Full Stack Development with MERN:

Project Documentation

---

1. Introduction

- **Project Title :** BookNest – Where Stories Nestle
- **Team Id :**LTVIP2025TMID55365
- **Team Members:**

- **Pattapu Chohitha** – **Team Leader**
  Led the project from ideation to deployment. Managed planning, development, and integration of all modules. Took primary responsibility for the frontend, backend, database, and documentation.
- **Pantula Vindhya Kanaka Sai Pavani** – **Team Member, Frontend Developer (Support Role)**
  Assisted in designing UI components and maintaining visual consistency.
- **Panguluri Vishnu Vardahn** – **Team Member, Backend Developer (Support Role)**
  Provided support in building REST APIs and implementing user authentication logic.
- **Pamarthi Sri Lakshmi** – **Team Member, Database Developer (Support Role)**
  Contributed to initial schema discussions and data modeling.

---

2. Project Overview

- **Purpose:**
  The primary goal of BookNest is to create a seamless, user-friendly online bookstore that caters to modern readers and local sellers. It aims to bridge the gap between traditional book shopping and digital convenience by offering personalized browsing, role-based dashboards, and efficient order management. This project not only simplifies the book-buying experience for readers like Sarah (our scenario user) but also empowers sellers to manage their inventory

efficiently. To provide a seamless and immersive digital bookstore experience where users can explore, search, and buy books, and sellers/admins can manage book listings and analytics.

- ## Key Features:

- **User Registration and Authentication:** Allow users to register accounts securely, log in, and authenticate their identity to access the book store platform.

- **Book Listings:** Display a comprehensive list of available books with details such as title, author, genre, description, price, and availability status.

- **Book Selection:** Provide users with options to select their preferred books based on factors like genre, author, ratings, and popularity.

- **Purchase Process:** Allow users to add books to their cart, specify quantities, and complete purchases securely. Upon successful completion, an order is generated, and the inventory is updated accordingly.
- **Order Confirmation:** Provide users with a confirmation page or notification containing details of their order, including book information, total price, and order ID.

- **Order History:** Allow users to view their past and current orders, providing options to track shipments, review purchased books, and rate their shopping experience.

- **Organizer Dashboard:** Offer administrators an interface to manage book listings, inventory levels, user accounts, orders, and other platform-related activities.

- **Create Item:** Organizer can create items and add new items and he can get the items and he can update items.

- **Admin Dashboard:** Offer administrators an interface to manage book listings, inventory levels, user accounts, orders, and other platform-related activities. Manage the users and organizers.
- **Reporting and Analytics:** Generate reports and analytics on book sales, popular genres, user demographics, and other relevant metrics to gain insights into platform usage and performance.

- **Integration with External APIs:** Integrate with third-party APIs for services like payment processing, shipping logistics, and book recommendations to enhance the functionality and user experience of the book store platform.

---

3. Architecture

- **Frontend (React):**
    - Built with React functional components and React Router.
    - Styled with Tailwind CSS/Bootstrap.
    - Responsive design for all screen sizes.
- **Backend (Node.js + Express):**
    - RESTful API endpoints.
    - Modular architecture separating controllers, routes, and middleware.
    - Integrated JWT for authentication.
- **Database (MongoDB):**
    - Schemas for Users, Books, Orders, and Admin roles.
    - Mongoose ODM for model interaction.
    - Efficient querying and schema relationships.

---

4. Setup Instructions

- **Prerequisites:**
    - Node.js (v18+)
    - MongoDB Atlas/local instance
    - Git
- **Installation:**

1. Clone the repo:

   git clone [your-repo-url]

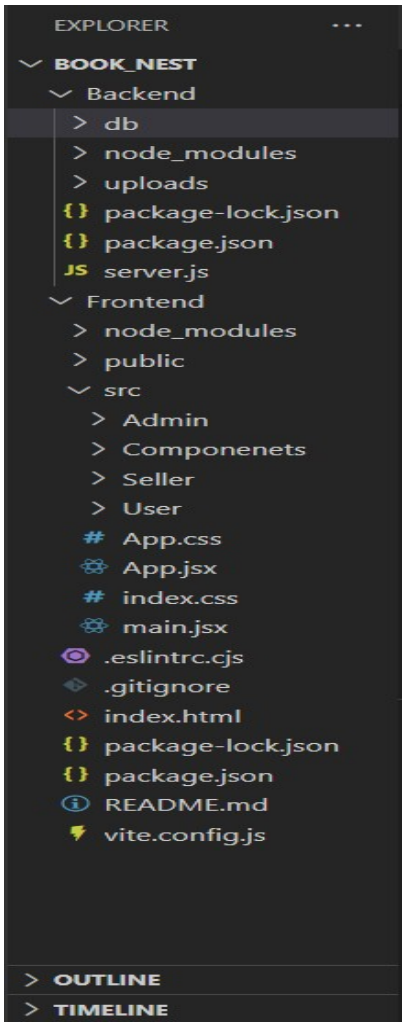
2. Navigate into directories:

   cd client
   npm install
   cd ../server
   npm install

3. Add .env file with Mongo URI and JWT_SECRET in server folder.

---

5. Folder Structure



- **Client (React Frontend):**
    - src/components: Reusable UI components
    - src/pages: Route-based pages (Home, Login, etc.)
    - src/App.js: Root component with routing logic
- **Server (Node Backend):**
    - routes/: Route definitions
    - controllers/: Business logic
    - models/: Mongoose schemas
    - middleware/: JWT verification and error handlers

---

6. Running the Application

- **Frontend:**

  cd client
  npm start

- **Backend:**

  cd server
  npm start

---

7. API Documentation

Endpoint	Method	Description
/api/register	POST	Register new user
/api/login	POST	Authenticate user
/api/books	GET	Fetch all books
/api/cart	POST	Add book to cart
/api/orders	POST	Place order
/api/admin/books	CRUD	Admin book management

Note: Full request/response examples in Postman collection.

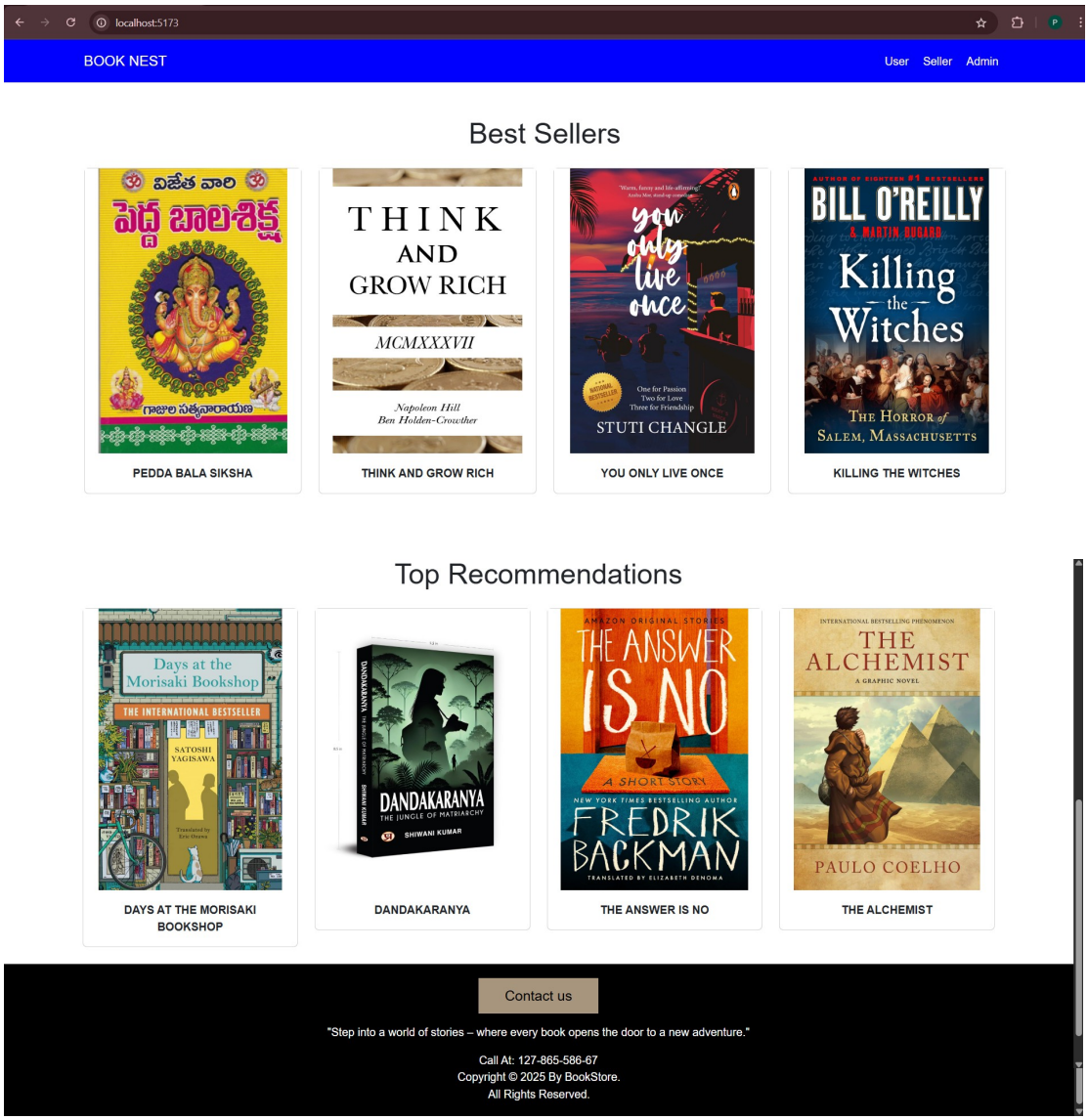
8. Authentication

- Uses **JWT tokens** stored in local storage.
- Backend middleware validates tokens on protected routes.
- Separate login access for users, sellers, and admin roles.

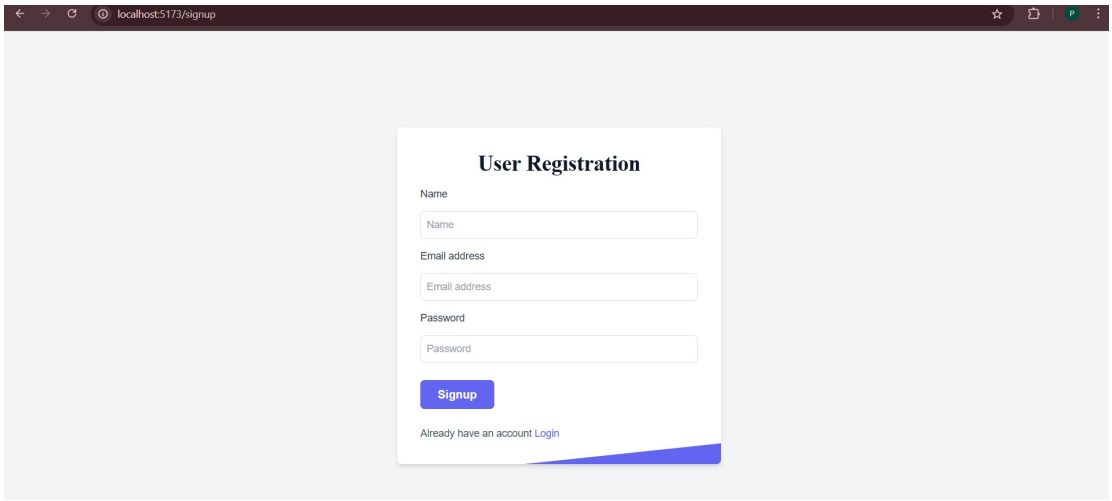
---

9. User Interface

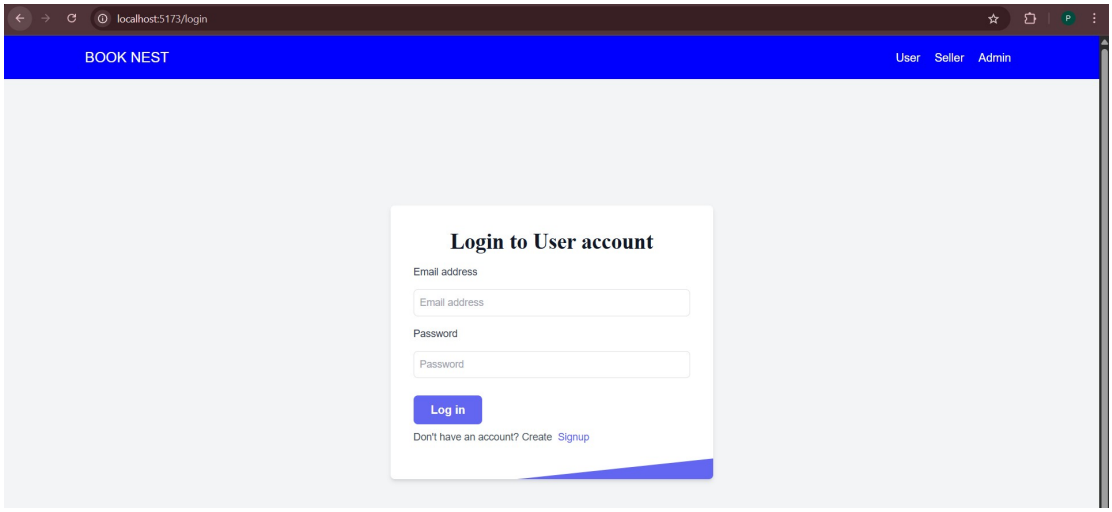
- Fully responsive design using TailwindCSS.
- **Landing Page**

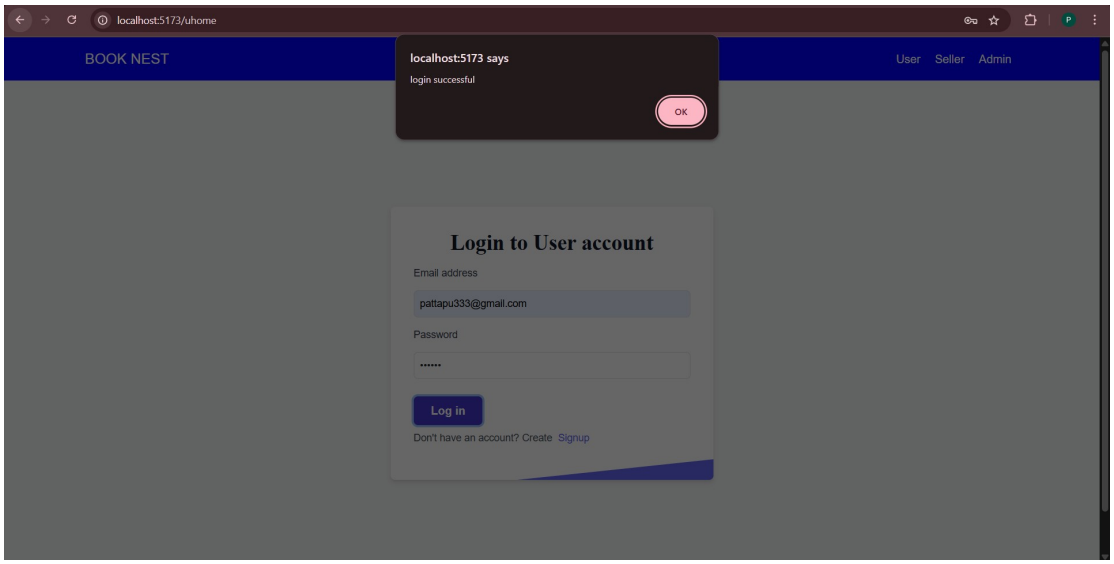


- **User Registration Page**

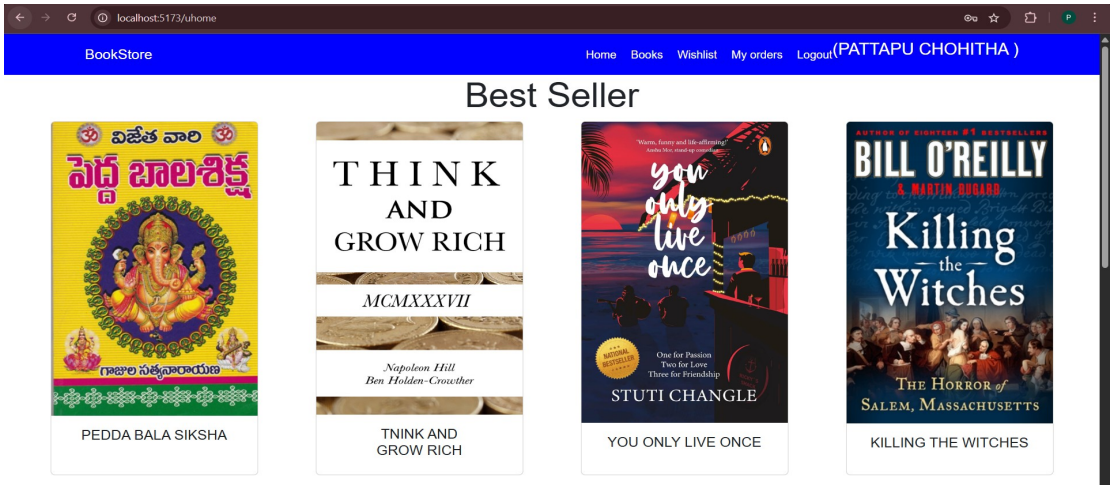


- **User Login Page**

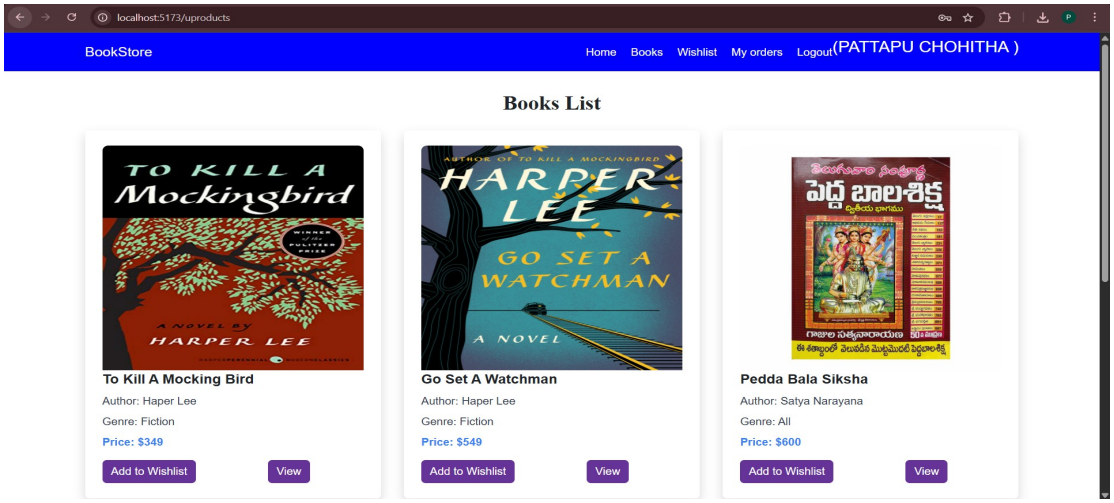




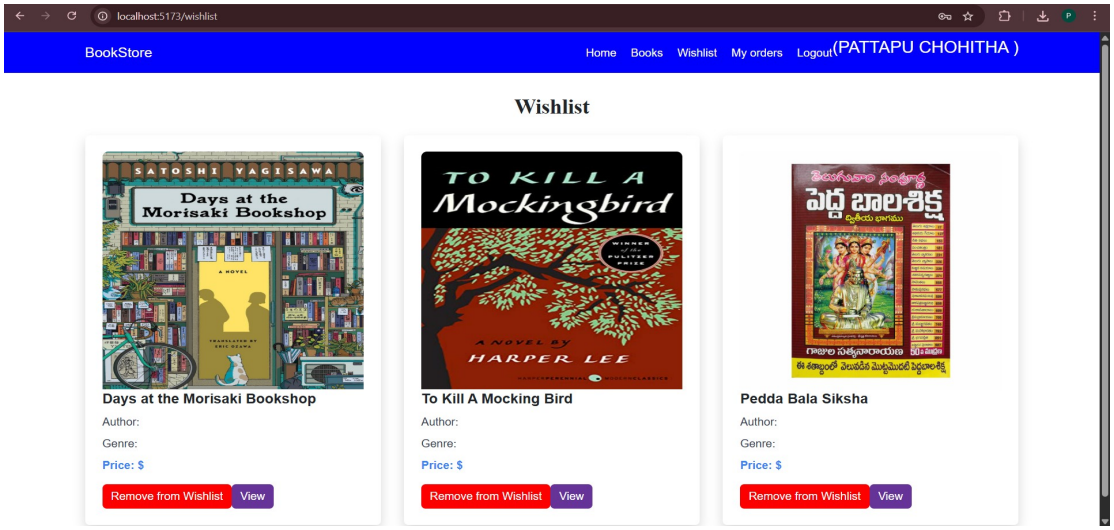
- **User Dashboard**



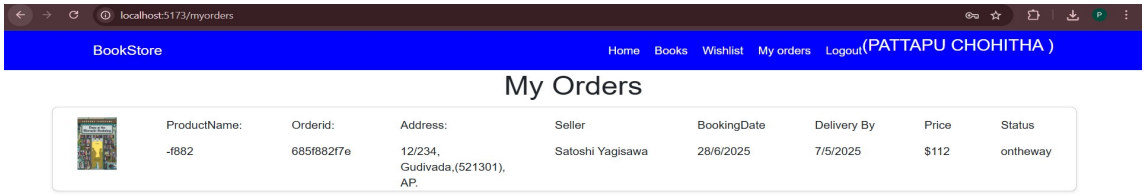
- **Book List Page**



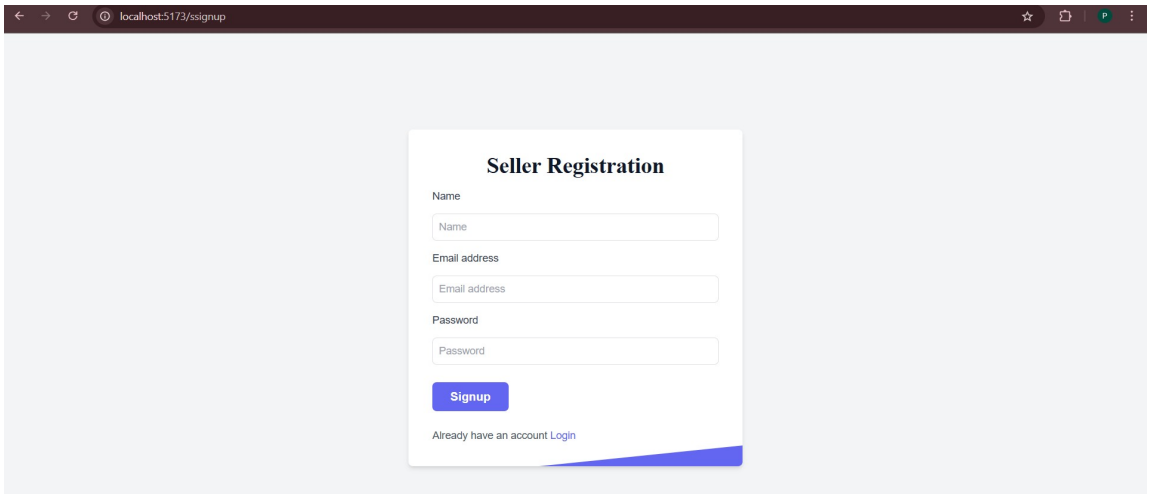
- **User Wishlist Page**



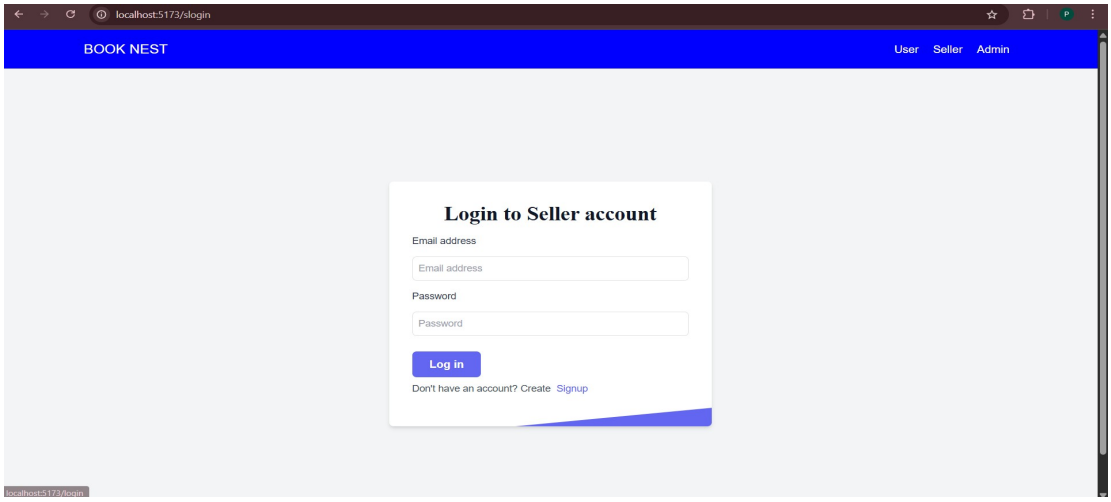
- # User Orders Page

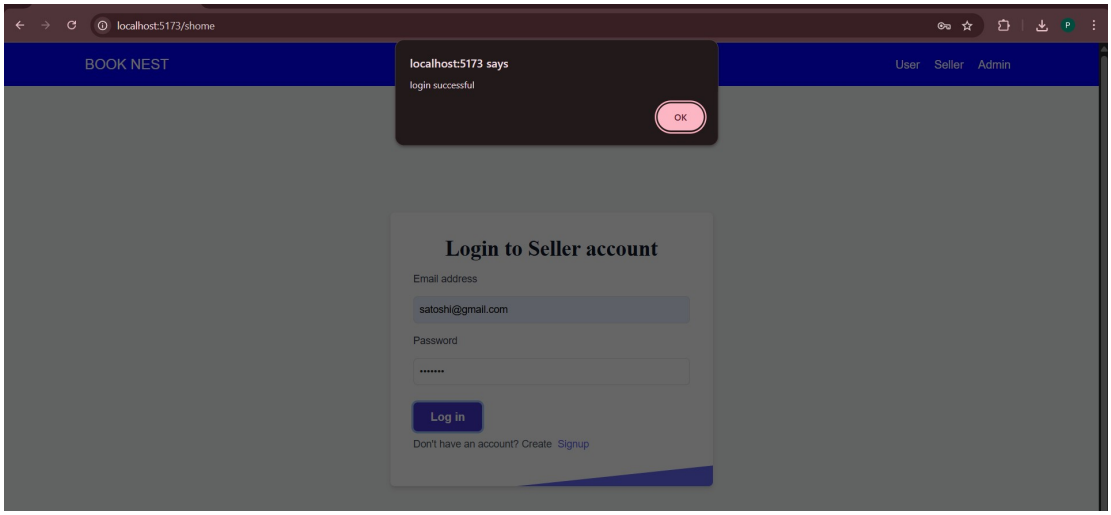


- # Seller Registartion Page

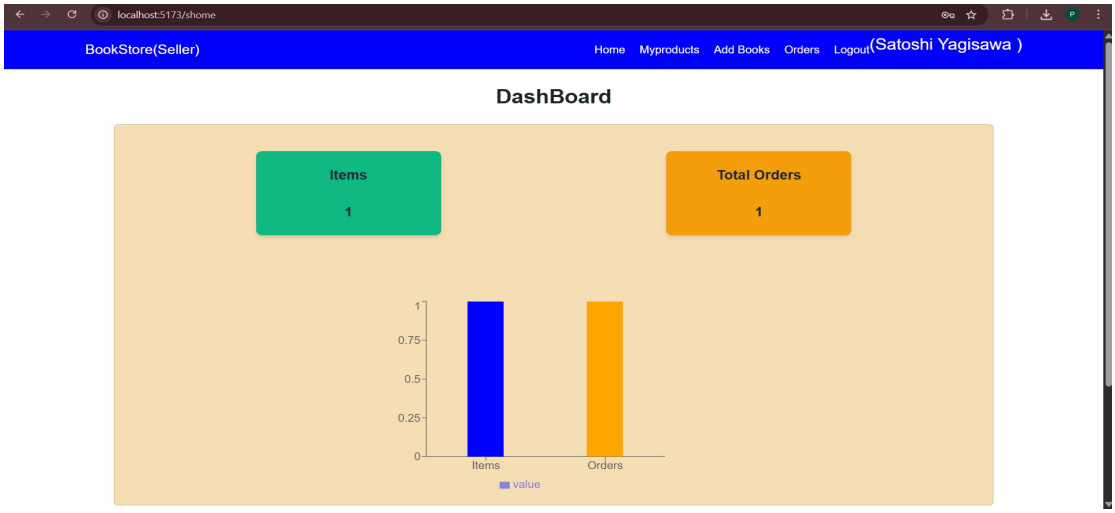


- # Seller Login Page

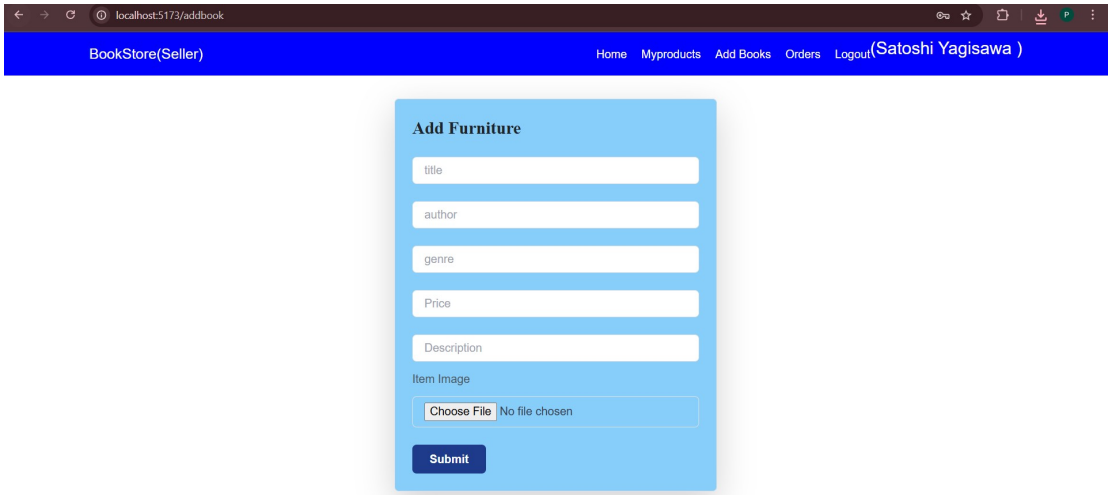


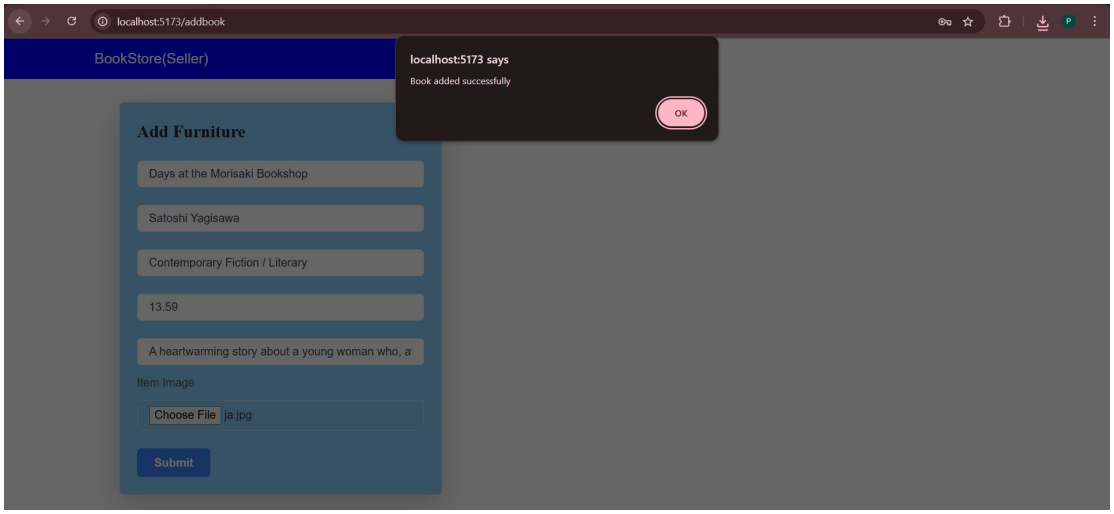


- **Seller Dashboard**

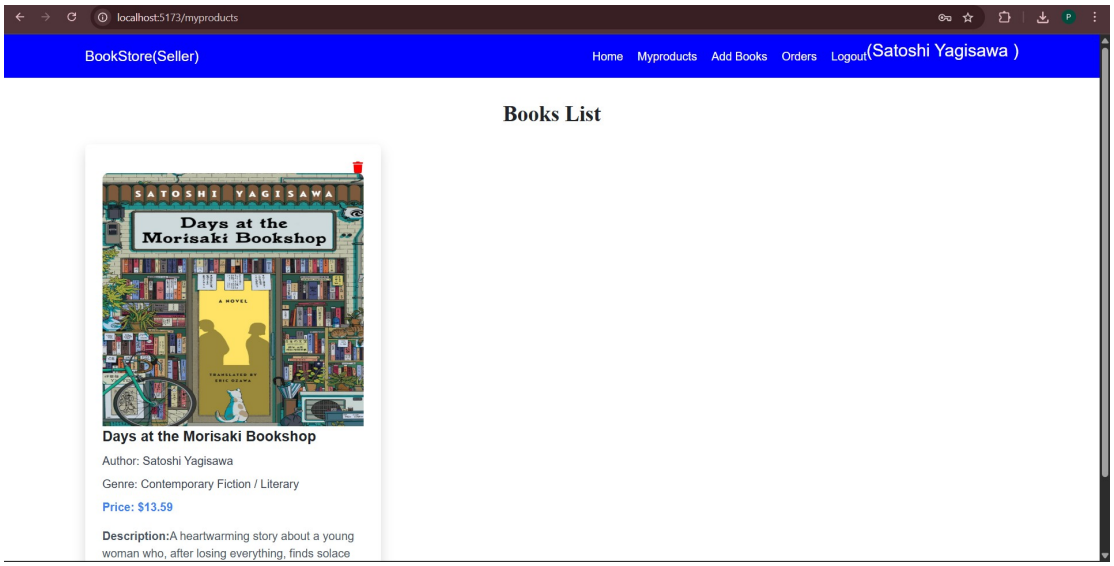


- **Seller Add Book Page**

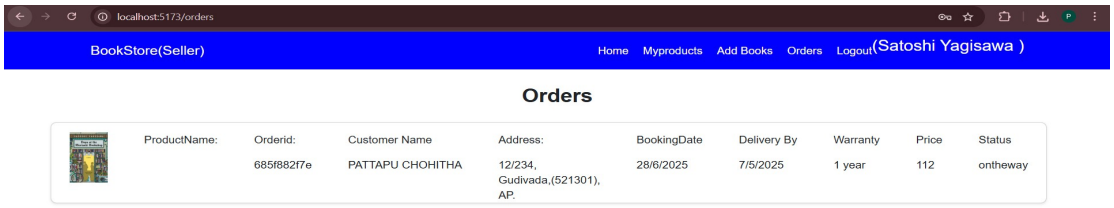




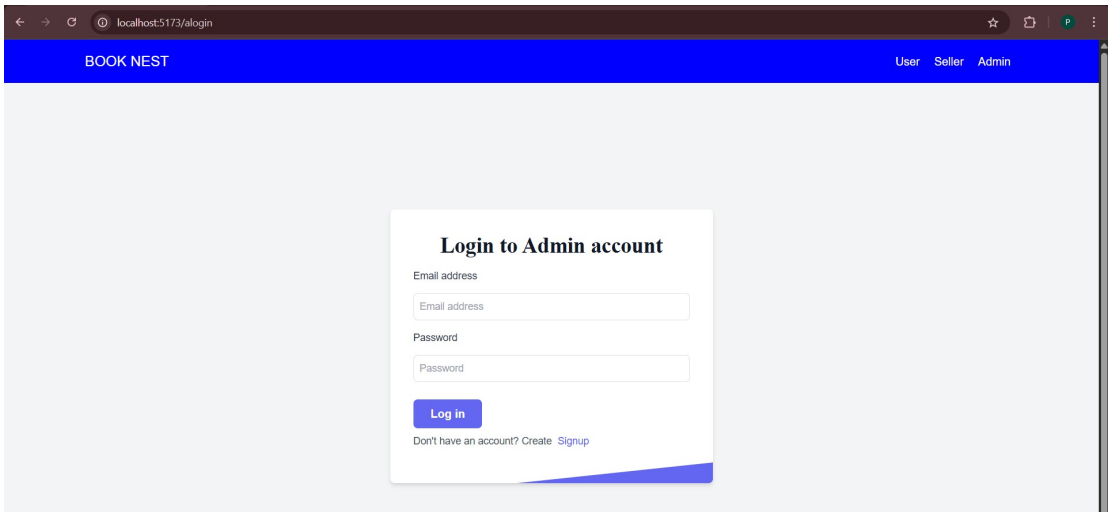
- **Seller Book List Page**



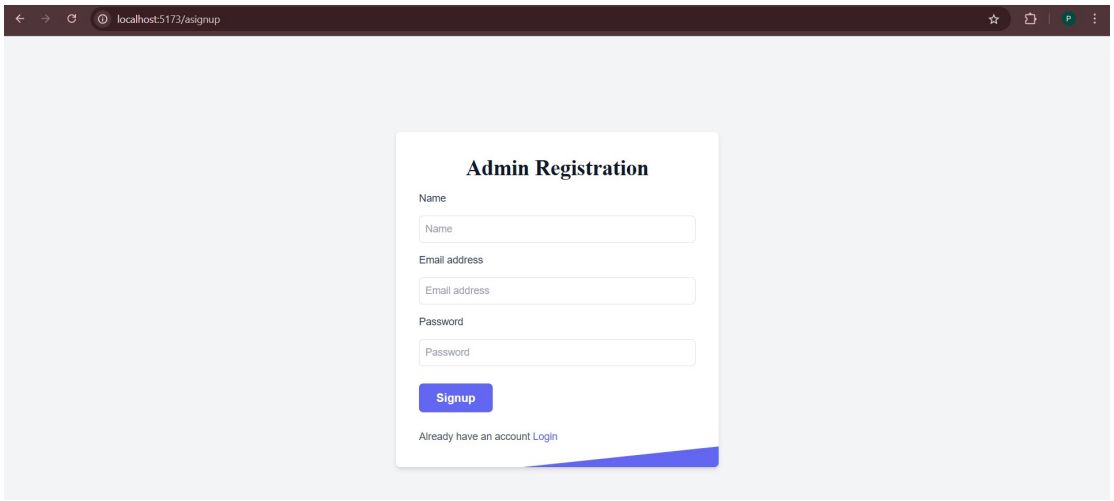
- **Seller Order List Page**

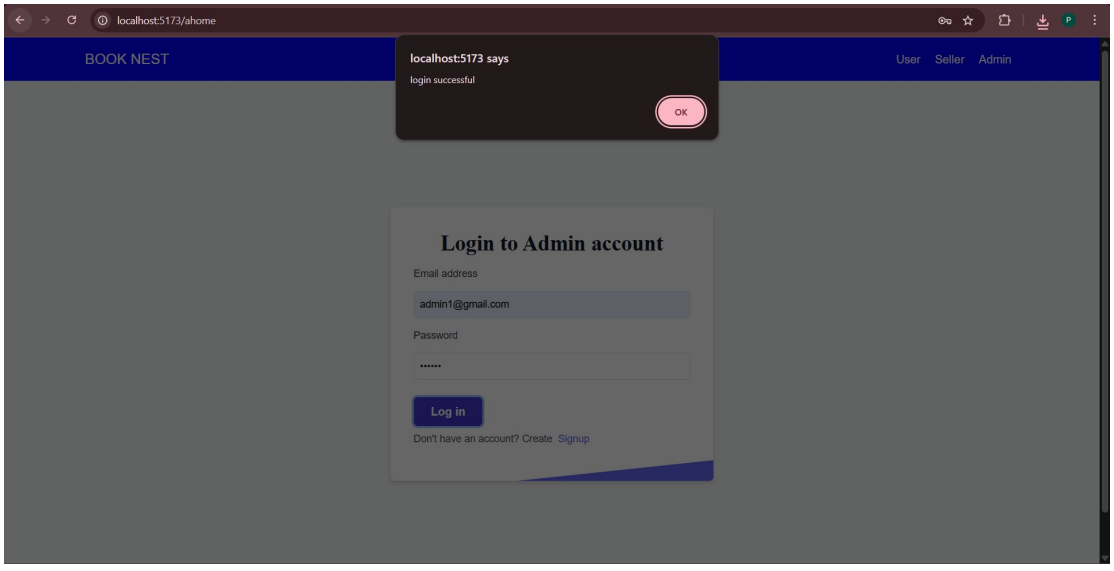


- **Admin Login Page**

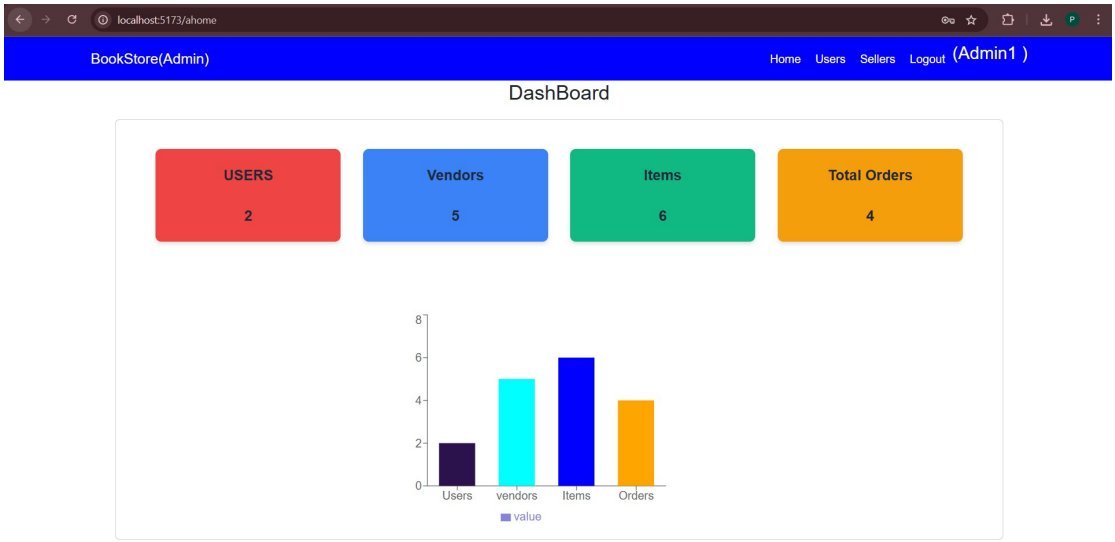


- **Admin Registration Page**

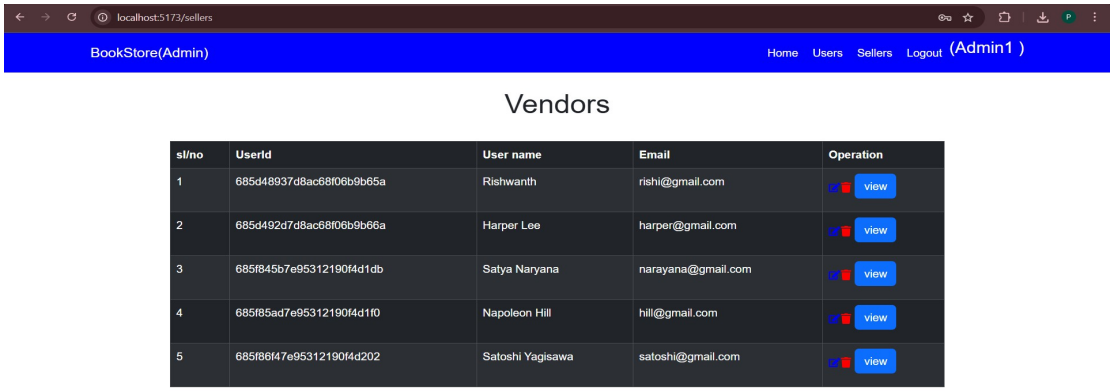




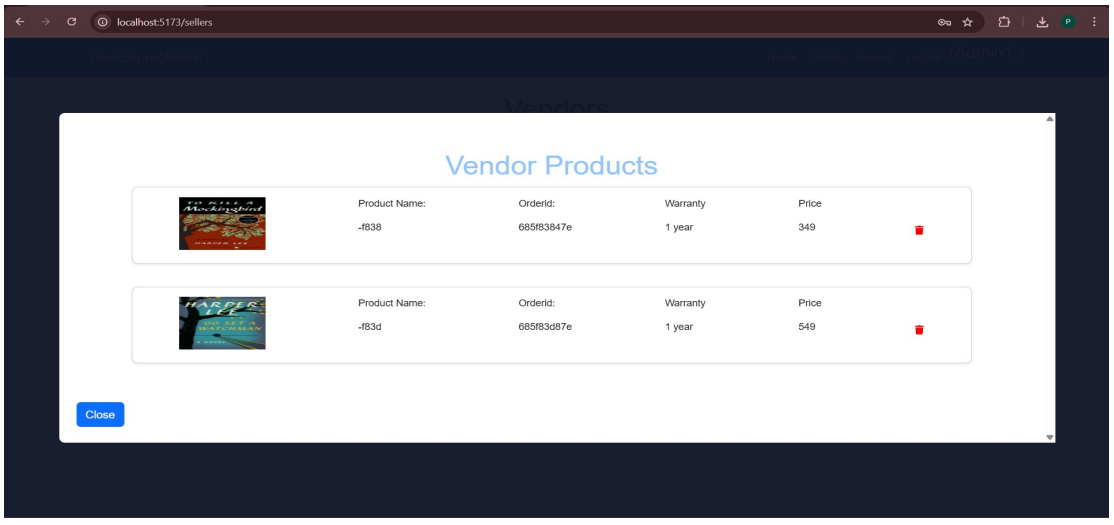
- # Admin Dashboard



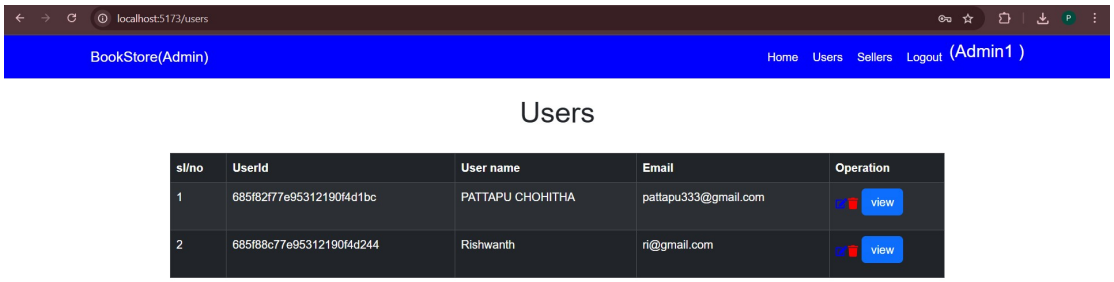
- # Vendors List Page



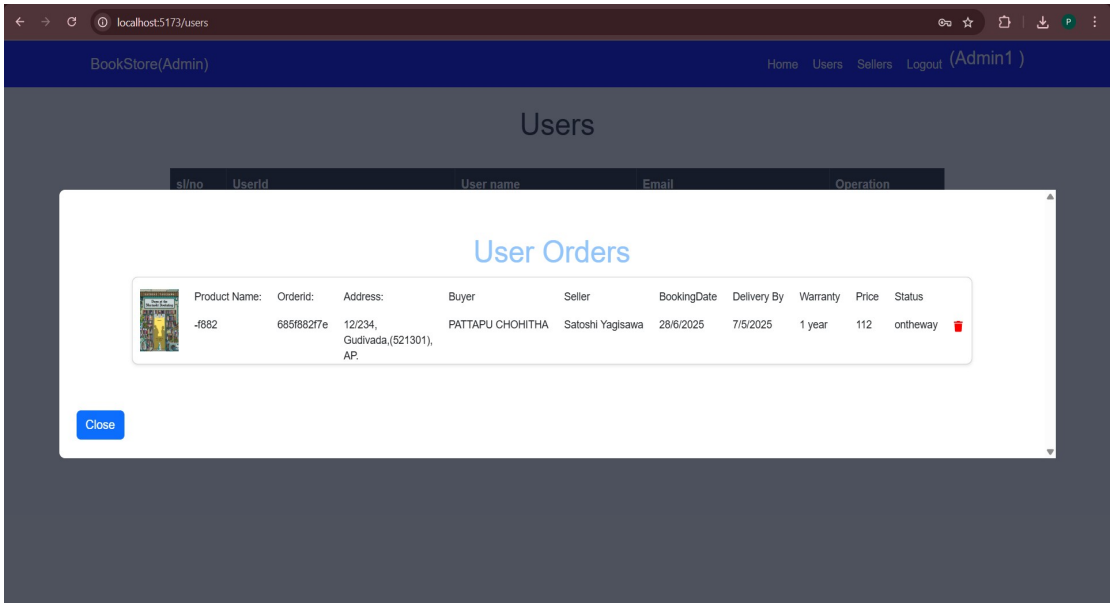
- # Vendor Products Page



- **Users List Page**



- **User Orders Page**



---

10. Testing

- Manual testing using Postman for API validation.
- Load testing via Locust.
- Unit testing planned using Jest (TBD).

---

11. Demo

https://drive.google.com/file/d/1IV60sPoNLlNH6ZdNAh3YPdjUqR8SD4d1/view?usp=sharing

---

12. Known Issues

- No OTP/email verification currently.

---

13. Future Enhancements

- Add payment gateway integration (Razorpay/Stripe)
- Recommendation engine using ML
- User reviews and ratings for books
- Chatbot for book search assistance
- Mobile app version