`: 1.19.0.

На малюнку 2.6 зображено файлову структуру сервера.

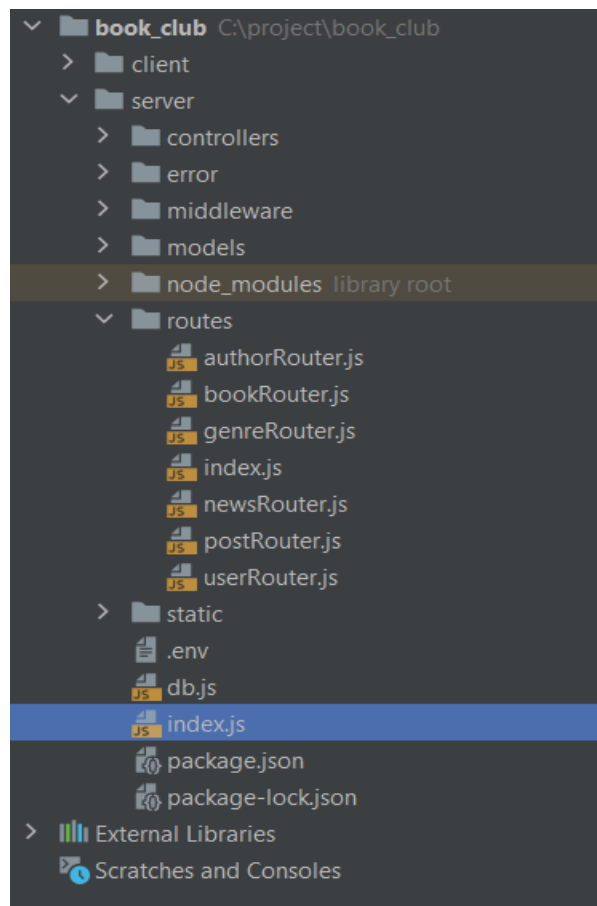


Рисунок 2.6 – Файлова структура сервера

Щоб додати функцію підключення бази даних PostgreSQL до програми Express, усе, що нам потрібно зробити, це завантажити модуль `mysql` у нашу програму.

Щоб завантажити файли на сервер, нам потрібно встановити модуль `multer`, налаштувати тип і розмір файлу, вказати нове ім'я файлу та каталог, де він буде зберігатися.

Однією з найбільш істотних відмінностей, яку було відмічено, є те, що Express надає маршрутизатор за замовчуванням. Нам не потрібно вручну аналізувати URL-адреси, щоб вирішити, що робити, натомість ми визначаємо маршрутизацію програми за допомогою `app.get`, `app.post`, `app.put` тощо, і вони транслюються у відповідні HTTP-запити які описані в окремих `js` файлах (рисунок 2.7)

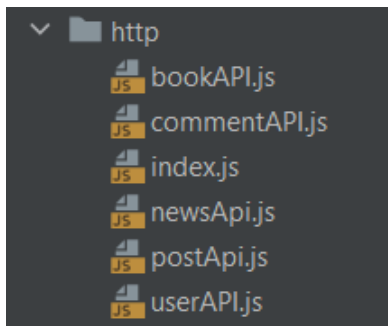


Рисунок 2.7 – Файлова структура арі файлів

Однією з найпотужніших концепцій, які реалізує Express, є шаблон Middleware (рисунок 2.8).

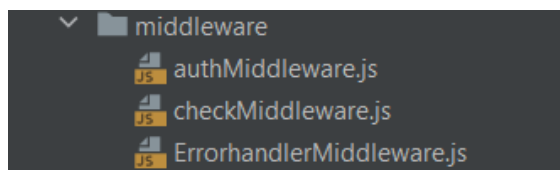


Рисунок 2.8 – Файлова структура middleware файлів

Щоб обробляти помилки в Express, нам потрібно створити спеціальне проміжне програмне забезпечення обробки помилок із чотирма вхідними

параметрами. Обробник помилок має бути останньою функцією, доданою за допомогою `app.use`, і наступним має зворотний виклик. Його можна використовувати для об'єднання кількох обробників помилок.

2.1.6 Додавання інформації на сервер

Для додавання інформації на сервер було використано Bootstrap як один з найпопулярніших фреймворків для розробки веб-інтерфейсів, який надає велику кількість готових компонентів для швидкої та простої розробки веб-додатків. Один з таких компонентів – це спливаючі вікна, які було використовувано для додавання інформації на сервер (рисунок 2.9).

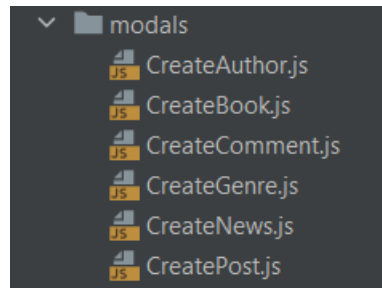


Рисунок 2.9 – Файлова структура файлів спливаючих вікон

Щоб використовувати спливаючі вікна Bootstrap для додавання інформації на сервер, спочатку потрібно створити форму для введення даних. Для цього можна використати компонент форми Bootstrap та його елементи, такі як `input`, `textarea` та кнопка `Submit`.

Наступним кроком є створення спливаючого вікна, яке буде містити створену форму. Для цього було використано компонент `Modal` Bootstrap та його елементи, такі як `Modal Header`, `Modal Body` та `Modal Footer`. Модальне вікно може бути відкрите при натисканні на кнопку або при певній події на сторінці.

Після створення форми та спливаючого вікна, було додано JavaScript-код, який буде відправляти дані з форми на сервер. Для цього було використовуватано бібліотеку `Axios`, яка дозволяє здійснювати запити на сервер за допомогою `AJAX`.

Використання спливаючих вікон Bootstrap для додавання інформації на сервер досить просте і зручне. Вони дозволяють швидко створити форму для введення даних, а модальне вікно додає можливість зручної взаємодії з користувачем.

2.1.7 Файли контролери

Файли контролера є важливою частиною даного проекту, оскільки вони допомагають керувати потоком даних між сервером і клієнтом. По суті, файл контролера – це сценарій на стороні сервера, який обробляє вхідні запити від клієнта та відповідає відповідним чином. В даному проєкті для кожної окремої таблиці було підключено окремий файл (рисунок 2.10).

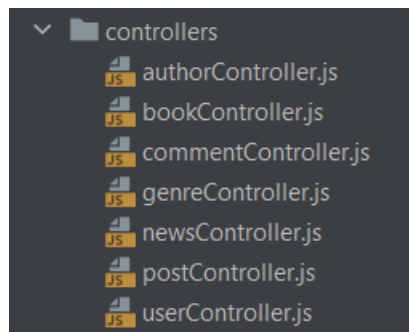


Рисунок 2.10 – Файлова структура контролерів

Однією з основних функцій файлу контролера є надання доступу до певних ресурсів або функцій на сервері. Наприклад, якщо користувач хоче подати форму або отримати дані з бази даних, файл контролера обробить запит і поверне актуальну відповідь.

Щоб використовувати файл контролера для доступу до сервера, зазвичай потрібно виконати такі дії:

- Визначити маршрути. Першим кроком у створенні файлу контролера є визначення маршрутів, які оброблятиме сервер. Ці маршрути визначають шаблони URL-адрес, які клієнт може використовувати для доступу до різних функцій на сервері.

- Визначити функції контролера. Коли ми визначили маршрути, нам потрібно буде створити функції контролера, які оброблятимуть вхідні запити

для кожного маршруту. Ці функції зазвичай виконують деякі дії на сервері, такі як запит до бази даних або обробка надсилання форми [14].

- Реалізувати логіку контролера. Логіка контролера – це код, який виконується, коли надходить запит для певного маршруту. Ця логіка включає обробку даних запиту, виконання будь-яких необхідних операцій на стороні сервера та повернення відповіді клієнту.

- Підключити контролер до сервера. Це було реалізовано завдяки імпорту файлу контролера в головний серверний файл і зіставлення маршрутів із відповідними функціями контролера.

РОЗДІЛ 3

ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Авторизація користувачів

Для використання реалізованого веб-додатку необхідно мати встановлений будь-який інтернет-браузер. Ви повинні переконатися, що ваша система відповідає цим вимогам. Хоча зазвичай це не проблема, коли мова йде про вимоги до обладнання, ви можете помітити, що це зовсім інша історія, наприклад, коли мова йде про підтримувані операційні системи. Користувачі Firefox у Windows 2000, наприклад, помітять, що вони не зможуть оновити Firefox 12 до 13 найближчим часом, оскільки Mozilla зменшила підтримку цієї операційної системи, починаючи з цієї версії браузера. Підтримка браузера Google Chrome для операційних систем починається з Windows XP SP2, OS X 10.5.6, Ubuntu 10.04, Debian 6, OpenSuse 11.3 і Fedora Linux 14.

Для забезпечення доступу до системи лише авторизованих користувачів розроблено авторизацію (рисунок 3.1).

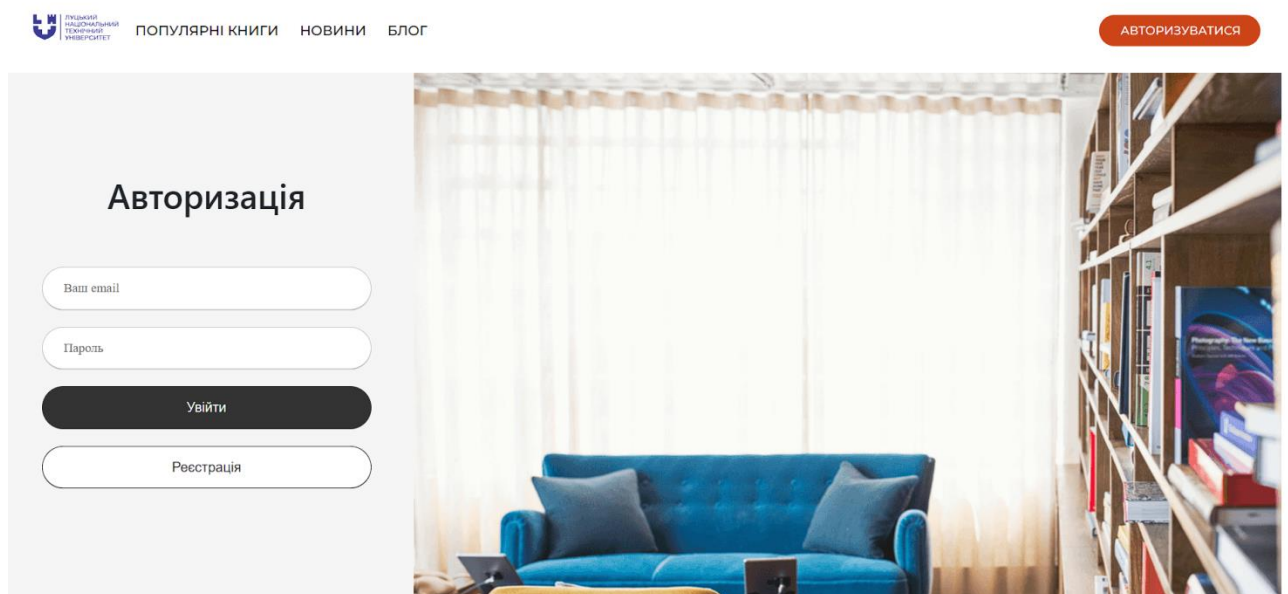


Рисунок 3.1 – Сторінка авторизації

Для зареєструватися було розроблено сторінку реєстрації (рисунок 3.2).

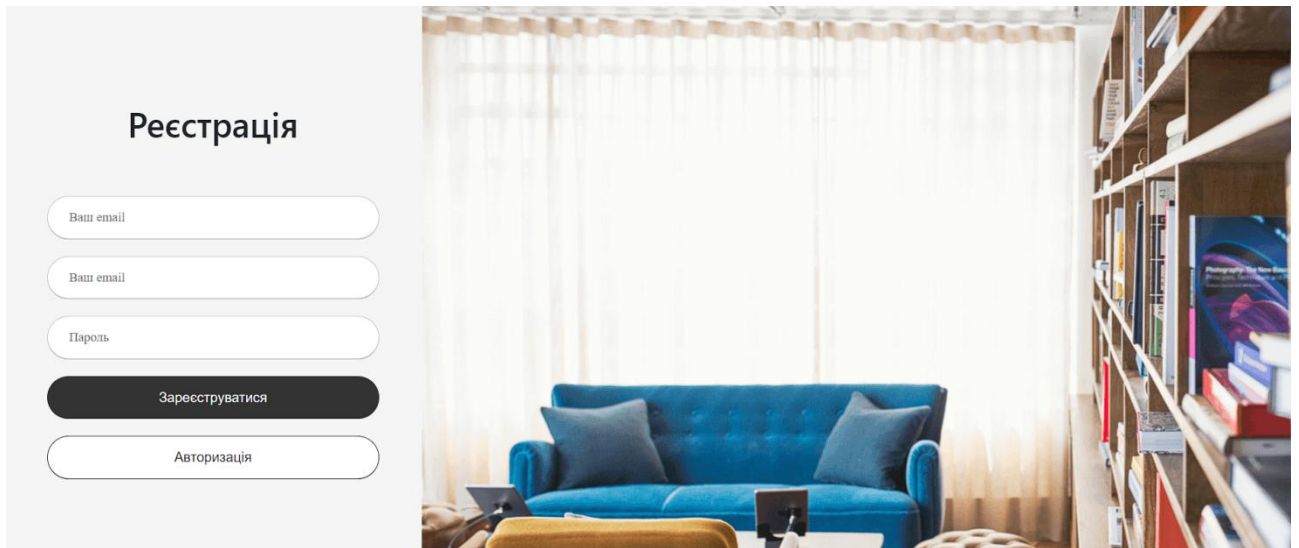


Рисунок 3.2 – Сторінка реєстрації

3.2 Сторінка найпопулярніших книг

Після успішної авторизації користувач має змогу переглядати запропоновані книги та залишати коментар (рисунок 3.3).



Рисунок 3.3 – Сторінка найпопулярніших книг

Студент має можливість переглядати додаткову інформацію про книгу додаткова інформація про книгу може бути корисною як для читачів, які збираються прочитати книгу, так і для тих, хто вже її прочитав.

Перш за все, додаткова інформація про книгу може допомогти читачам визначити, чи варто читати цю книгу. Наприклад, рецензії, огляди і рекомендації можуть дати читачам уявлення про те, чого вони можуть очікувати від книги, і допомогти їм зробити відповідне рішення про те, чи читати цю книгу. Також додаткова інформація про автора, його стиль письма та інші творчі роботи можуть стати корисними для тих, хто не знайомий з автором.

Для тих, хто вже прочитав книгу, додаткова інформація може допомогти глибше зрозуміти твір і відкрити нові аспекти його інтерпретації (рисунок 3.4).



Кобзар

Опис

«Кобзар» — назва першої книги-збірки поетичних творів Тараса Шевченка 1840 року. У наш час під назвою «Кобзар» видають повне зібрання українських поезій Шевченка з додатком двох російських поем «Тризна (Безтапанний)» та «Сліпа». Перші «Кобзарі», до 1908 року, містили твори тільки українською мовою. Всього за життя Шевченка створив 235 поетичних творів: віршів, поем та балад, при цьому, деякі з віршів поета мають декілька дуже відмінних один від одного варіантів, кожен з яких вважається окремим твором.

Історія

Писати вірші Т. Г. Шевченко почав ще кріпаком, за його свідченням, у 1837 році і через рік, коли він вже став вільною людиною та був зарахований студентом до Петербурзької академії мистецтв, про його пристрась до поезії стало відомо поету Є. П. Гребінці, з яким Шевченко був знайомий з 1835 року. — під враженням від звістки про смерть у Полтаві І. П. Котляревського Шевченко написав йому посвяту «На вічну пам'ять Котляревському», яку разом з чотирма іншими поезіями передав Гребінці для публікації в його альманасі «Ластівка»

В кінці 1839 року Гребінка познайомив Шевченка зі своїм приятелем по Ніжинській гімназії відставним штаб-ротмістром, полтавським поміщиком П. І. Мартосом, небайдужою до літератури людиною, гімназійним товаришем М. В. Гоголя і дописувачем у літературні альманаси[2]. За рекомендацією Гребінки Мартос замовив у Шевченка свій акварельний портрет і під час одного із сеансів, коли йому довелося позувати у квартирі художника, він побачив рукописи з його віршами. Вражений прочитаним, Мартос, отримавши схвальний відгук про них Гребінки, вирішив видати вірші Шевченка окремою книжкою-збіркою. Як згадував Мартос, Шевченко не відразу погодився на видання, його довелося умовляти[2][1]. Надалі думки й настрої Шевченка перед виданням збірки відбилися у вірші «Думи мої, думи мої».

Рисунок 3.4 – Додаткова інформація

Прочитавши книгу студент також висловити свою думку щодо книги водночас оцінити (рисунок 3.5).

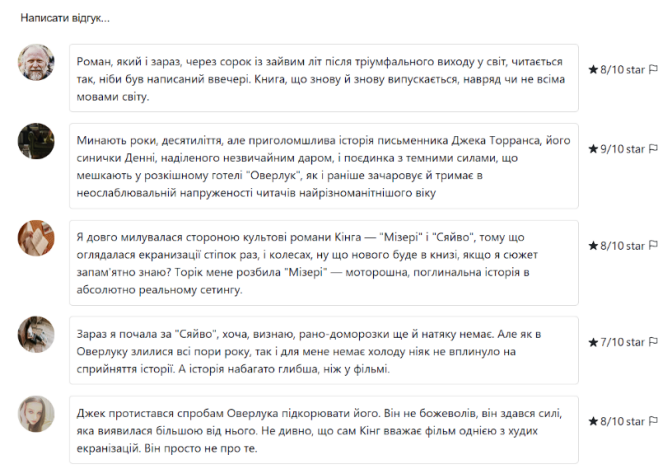


Рисунок 3.5 – Коментарі

Додавання коментарів здійснюється завдяки спливаючому вікні при натисканні кнопки «Написати відгук» (рисунок 3.6). При натисканні висвітлюється спливаюче вікно яке пропонує написати відгук і оцінити вище описану книгу а також додати аватарку що буде розміщена біля коментарю.

Додати новий коментар



введіть текст

0

Вибрати файл

Файл не вибрано

Close

Agree

Рисунок 3.6 – Спливаюче вікно додавання нового коментаря

3.3 Сторінка панелі адміністратора

Панель адміністратора, також відома як адміністративна інформаційна панель або панель керування в даному проекті забезпечує централізоване додавання інформації щодо книг і авторів жанрів і також інформація щодо статей і оголошень веб-сайту (рисунок 3.7).



Рисунок 3.7 – Панель адміністратора

Щоб зайти в адміністративну панель треба авторизуватися за спеціальною електронною адресою. Після цього натиснути кнопку «Адмін панель» у верхньому правому куті головної сторінки і з'являється п'ять стрічок при натисканні на які вилетить спливаюче вікно (рисунок 3.8).

 A floating window titled "Додати нову книгу" with a close button (X) in the top right corner. The window contains the following elements:

- Two blue buttons with white text and dropdown arrows: "жанр" and "автор".
- A text input field with the placeholder "Введіть назву книги".
- A text input field with the placeholder "0".
- A text input field with the placeholder "Введіть короткий опис".
- A file upload section with a button "Вибрати файл" and the text "Файл не вибрано".
- A button "Додати нову властивість".
- Below that, three input fields: "Введіть назву властивості", "Введіть опис властивості", and a red "Видалити" button.
- At the bottom right, two buttons: "Close" (red) and "Agree" (green).

Рисунок 3.8 – Спливаюче вікно для додавання інформації про книгу

При додаванні нових авторів і жанрів вони з'являються в панелі додавання книг при натисканні будуть запропоновані варіанти при натисканні на кнопку «жанр» чи «автор».

3.4 Сторінка оголошень

Оголошення про події в читацькому клубі можуть бути корисними для тих, хто хоче долучитися до зустрічей і обговорень, які проводяться в рамках цього клубу. Такі оголошення можуть містити інформацію про дату та час зустрічі, тему, яку будуть обговорювати, місце проведення, а також контактну інформацію для тих, хто хоче долучитися до клубу.

Зустрічі читацьких клубів можуть мати різний формат. Наприклад, це може бути клуб, який збирається раз на місяць, щоб обговорити одну конкретну книгу. Або ж це може бути клуб, який збирається щотижня, щоб обговорити кілька книг, а також обмінятися враженнями та рекомендаціями про книги, які вони читають в цей час. В даному проекті сторінка оголошень являє собою шість останніх оголошень доданих через адміністративну панель (рисунок 3.9).

Кожне оголошення складається з теми короткого опису оголошення дати публікації. Натиснувши на оголошення ви попадете на сторінку де описане дане оголошення (рисунок 3.10).

Оголошення

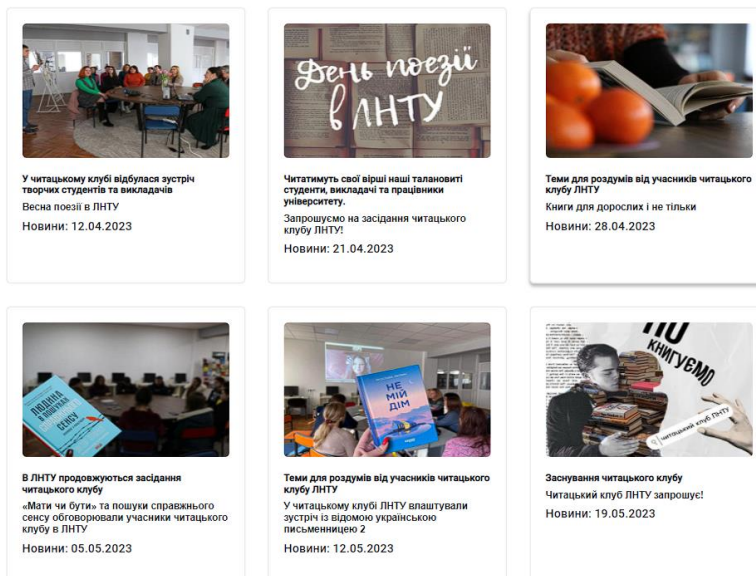
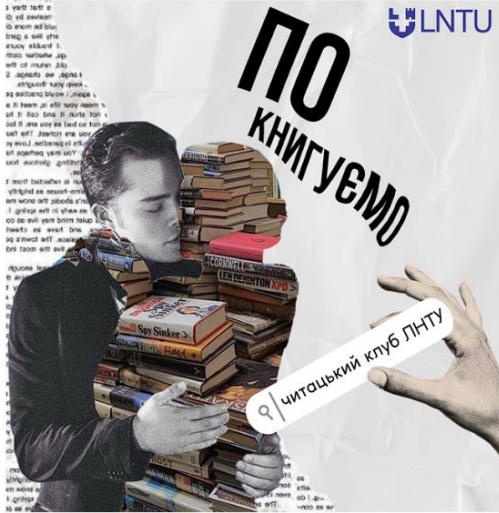


Рисунок 3.9 – Сторінка оголошень

Любиш читати?
 Прагнеш самодосконалення і мрієш змінити світ?
 Чеснею у читацькому клубі ЛНТУ!



Організатори: керівник студентського наукового гуртка «Соціально – історичні виміри суспільства», що діє на кафедрі соціокультурних технологій факультету ЦОСТ та бібліотека ЛНТУ.

Коли: зустрінаємось щомісяця кожного останнього четверга

Локація: бібліотека ЛНТУ

Час зустрічі: 15:00

Рисунок 3.10 – Сторінка оголошення

3.5 Сторінка блогу

Блог в читацькому клубі може бути корисним інструментом для тих, хто любить читати та хоче ділитися своїми думками з іншими. Він може сприяти формуванню спільноти читачів, які обговорюють книги та діляться своїми враженнями. Це також може бути корисно для тих, хто шукає нові книги для читання та хоче дізнатися думку інших читачів про конкретні книги. В даному проєкті, блог реалізовано за допомогою шести блокам (рисунок 3.10).

Недавні пости блогу







 <p>АВТОРСЬКИЙ КУТОЧОК Ексклюзивні інтерв'ю автора з книжковими клубами</p> <p>Ви книголюб і шукаєте наступного читання? Або письменник-початківець, який шукає натхнення та керівництва? Тоді позначте свої календарі, адже на нас чекає</p>	 <p>СПИСКИ ЛІТЕРАТУРИ П'ять невеликих пресів, про які варто знати у 2023 році</p> <p>Під час святкування Місяця малої преси ми хочемо виділити п'ять чудових малих пресів і назв малих прес, які, як ми сподіваємося, ви заглянете!</p>	 <p>ПОРАДИ ТА НАТХНЕННЯ КНИЖКОВОГО КЛУБУ Найкращі ідеї журналу читання для книжкових клубів</p> <p>Ведення щоденника може бути веселим і корисним досвідом для кожного, хто любить читати. Незалежно від того, читаєте ви звичайний чи серйозний книжковий</p>
 <p>ОСНОВНЕ З КНИЖКОВИХ КЛУБУ Bookclubs тепер пропонує преміум- і професійні функції!</p> <p>Bookclubs представляє преміум- та професійну версії нашої платформи організації книжкових клубів із новими функціями, які пропонують покращену організацію, спрощену</p>	 <p>Оновлення додатку та веб-сайту Як ніколи не відставати від читання книжкового клубу</p> <p>Представляємо Shelf Service, нашу нову службу доставки книг за підпискою, яка автоматично надсилає поштою копію вашої останньої книги книжкового клубу,</p>	 <p>СПИСКИ ЛІТЕРАТУРИ Книги, які варто прочитати під час місяця жіночої історії 2023</p> <p>Цього року темою Місяця жіночої історії є «Вшановуємо жінок, які розповідають наші історії». Тож цього місяця ми вшановуємо колишніх і сучасних жінок, які</p>

Рисунок 3.10 – Сторінка блогу

В яких описані назва посту, тема посту і короткий опис. Натиснувши на один з блоків ви перейдете на сторінку де описане даний пост блогу (рисунок 3.11).

Ексклюзивні інтерв'ю автора з книжковими клубами



Ми раді повідомити, що цієї весни Bookclubs організуватимуть захоплюючу серію ексклюзивних інтерв'ю з авторами в Instagram Live! Bookclubs Author Chats — це безкоштовний книжковий клуб із серією інтерв'ю щомісяця, у якому представлені відомі автори та онлайн-обговорення книг. Члени команди Bookclubs щомісяця будуть брати інтерв'ю в одного (ноді двох) талановитих романістів через Instagram Live. Підпишіться на інтерв'ю, а потім поділіться своїми думками на дошці оголошень. Якщо ви пропустите інтерв'ю, не проблема — посилання на записи всіх інтерв'ю можна знайти в примітках до минулої зустрічі.

Чати авторів квітня

У вівторок, 11 квітня 2023 року, об 11:00. Співавторка і генеральний директор ET Bookclubs бере інтерв'ю у підприємця та співробітника Гарвардської школи бізнесу Метта Хігінса про його нову книгу «Спали човни: викиньте план Б за борт і розкрийте свій повний потенціал». У «Спалити човни» Хігінс спирається на досвід власного бізнесу життя та уроки, отримані від тих, хто triumфував у бізнесі, спорті, політиці та мистецтві, щоб запропонувати шаблон для кожного, хто прагне досягнути.

У середу, 19 квітня 2023 року, о 19:00 за сідним часом, ми спілкуємося з відомою журналісткою та письменницею Даніель Фрідман про її найкращу книгу 2023 року в журналі New Yorker «Давайте профікувати», яка розкриває захоплюючу приховану історію сучасних жінок, фітнес-культура, яка в жордавній кінематографічній прові розповідає про те, як фізичні вправи еволюціонували з інструменту краси, який майже виключно представляв як спосіб «зменшення», до інструменту, який мільйони жінок використовували як шлях до психічного, емоційного та фізичного благополуччя. Книжкові клуби можуть насолоджуватися цим неймовірно цікавим і веселим набором Book Club із посібником для обговорення!

Березень авторські чати

Під час Місяця жіночої історії ми вшануємо жінок-оповідачок, які використали свою майстерність, щоб відобразити жіночий досвід. Ми не можемо придумати кращого способу вшанувати жінок-оповідачок, ніж інтерв'ю з письменницею Дайан Марі Браун 23 березня 2023 року о 18:00 за стандартним стандартичним часом через Instagram Live. Браун спілкуватиметься з партнером Bookclubs і Bookstagrammer, Морган Меніс, яка також очолює свій власний публічний книжковий клуб на Bookclubs. Нещодавно виданий роман Браун «Жінки з чорною свічкою» щойно був обраний березневим вибором книжкового клубу Read With Jenna. «Жінки з чорною свічкою» — це тепла й гостра сімейна драма з чарівним поворотом про чотири покоління чорношкірих жінок, сімейне прокляття та один дуже складний рік душевного болю, неправильного спілкування та навчання відпускати.

Лютій авторські чати

Понеділок, 27 лютого 2023 року, о 19:00 Співавторка книжкових клубів Ненсі Браун бере інтерв'ю у ДеШона Чарльза Вінслоу, відомого автора книг «Прийшли люди» та «У Вест-Міллі». Для нового фільму Вінслоу «Порядні люди» розгортається в 1976 році в містечку Вест-Міллі, штат Північна Кароліна, де доктор Меріан Хармон, єдиного чорношкірого лікаря в місті, знайшли застреленою разом із її братом Лавом і сестрою Марвою в їхньому домі. Багато хто підозрює Олімпуса Сеймора, відомого як Ліллі, зведеного брата, який жив по сусідству. Але біла влада в місті, здається, не дуже зацікавлена. І Джо Райт, яка щойно переїхала, має поставити кілька складних запитань і шукати вбивцю.

Рисунок 3.11 – Сторінка блогу

3.6 Технічні вимоги до середовищ використання

Мінімальними вимогами для процесорів, які запускаються в браузер на операційній системі Windows є Internet Explorer 8. Він потребує 1 Ghz процесор; мінімум 4 Gbyte RAM (для Windows 7), рекомендовано 4 Gbyte RAM; мінімум 50 Gbyte (Windows 7), 50 Gbyte (Windows 10 Home) вільного місця на диску.

3.7 Загальний опис роботи веб-застосунку

Технологія використання застосунку полягає в наступному: студент виконує вхід в систему, після чого він бачить головну сторінку, на якій має можливість переглянути найпопулярнішу книгу (за активністю читання книги),

перейти на іншу сторінку, наприклад, «найкращі книги». На цій сторінці є можливість переглянути короткий опис книги, а за бажанням перейти по посиланню і переглянути повний опис книги і, за бажанням, залишити коментар.

ВИСНОВКИ

У даній кваліфікаційній роботі було проведено аналіз існуючих програмних рішень, щоб оцінити доступні опції та вибрати найбільш підходящі з них. Цей етап дав змогу зрозуміти, які функціональність і можливості вже існують на ринку.

Здійснено вибір доступних інструментів для створення проєкту. Залежно від потреб та вимог проєкту були розглянуті різні опції, включаючи мови програмування, фреймворки та інші технології. На основі цього огляду були вибрані найбільш підходящі засоби для реалізації проєкту.

Спроектовано та реалізовано архітектура системи. Цей етап включав створення високорівневого плану системи, визначення основних компонентів, їх взаємодії та структури. Проектування архітектури системи допомогло зорієнтуватися в структурі проєкту та спростити його розробку.

Здійснено реалізацію проєкту. Було розроблено програмне забезпечення з використанням обраних засобів і відповідно до спроектованої архітектури. Розробка включала в себе написання коду, тестування та відлагодження щоб забезпечити належну функціональність та продуктивність системи.

Завершальним етапом процесу було описати розроблений веб-застосункок. Це включало написання документації, де були пояснені основні функції, можливості та використання проєкту. Опис розробленого веб-застосунку є зрозумілою інструкцією для користувачів та інших розробників, які можуть взаємодіяти з системою.

В цілому, виконання всіх цих кроків у процесі розробки проєкту дало змогу досягти успішного створення програмного забезпечення, яке відповідає вимогам та виконує необхідні функції. Після завершення процесу розробки, отриману систему можна використовувати як читацький клуб або літературний портал.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-фреймворк Express (Node.js/JavaScript). MDN Web Docs. URL: https://developer.mozilla.org/ru/docs/Learn/Server-side/Express_Nodejs (дата звернення: 03.03.2023).
2. Введення | axios docs. Axios. URL: <https://axios-http.com/ua/docs/intro> (дата звернення: 05.03.2023).
3. Для чого потрібен webstorm - альтернативна наука Віталія Сема. Альтернативна Наука Віталія Сема. URL: <https://alternativescience.net/programming/204-dlya-chogo-potriben-webstorm/> (дата звернення: 10.03.2023).
4. Переваги веб-розробки на react у 2022. Маркетингове агентство МАВР в Харькове - услуги интернет-маркетинга в Украине. URL: <https://mavr.ua/ua/perevagi-veb-rozrobki-na-react-u-2022/> (дата звернення: 12.03.2023).
5. Учасники проєктів Вікімедіа. Goodreads – вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Goodreads> (дата звернення: 17.03.2023).
6. Учасники проєктів Вікімедіа. WebStorm – вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/WebStorm> (дата звернення: 20.03.2023).
7. Як написати дипломну роботу | На5ку. На5ку - Якісне виконання всіх видів студентських робіт з будь-яких дисциплін. URL: <https://na5ku.com.ua/uk/kak-napisat-diplom/> (дата звернення: 20.03.2023).
8. Bertoli M. React Design Patterns and Best Practices: build easy to scale modular applications using the most powerful components and design patterns. Packt Publishing, 2020. 318 p.
9. Book riot. *BOOK RIOT*. URL: <https://bookriot.com/> (дата звернення: 20.03.2023).
10. Book riot | civilian publications | LIS publications wiki. LIS Publication

Wiki. URL: <https://ischoolwikis.sjsu.edu/lispublications/wiki/civilian-publications/book-riot/> (дата звернення: 22.03.2023).

11. Cooper R. VSCode vs WebStorm. Robert Cooper. URL: <https://robertcooper.me/post/vscode-vs-webstorm> (дата звернення: 25.03.2023).

12. Getting started with React - Learn web development | MDN. MDN Web Docs. URL: https://developer.mozilla.org/enUS/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started (дата звернення: 28.03.2023).

13. Goodreads. *Goodreads*. URL: <https://www.goodreads.com/> (дата звернення: 20.03.2023).

14. Manager. Розробка додатків на REACT -. Разработка мобильных приложений на заказ для Украины, СНГ, Европы - Заказать создание приложения в студии KITAPP. URL: <https://kitapp.pro/uk/rozrobka-dodatktiv-na-react/> (дата звернення: 30.03.2023).

15. How to use CORS in Node.js with Express. Engineering Education (EngEd) Program | Section. URL: <https://www.section.io/engineering-education/how-to-use-cors-in-nodejs-with-express/> (дата звернення: 02.04.2023).

16. Learn to code, change your career. Code Institute. URL: <https://codeinstitute.net/global/blog/advantages-of-javascript/> (дата звернення: 02.04.2023).

17. PostgreSQL vs mysql: the critical differences. Integrate.io. URL: <https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/> (дата звернення: 02.04.2023).

18. PostgreSQL services & support. Cybertec. URL: <https://www.cybertec.com/en/postgresql-overview/advantages-of-postgresql/medium.com/@zwacky/design-a-beautiful-rest-api-901c73489458> (дата звернення: 02.04.2023).

19. Purcell. Javascript master handbook. Premier Press, 2021. 550 p.

20. Poetry book club - the rumpus.net. The Rumpus.net. URL: <https://therumpus.net/book-clubs/> (дата звернення: 20.03.2023).

21. React.js. Створення і розробка інтернет-магазину Харків, Київ,

створення і розробка інтернет магазинів під ключ - Brander. URL: <https://brander.ua/technologies/reactjs> (дата звернення: 02.04.2023).

22. Semah B. What exactly is node.js? Explained for beginners. freeCodeCamp.org. URL: <https://www.freecodecamp.org/news/what-is-node-js/> (дата звернення: 07.04.2023).

23. Sidelnikov G. React. js book: learning react javascript library from scratch. Independently Published, 2020.

24. Simplilearn | online courses - bootcamp & certification platform. Simplilearn.com. URL: <https://www.simplilearn.com> (дата звернення: 10.04.2023).

25. The rumpus.net. The Rumpus.net. URL: <https://therumpus.net/> (дата звернення: 30.05.2023).

ДОДАТКИ

Додаток А

Центральний jsx-код серверної частини

```
require('dotenv').config()
const express = require('express')
const sequelize = require('./db')

const models = require('./models/models')
const cors = require('cors')
const fileUpload = require('express-fileupload')
const router = require('./routes/index')
const app = express()
const errorHandler = require('./middleware/ErrorhandlerMiddleware')
const path = require('path')

const PORT = process.env.PORT || 5000

app.use(cors())
app.use(express.json())
app.use(express.static(path.resolve(__dirname, 'static')))
app.use(fileUpload({}))
app.use('/api', router)

app.use(errorHandler)
const start = async () => {
  try{
    await sequelize.authenticate()
    await sequelize.sync()
    app.listen(PORT, () => console.log(`Server started on port ${PORT}`))
  } catch (e) {
    console.log(e)
  }
}

start()
```

Додаток Б

jsx-код головної сторінки

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css'
import {createContext} from "react";
import App from "./App";
import UserStore from "./store/UserStore";
import BookStore from "./store/BookStore";
import BlogStore from "./store/BlogStore";
import NewsStore from "./store/NewsStore";

console.log(process.env.REACT_APP_API_URL)

export const Context = createContext(null)

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Context.Provider value={{
    user : new UserStore(),
    book : new BookStore(),
    post : new BlogStore(),
    news : new NewsStore(),

  }}>
    <App/>
  </Context.Provider>);
```

Додаток В

jsx-код контролера таблиці book бази даних

```
const uuid = require('uuid')
const path = require('path')
const {Book,BookInfo} = require('../models/models')
const ApiError = require('../error/ApiError')
class BookController {
  async create(req, res,next){
    try{
      let {name, rating, authorId, summary, genreId, info } = req.body
      const {img} = req.files
      let fileName = uuid.v4() + ".jpg"
      img.mv(path.resolve(__dirname, "..", 'static', fileName))

      const book = await Book.create({name, rating,summary, authorId,
genreId, img: fileName })

      if (info){
        info = JSON.parse(info)
        info.forEach(i =>
          BookInfo.create({
            title:i.title,
            description: i.description,
            bookId:book.id

          )))
      }

      return res.json(book)
    } catch (e) {
      next(ApiError.badRequest(e.message))
    }
  }

  async getAll(req, res){
    let {authorId, genreId, limit, page} = req.query
    page = page || 1
    limit = limit || 9
    let offset = page * limit - limit
    let books;
```

```

    if (!authorId && !genreId)

        books = await Book.findAndCountAll({limit, offset})
    }
    if(authorId && !genreId){
        books = await Book.findAndCountAll({where:{authorId}, limit,
offset})
    }
    if(!authorId && genreId){
        books = await Book.findAndCountAll({where:{genreId},limit, offset})
    }
    if (authorId && genreId){
        books = await Book.findAndCountAll({where:{genreId, authorId},
limit, offset})
    }
    return res.json(books)

}

async getOne(req, res){
    const {id} = req.params
    const book = await Book.findOne(
        {
            where: {id},
            include: [{model:BookInfo, as: 'info'}]
        },
    )
    return res.json(book)
}

}

module.exports = new BookControl()

```