DESIGN DOCUMENTATION

Subject : Full Stack Application Development SEZG503

Professor : Mrs. Akshaya Ganesh

Assignment on BOOK EXCHANGE PLATFORM

Problem statement :

Book lovers frequently accumulate a collection of books they have read and look for other recommendations. They are always eager to explore new reading material. Traditional methods of exchanging books, such as local book swaps or lending among friends, are limited in scope and accessibility. Therefore, it is imperative to have a digital platform that can facilitate book exchanges on a larger scale. This platform should connect users with similar reading interests, enabling them to tradebooks easily and efficiently. This project aims to develop a full-stack web application that serves as a centralized platform for users to exchange, lend, and borrow books with other users. The platform should provide a user-friendly interface, robust search and recommendation features, and secure transaction capabilities.

Full-stack web applications are a combination of frontend and backend technologies, collectively forming a stack that powers the entire application. This technology stack encompasses various tools, frameworks, and languages, each serving a specific purpose within the development ecosystem.

**Technology Related to Full Stack Development**

**Front-end Development**

It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the frontend portion of the web application or website.

**Front-end Technologies**

The frontend portion is built by using some languages which are discussed below:

* **HTML:** HTML stands for Hyper Text Markup Language. It is used to design the frontend portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within tag which defines the structure of web pages.
* **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
* **JavaScript:** JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

**Front End Libraries and Frameworks**

* **React.js:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.
* **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.
* **jQuery:** jQuery is an opensource JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.  
  Some other libraries and frameworks are:  Semantic-UI, Foundation, Materialize, Backbone.js, Ember.js etc.
* **Back-end Technologies**
* It refers to the server-side development of web application or website with a primary focus on how the website works. It is responsible for managing the database through queries and APIs by client-side commands. This type of website mainly consists of three parts front end, back end, and database. The backend portion is built by using some libraries, frameworks, and languages which are discussed below:
* **Java:** Java is one of the most popular and widely used programming language and platform. It is highly scalable. Java components are easily available.
* **Python:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
* **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it’s not a programming language. Most of the people are confused and understand it’s a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It’s used in production by large companies such as Paypal, Uber, Netflix, Walmart and so on.
* **Back End Frameworks:** The list of backend frameworks are: Express, Django, Rails, Laravel, Spring etc.
* The other backend program/scripting languages are: C , Ruby, REST, GO etc.
* Security of data is important.

**Database:** Database is the collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

* **Oracle:** Oracle database is the collection of data which is treated as a unit. The purpose of this database is to store and retrieve information related to the query. It is a database server and used to manages information.
* **MongoDB:** MongoDB, the most popular NoSQL database, is an opensource document-oriented database. The term ‘NoSQL’ means ‘non-relational’. It means that MongoDB isn’t based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data.
* **SQL:** Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

To develop this full-stack web application for the book exchange platform, let’s break down the core components and outline the tech stack, application architecture, and key features to focus on for the prototype.

Stack contains

Frontend : React.js with Material-UI for a responsive, user-friendly interface

**React.js:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based frontend library responsible only for the view layer of the application. It is maintained by Facebook.

- Backend : Node.js with Express for RESTful API endpoints

* **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it’s not a programming language. Most of the people are confused and understand it’s a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It’s used in production by large companies such as Paypal, Uber, Netflix, Walmart and so on.

- Database : MongoDB for storing book listings, user profiles, and transactions.

Database is the collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

- Authentication : JSON Web Tokens (JWT) for secure session management

- Real-time Messaging : WebSockets or a library like Socket.io for user messaging

- Recommendation System : Optional - A basic recommendation model using collaborative filtering or genre-based matching.

2. Application Architecture

- Frontend-Backend Separation : A React frontend making API calls to the backend, ensuring modularity and separation of concerns.

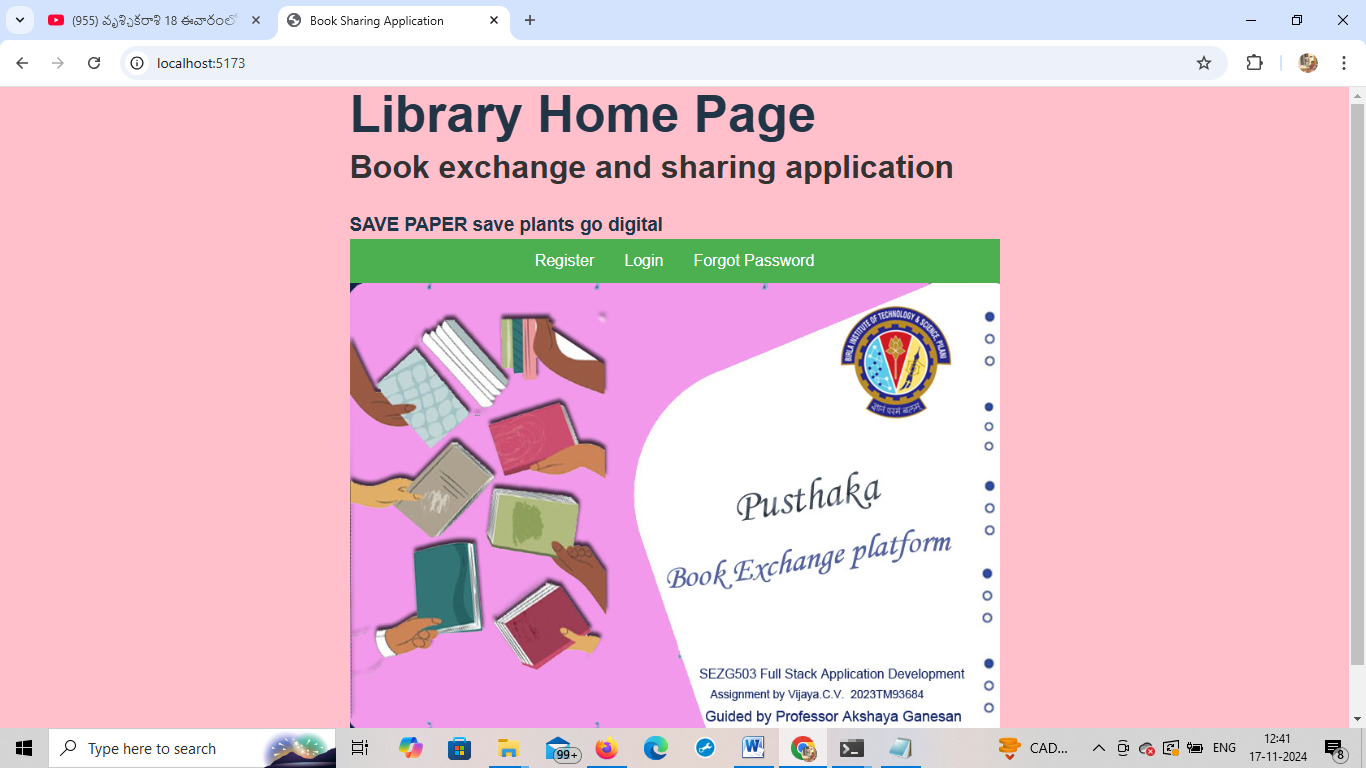
- API Layer : RESTful APIs to handle CRUD operations on user profiles, book listings, search functionalities, exchange requests, and messaging.

- Database Schema : MongoDB collections for `Users`, `Books`, `Requests`, and `Transactions`.

- Real-Time Updates : Using WebSockets to enable instant messaging between users and notifications for transaction status changes.

- Middleware : Authentication and role-based access control middleware for user operations.

The first screen of my assignment



Three User Stories and Acceptance Criteria for the Book Exchange Platform

User Story 1: User Authentication

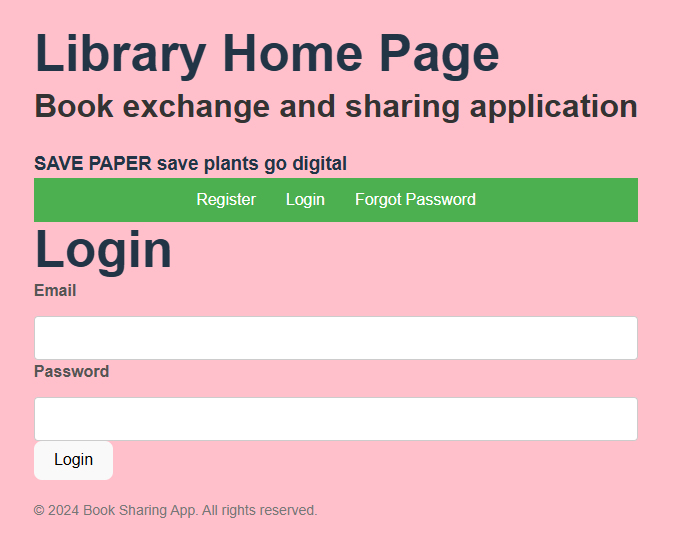
- Endpoints :

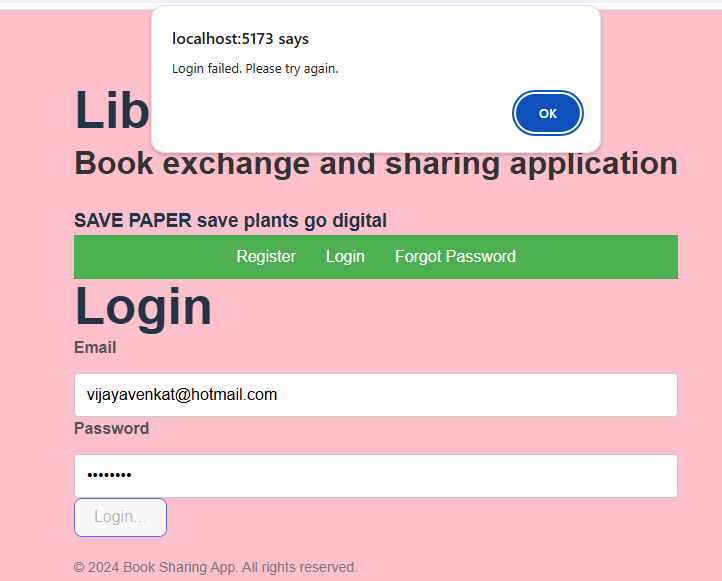
- `POST /api/auth/register`: Registers a new user with email, password (hashed).

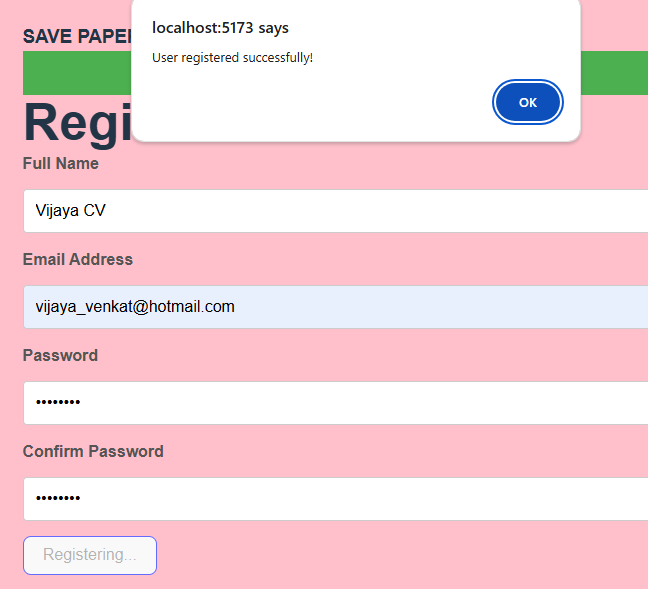
- `POST /api/auth/login`: Authenticates users and provides a JWT token.

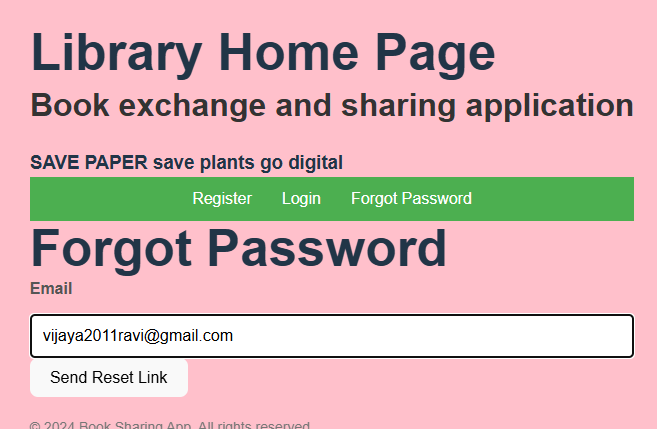
- `POST /api/auth/reset-password`: Handles password reset requests.

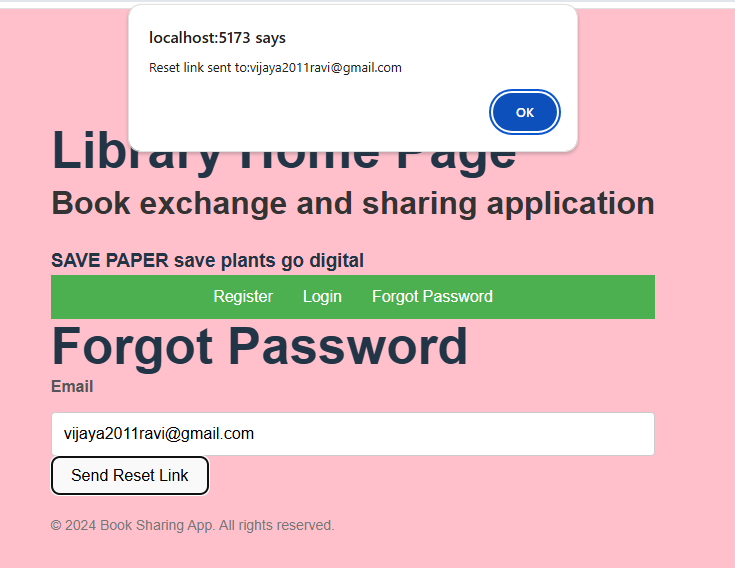
- Frontend : A registration and login form with client-side validation.











User Story 2: Book Listing

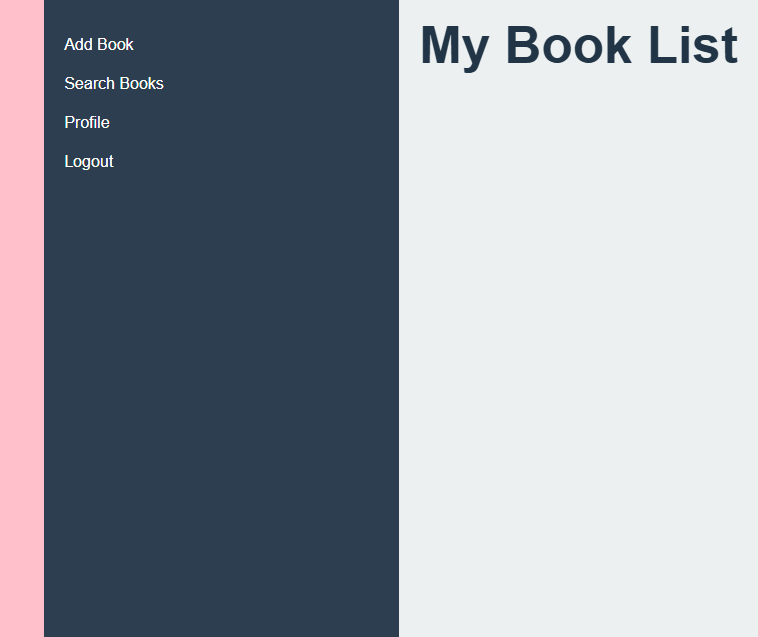
- Endpoints :

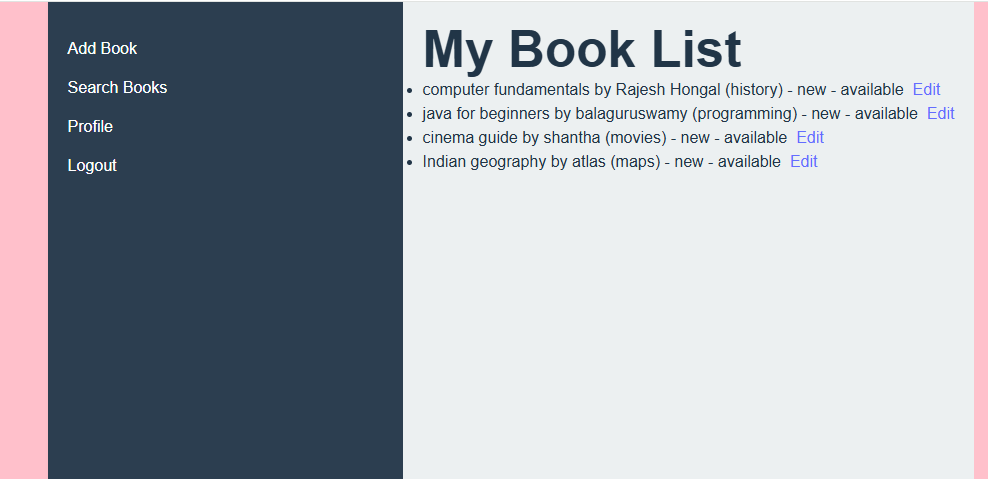
- `POST /api/books`: Allows users to add a book listing.

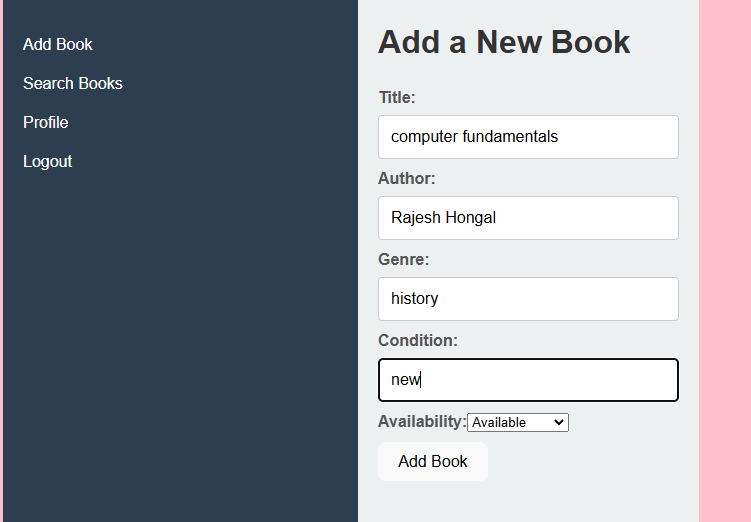
- `GET /api/users/:userId/books`: Fetches books listed by a specific user.

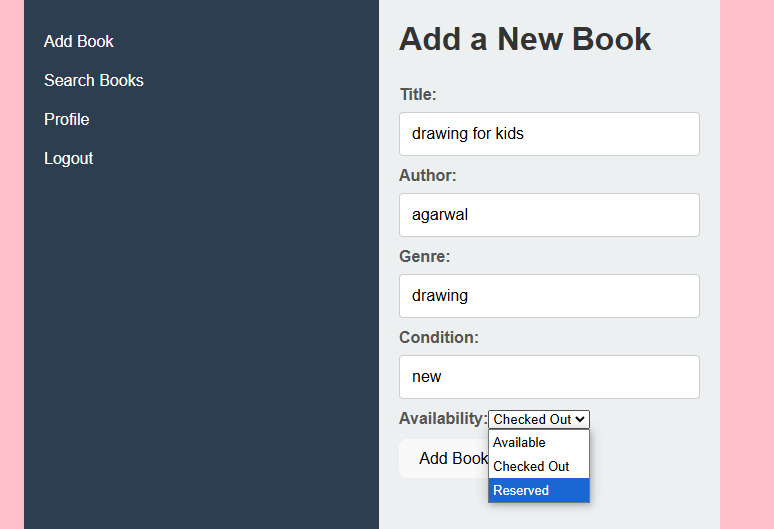
- `PUT /api/books/:bookId`: Allows users to update or delete their book listings.

- Frontend : Form for book details (title, author, genre, etc.) with a list display on the profile.





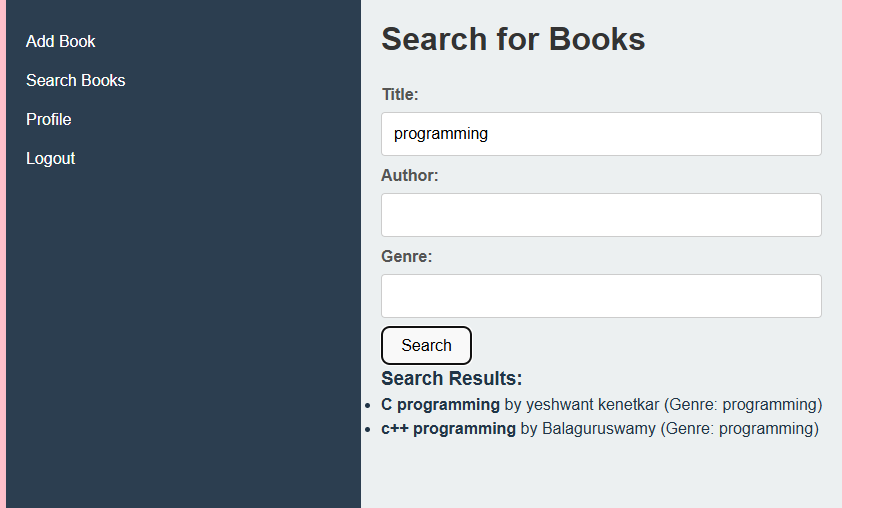




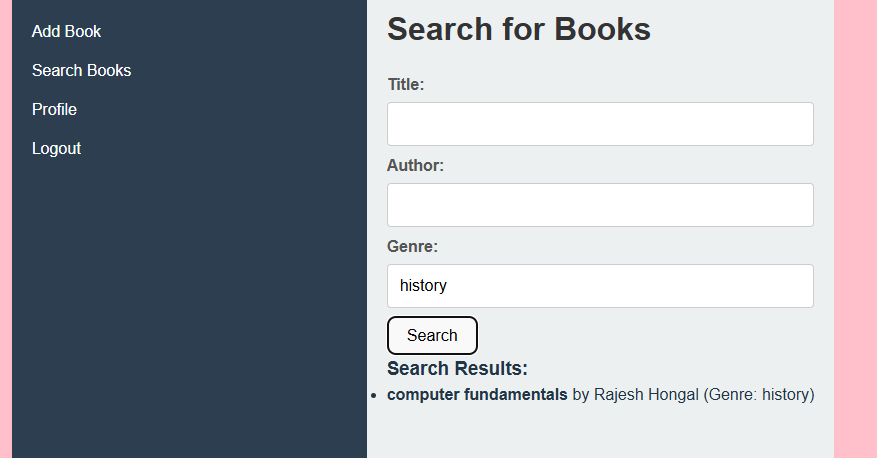
User Story 3: Book Search

- Endpoints :

- `GET /api/books/search`: Allows users to search and filter books by title, author, genre, etc. - Frontend : Search bar and filter options; paginated results for improved user experience.







User Flow

- User Registers/Login →

Once the user logs in the next navigation will move to

Books page. Add / Renewal of books Book Listings → Searches for Books →

This approach provides a structured, modular design that balances the key features for a book exchange platform prototype. The focus is on a seamless user experience and ensuring core functionality is robust for future scaling.

Sqlite database

# DB Browser for SQLite V3.13.1

DB Browser for SQLite (DB4S) is a high quality, visual, [open source](https://github.com/sqlitebrowser/sqlitebrowser) tool designed for people who want to create, search, and edit [SQLite](https://www.sqlite.org/) or [SQLCipher](https://www.zetetic.net/sqlcipher/) database files. DB4S gives a familiar spreadsheet-like interface on the database in addition to providing a full SQL query facility. It works with [Windows](https://sqlitebrowser.org/dl#windows), [macOS](https://sqlitebrowser.org/dl" \l "macos), and most versions of [Linux](https://sqlitebrowser.org/dl#linux) and [Unix](https://sqlitebrowser.org/dl#freebsd).

React V18.16.1

REACT

**React** is the library for web and native user interfaces. Build user interfaces out of individual pieces called components written in JavaScript.

React apps are made out of components. A component is a piece of the UI (user interface) that has its own logic and appearance. A component can be as small as a button, or as large as an entire page.

React is a JavaScript library used for building user interfaces, particularly for single-page applications where dynamic, real-time user experiences are needed. It was developed by Facebook (now Meta) and is widely used in web development for creating interactive and reusable UI components.

Key Features of React:

1. Component-Based Architecture

React divides the UI into independent, reusable pieces called components . Each component can manage its own state and can be composed to build complex UIs.

2. Declarative Syntax

React allows developers to describe what the UI should look like for different states, and React updates and renders the UI accordingly when the state changes.

3. Virtual DOM

React uses a lightweight copy of the actual DOM called the Virtual DOM . This allows React to efficiently determine which parts of the DOM need updating, making updates fast and improving performance.

4. State and Props Management

- State : Stores dynamic data within a component.

- Props : Passes data between components to enable communication.

5. JSX (JavaScript XML)

JSX is a syntax extension that allows developers to write HTML-like code in JavaScript. It makes writing React components more intuitive.

```jsx

function HelloWorld() {

return <h1>Hello, World!</h1>;

}

```6. React Ecosystem

While React itself focuses on the UI, it integrates well with other libraries and frameworks for tasks like state management (e.g., Redux, Context API) and routing (e.g., React Router).

7. Rich Developer Tooling

Tools like the React Developer Tools extension for browsers help developers inspect components, state, and props in real time.

Use Cases

- Building dynamic, high-performance web applications.

- Mobile app development using React Native.

- Dashboards, data visualization tools, and SaaS platforms.

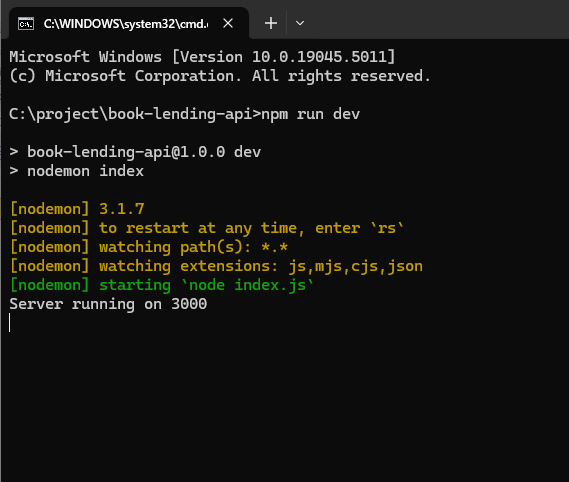
React's simplicity, performance, and flexibility make it a popular choice for developers across a range of industries.

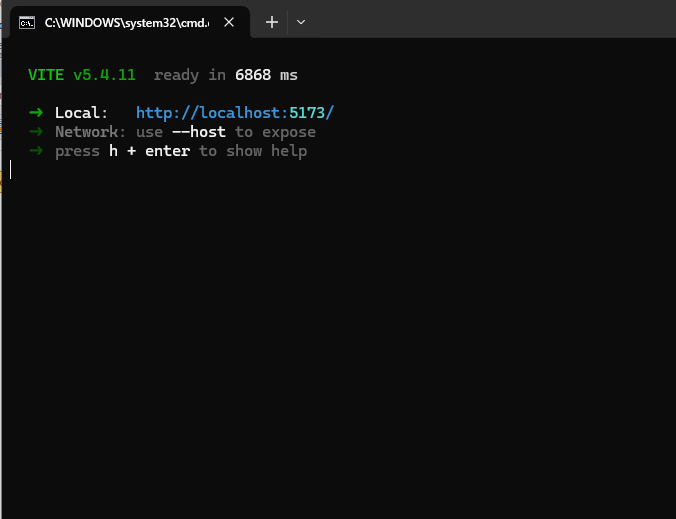
Node js / koa .js frame work

Koa is a new web framework designed by the team behind Express, which aims to be a smaller, more expressive, and more robust foundation for web applications and APIs. By leveraging async functions, Koa allows you to ditch callbacks and greatly increase error-handling. Koa does not bundle any middleware within its core, and it provides an elegant suite of methods that make writing servers fast and enjoyable.

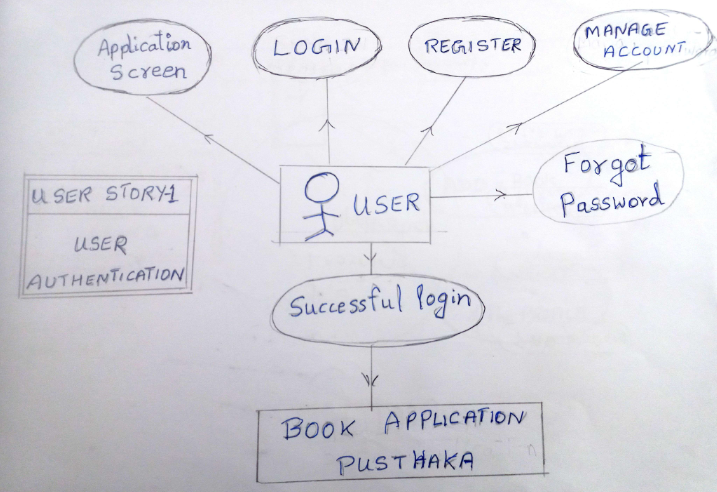
**Terminal 1.21**

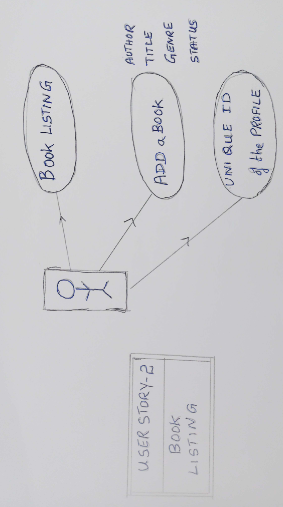
Windows Terminal is back with another preview release! Windows Terminal Preview 1.21 introduces long-awaited features like Buffer Restore and fontfall back as well as new experimental features like Scratchpad and the ability to load up an image as a texture.

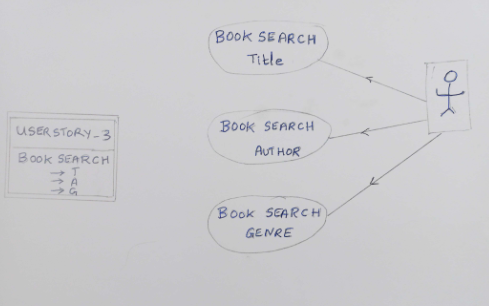




Architectural / design flow diagram.







The above mentioned assignment of problem statement BOOK EXCHANGE PLATFORM is been successfully completed by the guidance of our Professor Mrs. Akshaya Ganesan. Thankyou madam for the opportunity given to learn new technologies like front end, back end and database along with Githhub.

It was an immense pleasure and discovery idea for me during the completion of my assignment