**Full Stack Development with MERN**

**Project Documentation format**

**1. Introduction**

**Project** **Title**: Bookstore

**Team** **Members**: Rahul R (Full Stack Developer)

The Bookstore project is a web‑based application designed to bring the book‑shopping experience into a smooth digital format. Users can discover new titles, search based on their interests, read reviews, and purchase books—all from one platform. The aim is to make book buying easy, fast, and enjoyable.

**2. Project Overview**

1. **Purpose:** To build a convenient online system where book lovers can browse, evaluate, and purchase books without the limitations of physical stores. The platform focuses on usability, availability of titles, and reliable decision‑making features such as reviews and search filters.
2. **Key Features:**

* Secure user registration and login
* Browse and search books by title, author, category, and genre
* Wishlist and cart to manage selected books
* User reviews and ratings to help decision making
* Admin dashboard for managing inventory and users

**3. Architecture**

• **Frontend:** Implemented using React, structured with reusable components. Context API is used to handle global state, particularly for authentication, wishlist, and cart. React Router ensures smooth page navigation.

• **Backend:** Developed using Node.js and Express.js, organized using the MVC architecture. Routes handle incoming requests, controllers manage logic, and models interact with the database

• **Database:** MongoDB stores users, books, reviews, and order data. Mongoose schemas define the data structure and maintain relationships between records.

**4. Setup Instructions**

• **Prerequisites:** Node.js, MongoDB, Git, VS Code.

• **Installation Overview:**

Clone the project repository

Install dependencies for both frontend and backend

Configure environment variables (e.g., MongoDB URI)

Start the development servers

**5. Folder Structure**

• **Client:** Consists of all React‑based UI features — pages, components, CSS files, assets, and context providers.

• **Server:** Includes routes, controllers, models, and middleware that handle backend logic and database interaction.

**6. Running the Application**

To run the project locally:

o **Frontend:** npm start in the client directory.

o **Backend:** npm start in the server directory.

**7. API Documentation**

• The backend exposes several REST APIs for user, book, wishlist, cart, and admin operations.

Endpoint Method Purpose

/api/auth/register POST Register a new user

/api/auth/login POST Authenticate user

/api/books GET Fetch all books

/api/books/:id GET Get book details

/api/books POST Add new book (Admin)

/api/books/:id PUT Update book (Admin)

/api/books/:id DELETE Delete book (Admin)

/api/cart POST Add book to cart

/api/wishlist POST Add book to wishlist

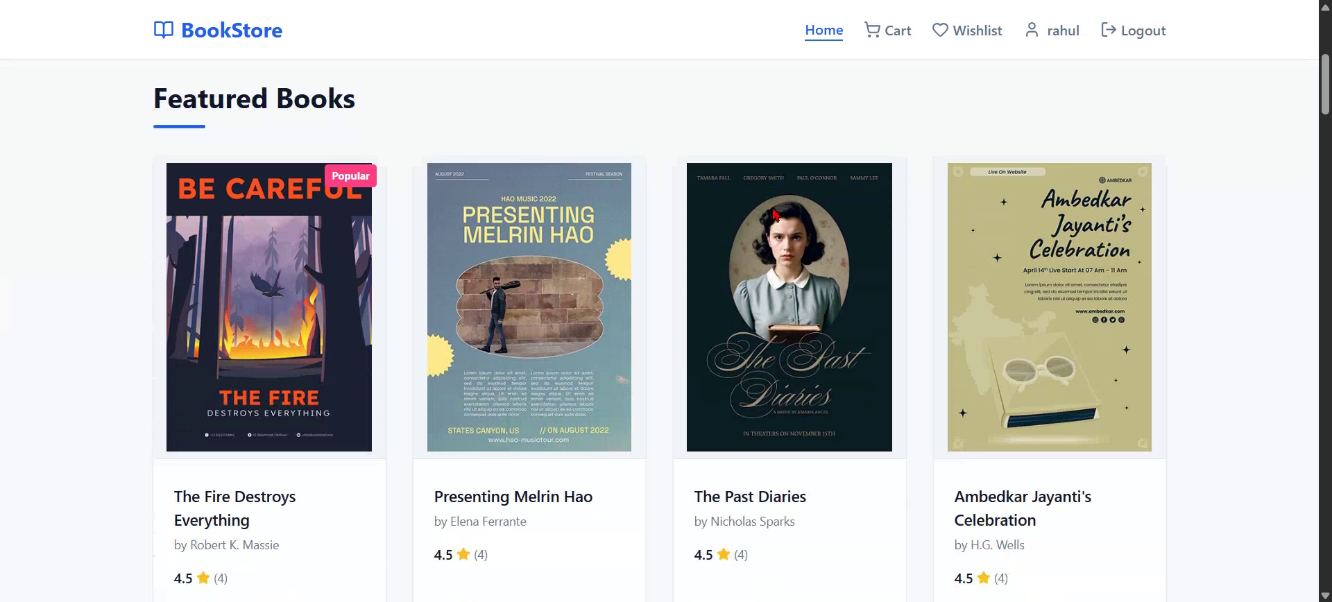
**8. Authentication**

• Authentication is handled securely on the backend. Login credentials are verified before providing access, and users with admin privileges receive elevated access rights. Based on the logged‑in role, users get different permissions for the platform.

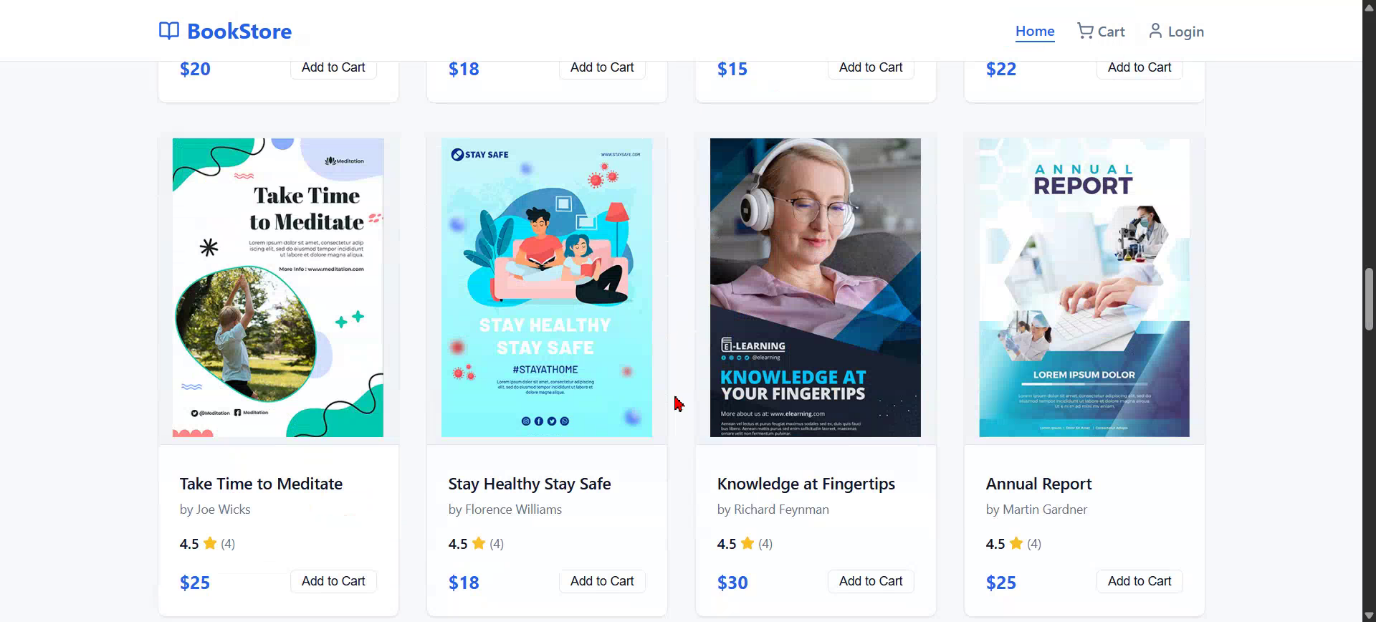
**9. User Interface**

• This section contains visual proof of functionality. Screenshots should be included for:

Homepage:



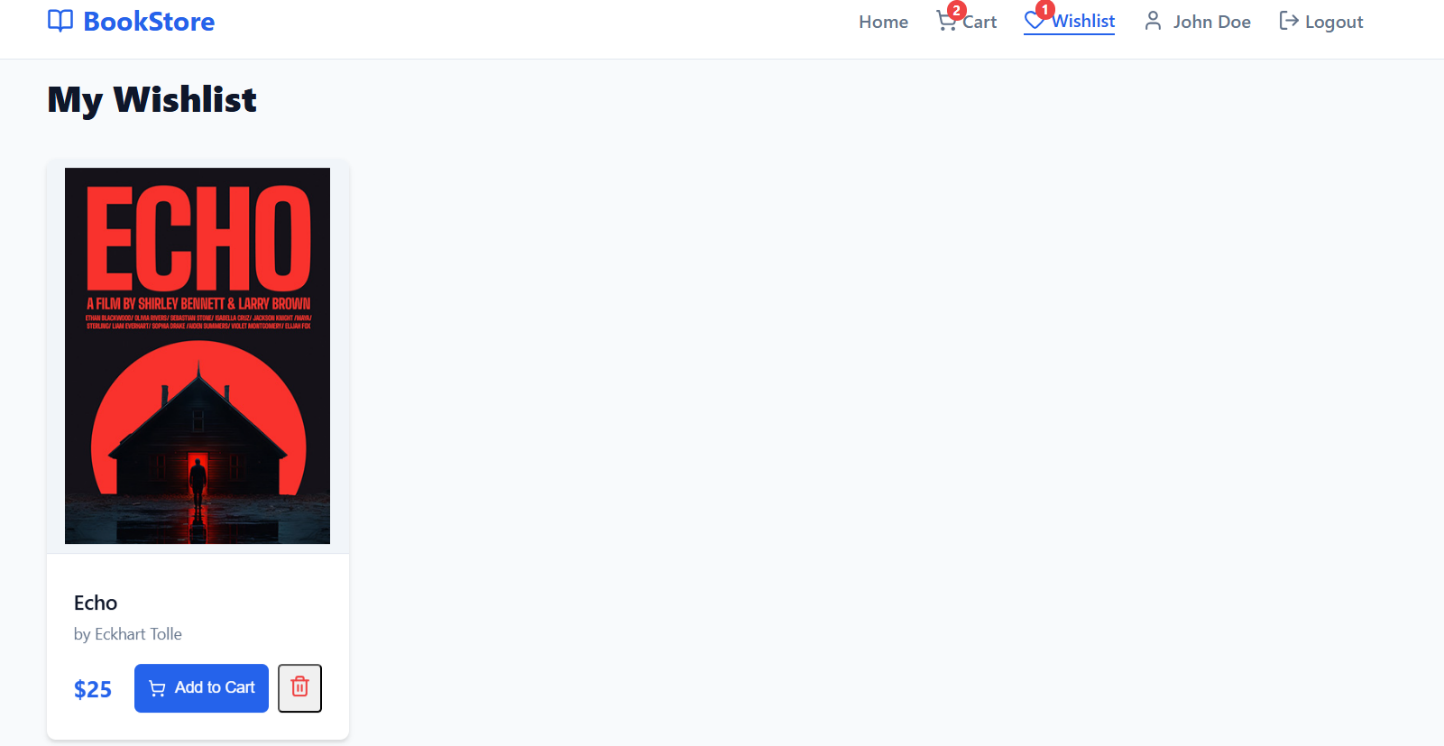
Book list:



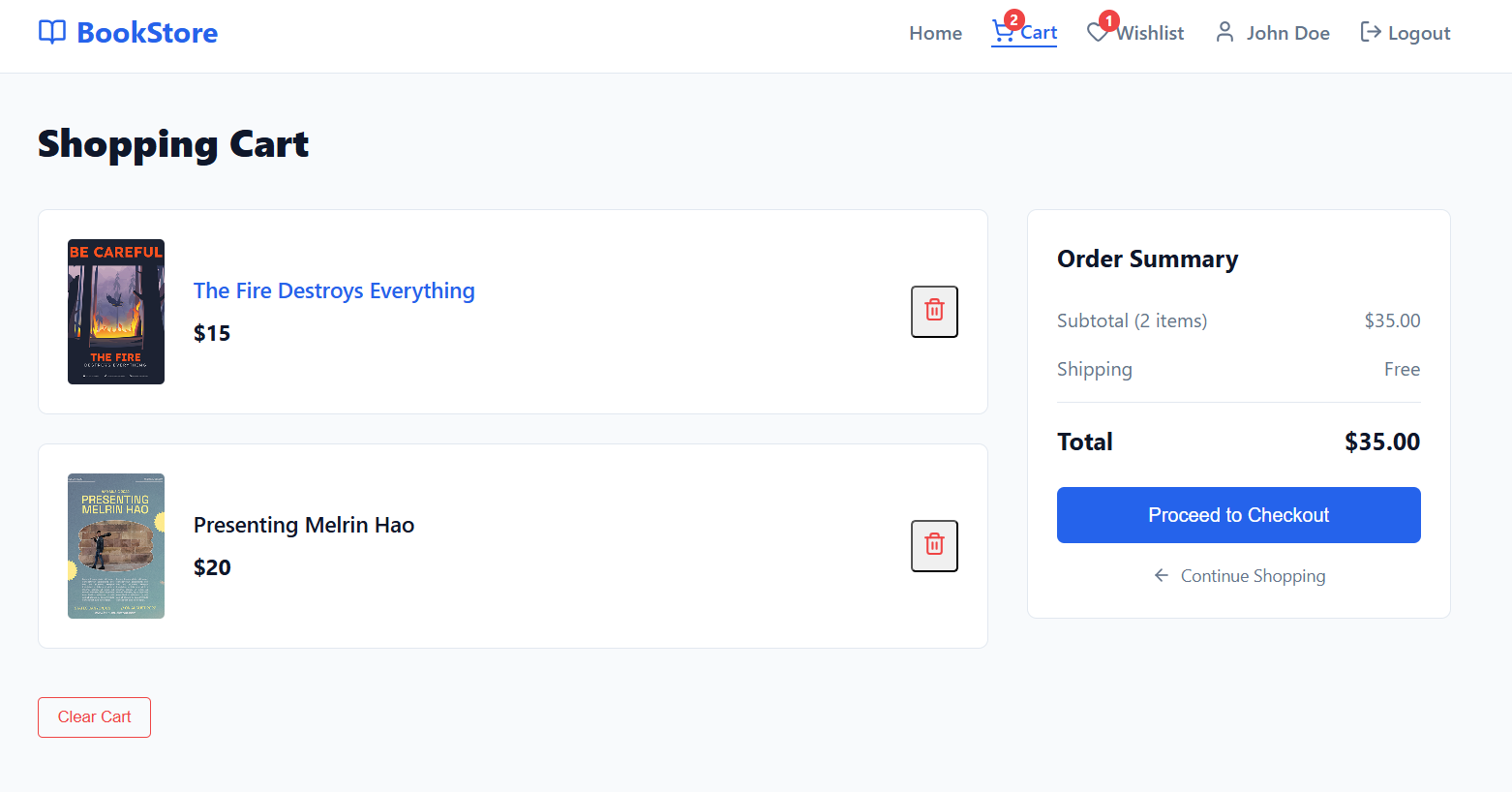
Book details:



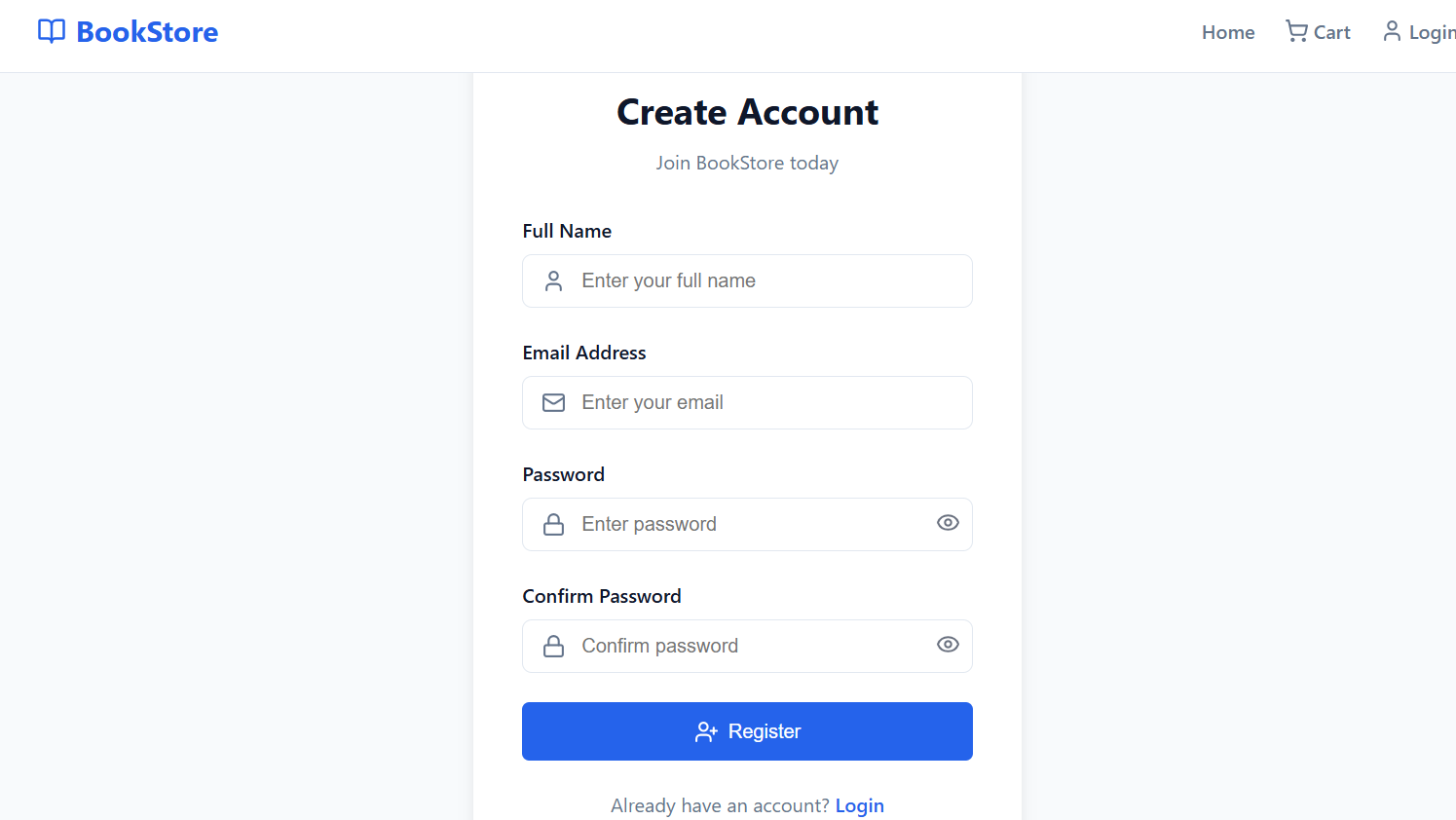
Wishlist:



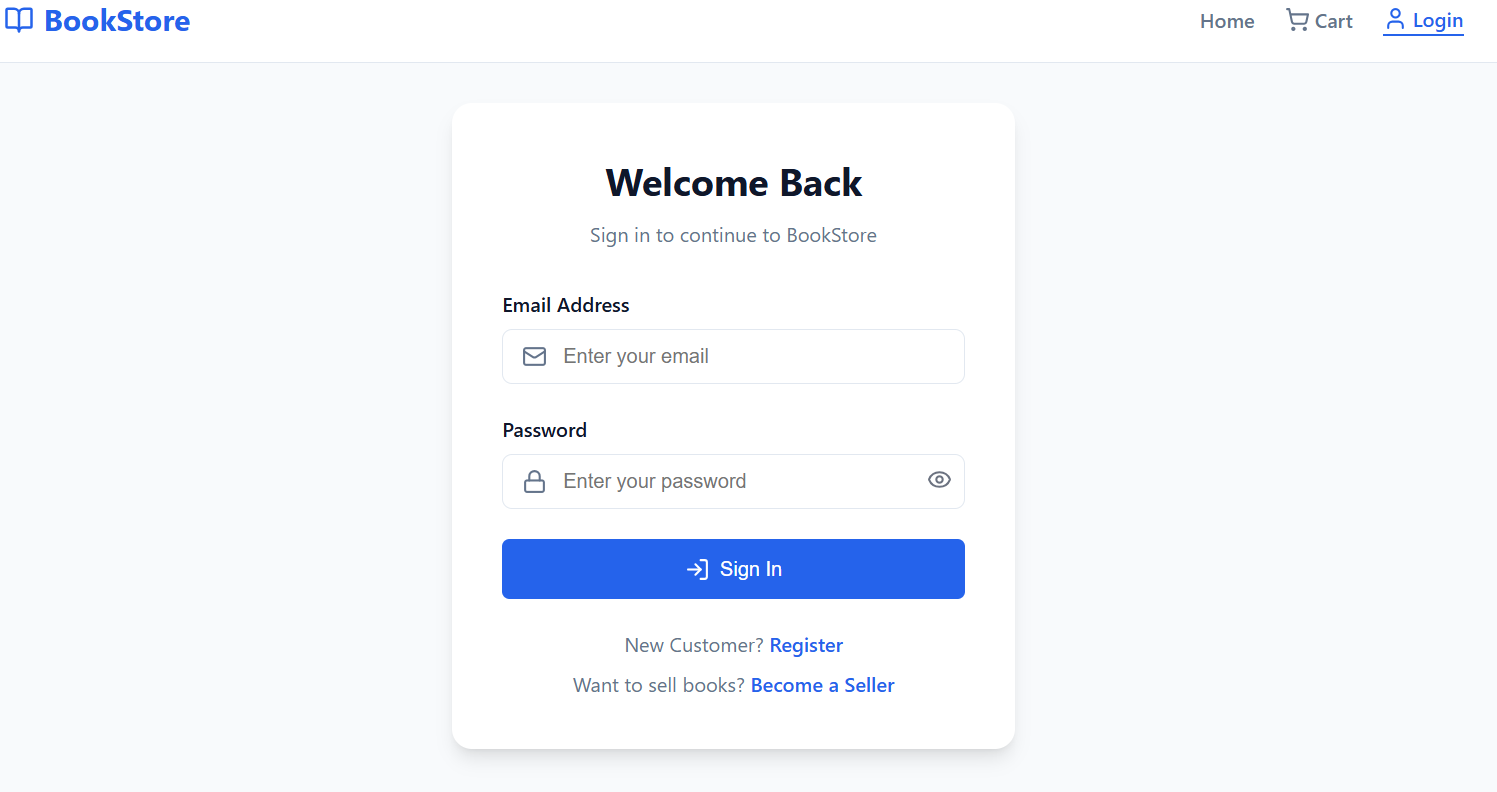
Cart:



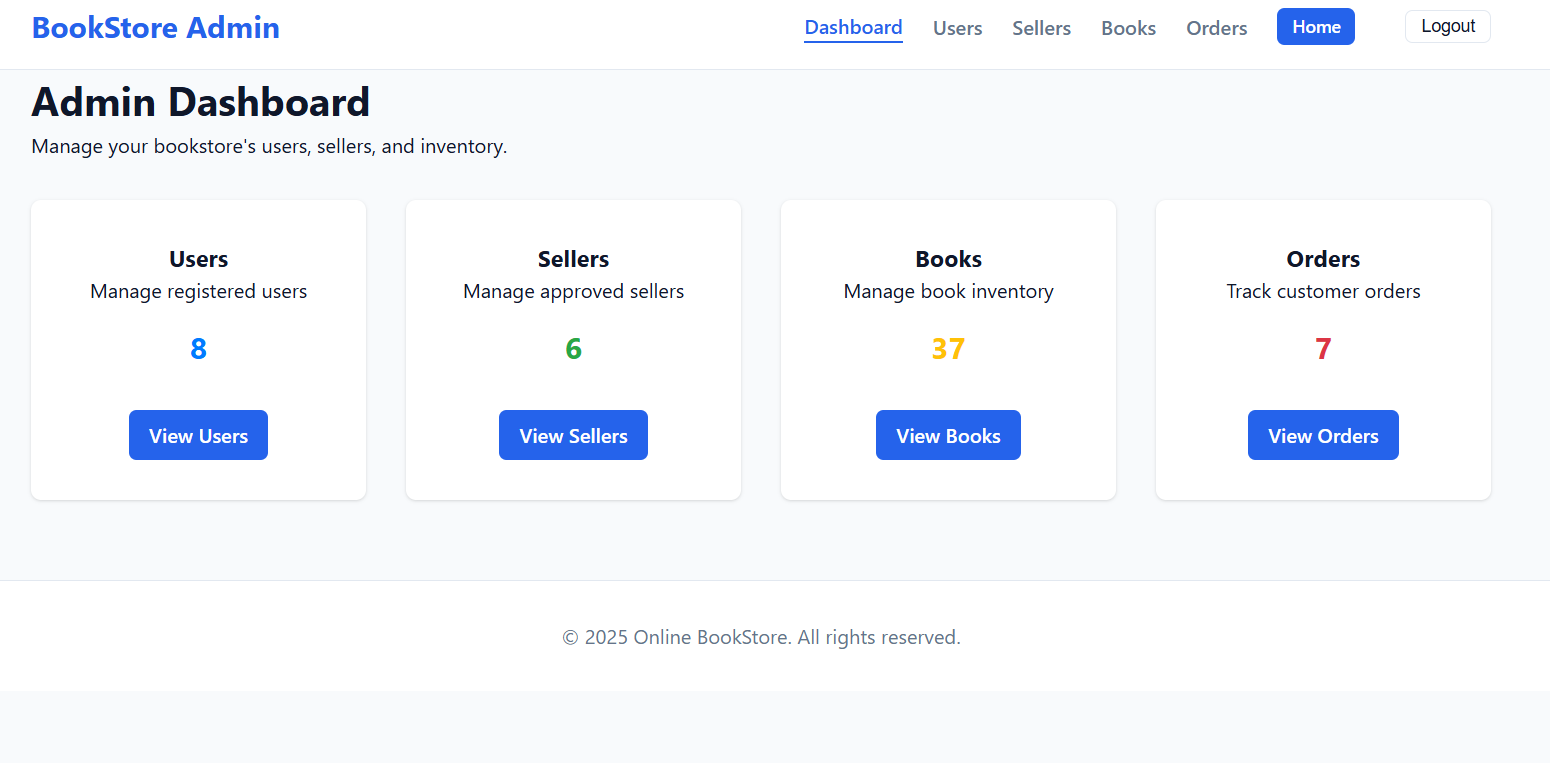
Register:



Login:



Admin section:



**10. Testing**

Testing was performed using:

* Manual UI testing for all user actions
* Postman to test REST API endpoints
* Debugging of key features such as login, wishlist, cart, and admin CRUD

**11. Screenshots or Demo**

• The application is fully functional and has been tested with real‑time usage. Screenshots of the working system have been captured and attached in the documentation, showing:

* Homepage
* Book browsing & search
* Wishlist & Cart
* Login & Register process
* Payment simulation (dummy payment)
* Admin dashboard

Demo Link:

[https://drive.google.com/file/d/1xKOBktpVP78RZ-CT6Fy0FxmWyMu3BITy/view?usp=drive\_link](https://drive.google.com/file/d/1Hz4NXjjuJEEBs5vVNIqnfl7vgjKjp6XI/view?usp=drive_link)

**12. Known Issues**

| **Issue** | **Description** |
| --- | --- |
| Payment | A dummy payment method is implemented; real payment gateway integration is not available yet |
| Seller Catalog Limitation | Sellers cannot add already existing books into their own catalog |
| Recommendation System | Personalized book recommendations are currently not implemented |

.

**13. Future Enhancements**

Future improvements that can make the platform more scalable and user‑friendly:

| **Enhancement** | **Purpose** |
| --- | --- |
| Real Payment Gateway (Razorpay / Stripe etc.) | Enable secure online transactions |
| Order Tracking System | Allow customers to track their book delivery status |
| Advanced Recommendation Engine | Suggest books based on user preferences and reading history |
| Multi‑Seller Catalog Sync | Allow sellers to add existing books to their catalog easily |
| Follow Authors / Notification System | Notify users when a favorite author releases new books |
| Mobile App (Android/iOS) | Expand platform accessibility across devices |