Project Report : BookNest - Where Stories Nestle

1. INTRODUCTION

1.1 Project Overview

BookNest, a revolutionary MERN Stack-based Book-Store Application. This project aims to transform the conventional bookstore experience into an intuitive, accessible, and immersive digital journey. BookNest offers users a seamless platform to explore, discover, and purchase books from a wide range of genres, authors, and categories.

Crafted using MongoDB, Express.js, React, and Node.js, this application ensures a scalable backend, responsive UI, and smooth performance—providing an enriched book browsing and purchasing experience to every user.

1.2 Purpose

The purpose of this project is to bring the joy of literature to users' fingertips. Whether it's discovering new releases or revisiting timeless classics, BookNest is designed to deliver a user-centric, responsive, and feature-rich platform for readers, sellers, and administrators.

---

2. IDEATION PHASE

2.1 Problem Statement

Sarah is an avid reader with a passion for exploring new genres and authors. However, her busy schedule often leaves her with limited time to visit physical bookstores. She is looking for a solution that allows her to discover and purchase books conveniently, without compromising her reading preferences or the joy of browsing through a bookstore.

2.2 Empathy Map Canvas

- **Thinks:** "Will I find a book I truly enjoy?"
- **Feels:** Frustrated by poor recommendations or lack of time.
- **Says:** "I wish I could explore a variety of books easily".
- **Does:** Searches multiple sites, checks reviews, adds to wishlist
- **Goal:** A smart, simple, and engaging online bookstore.

2.3 Brainstorming

- User Registration and Authentication
- Book Listings with search and filter
- Role-based dashboards (User/Seller/Admin)
- Cart, Order, and Wishlist functionality
- Inventory and Order Management Services
- Secure purchase and order confirmation

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

- User visits BookNest
- Registers/Login
- Browses or searches books
- Adds books to cart
- Checks out and receives confirmation
- Sellers manage books
- Admin tracks performance

3.2 Solution Requirements

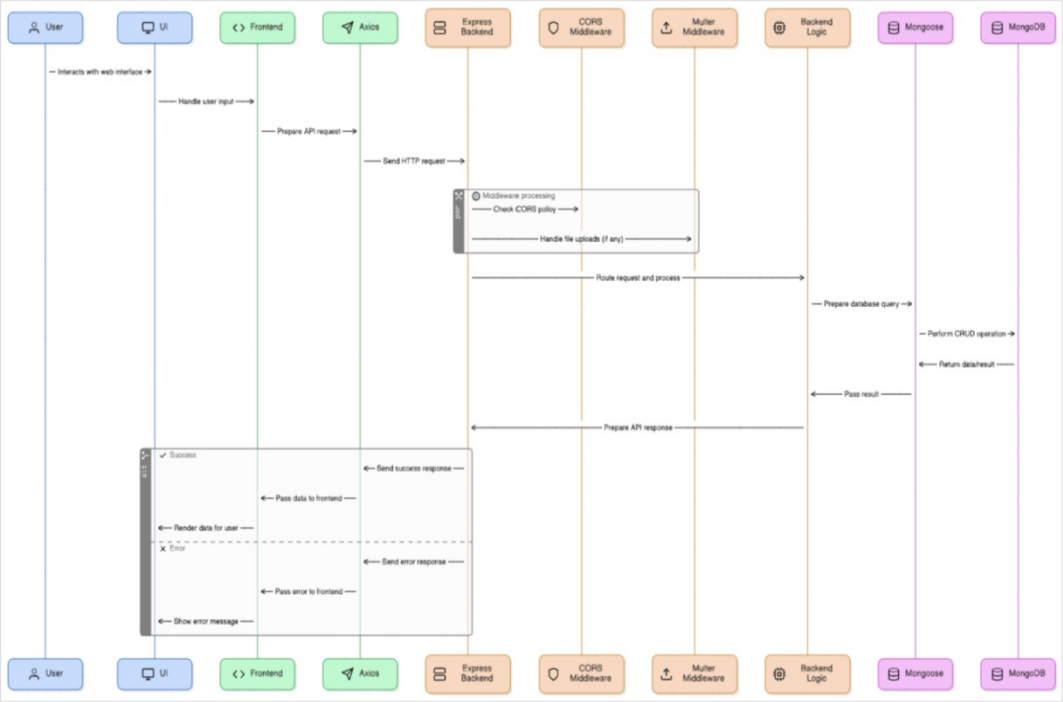
Functional:

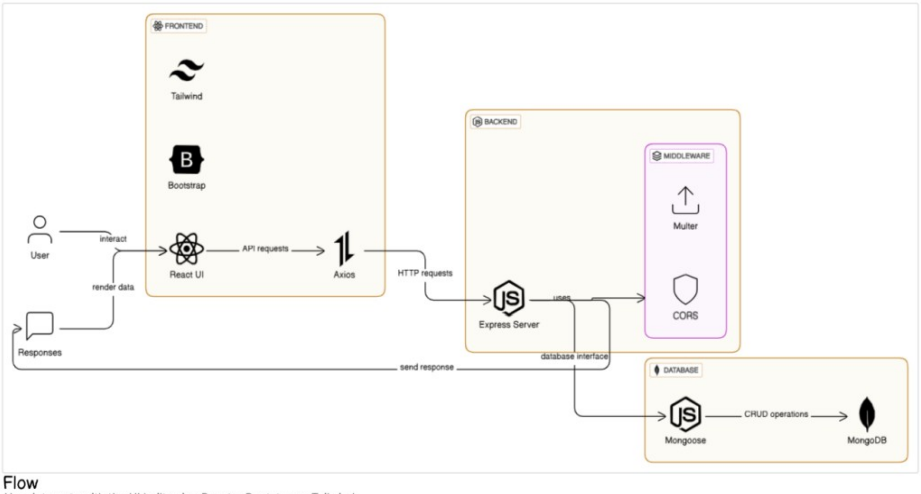
- User/Seller/Admin login
- Book browsing, search, and filter
- Cart and order placement
- Seller uploads and manages inventory
- Admin dashboard and sales tracking

Non-Functional:

- Security (JWT, hashed passwords)
- Usability (Intuitive UI/UX)
- Performance (fast loading, responsive)
- Scalability (cloud-hosted backend)

3.3 Data Flow Diagram





Flow
User interacts with the UI built using React + Bootstrap + Tailwind.
Frontend uses Axios to send API requests to Express routes.
Backend uses middleware like CORS and Multer (for uploads).
Backend logic interfaces with MongoDB via Mongoose to perform CRUD.
Responses are sent back to the frontend and rendered accordingly.

3.4 Technology Stack

- **Frontend:** React, TailwindCSS, Bootstrap
- **Backend:** Node.js, Express
- **Database:** MongoDB
- **Authentication:** JWT
- **Deployment:** Render / Vercel / Heroku

4. PROJECT DESIGN

4.1 Problemc - Solution Fit

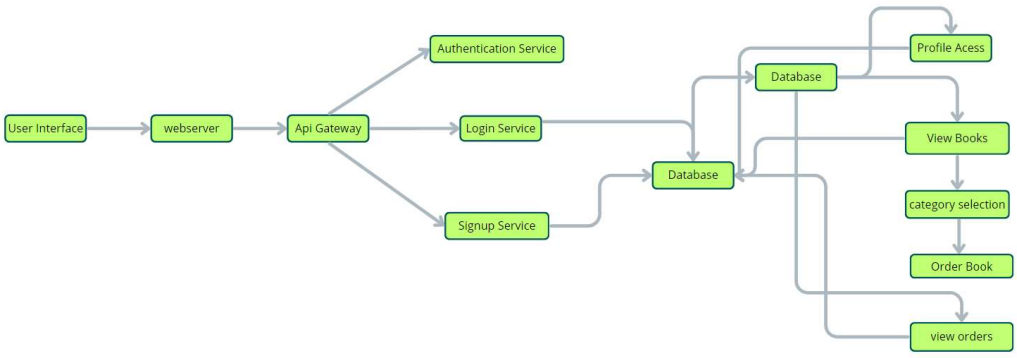
Readers like Sarah want simple, efficient, and immersive digital bookstores. BookNest meets this demand by offering a robust book management and ordering platform.

4.2 Proposed Solution

- Secure registration and login
- Browse books by category, author, or genre
- Smart filtering and cart functionality
- Seller upload and admin control panels
- Real-time order updates and confirmation.

4.3 Solution Architecture

- **User Interface** (React): Book browsing and selection
- **Web Server** (Node/Express): Request routing and handling
- **API Gateway:** Connects client requests to backend services
- **Authentication Service:** Manages tokens and access
- **Database** (MongoDB): Stores books, users, orders
- **Inventory Service:** Manages stock and availability
- **Order Management:** Manages purchases and order tracking



---

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project was structured using Agile Scrum methodology, divided into two focused sprints over 10 working days. Tasks were estimated and

prioritized using story points, with daily check-ins to maintain alignment and velocity.

- **Methodology:** Agile (2 Sprints)

Sprint	Duration	Key Tasks	Deliverables
Sprint 1	5 days	UI/UX Design for Homepage and Login - User/Admin/Seller Registration & JWT Authentication - MongoDB schema setup for Users/seller/admin & Books- Frontend-Backend connection testing	Functional UI with secure login Connected database models
Sprint 2	5 days	-Cart and Order module development- Seller dashboard and Admin panel- Inventory management service- Deployment to Render/Vercel	Deployed full-stack app Seller/Admin /User features and order flow

6. FUNCTIONAL & PERFORMANCE TESTING

6.1 Performance Testing

Response time for book search API: ~0.35s

Concurrent user testing: Stable till 50+ users

Endpoint	Avg Time	Max Users	Error Rate
GET/books/search	0.35s	50	0.1%
POST /orders	0.40s	30	0.8%
Admin Stats	0.28s	20	0%

Tools Used: Postman, Locust

Performance testing ensured the application performed well under expected loads. Locust was used for load testing, and Postman validated

API response times. The system proved stable for up to 50 concurrent users, with a response time of under 0.5 seconds.

Additional test cases executed:

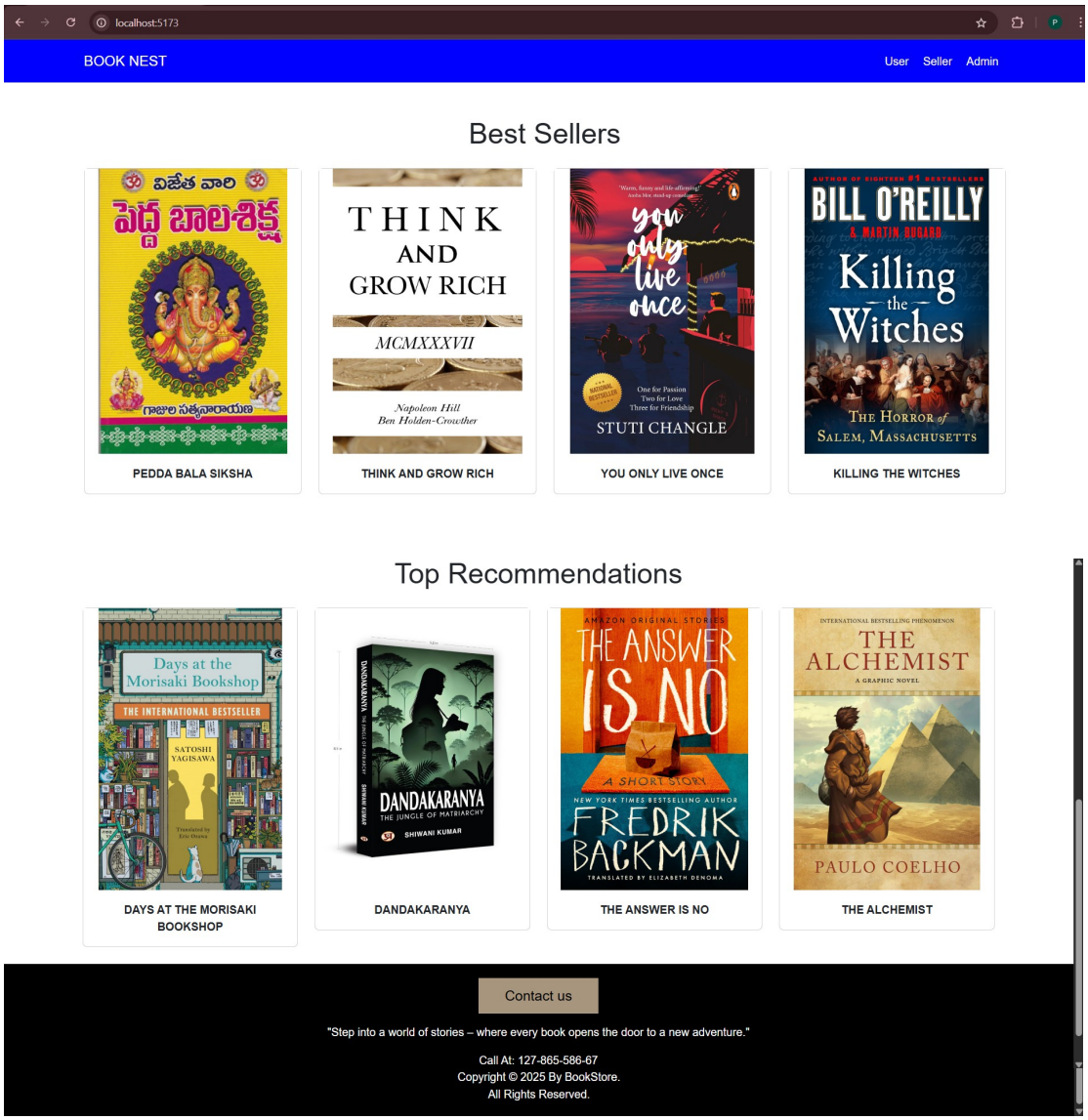
- **User Login Test**: Ensured secure login/logout functionality with valid and invalid credentials.
- **Cart Functionality Test**: Verified addition and removal of books from cart.
- **Order Flow Test**: Ensured smooth transition from cart to order confirmation

---

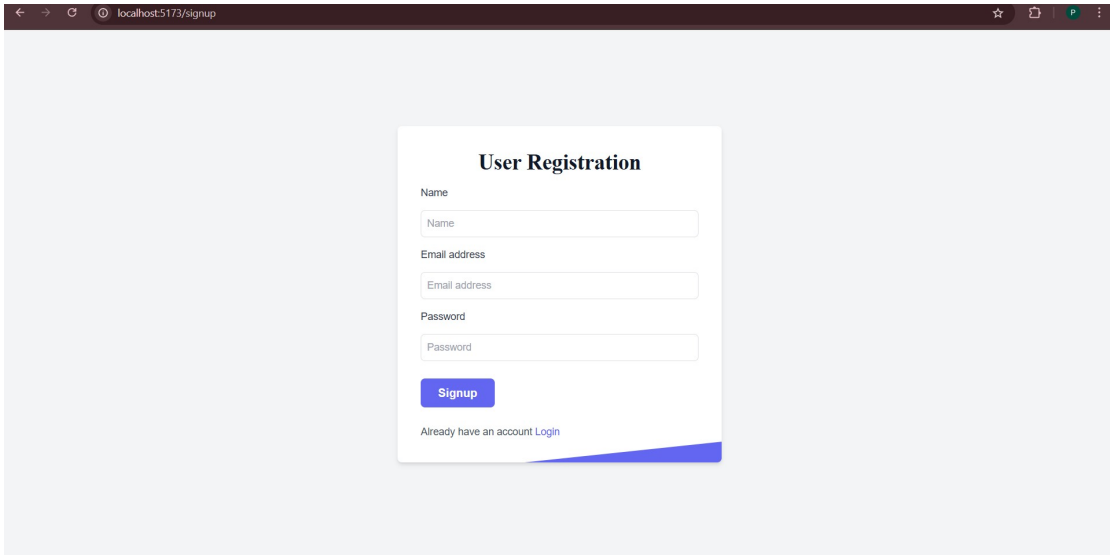
7. RESULTS

7.1 Output Screenshots

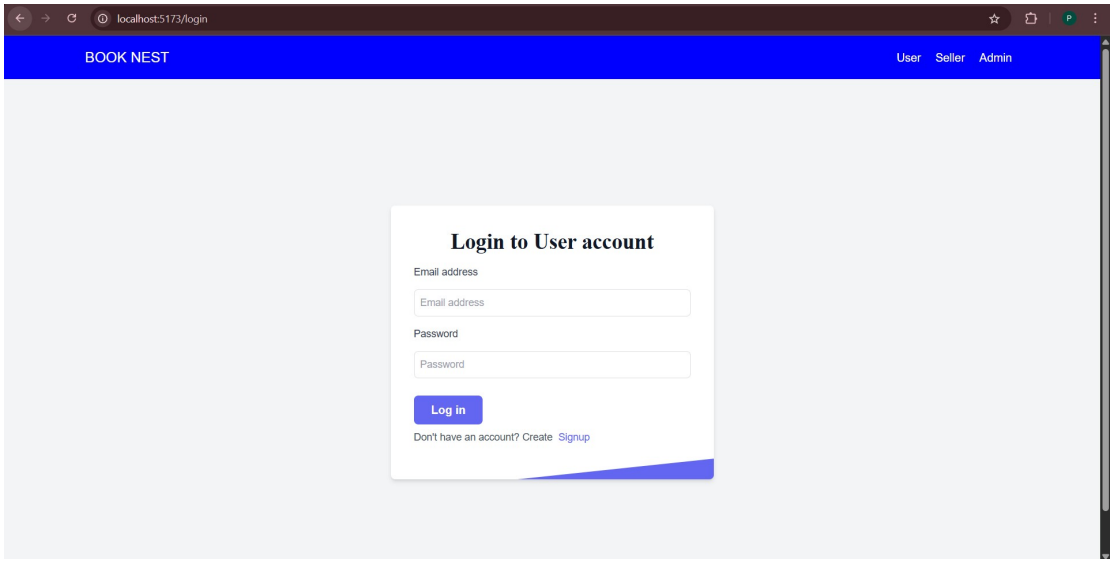
- **Landing Page**

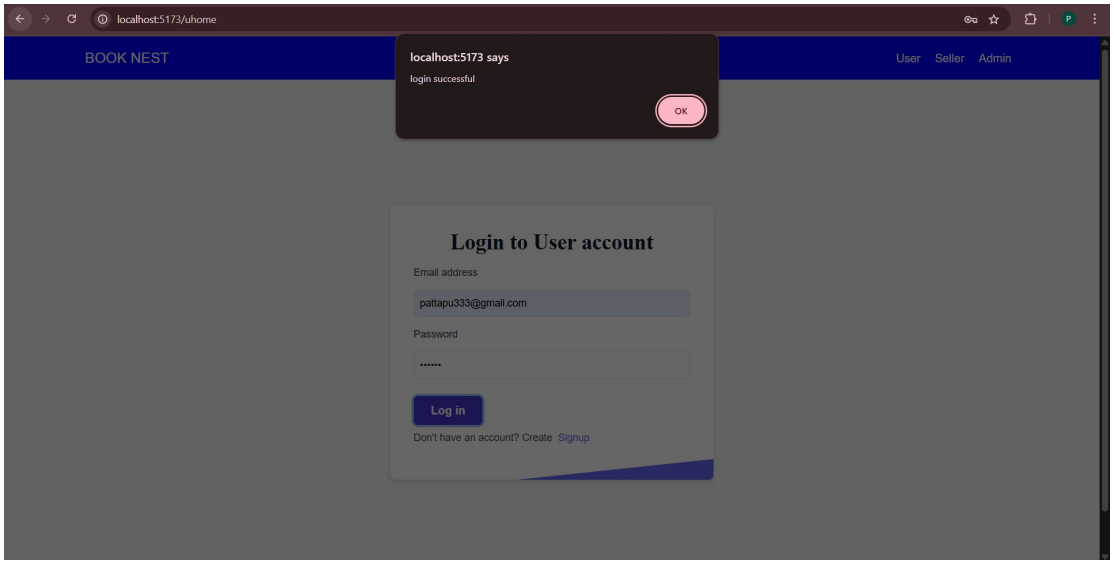


- **User Registration Page**

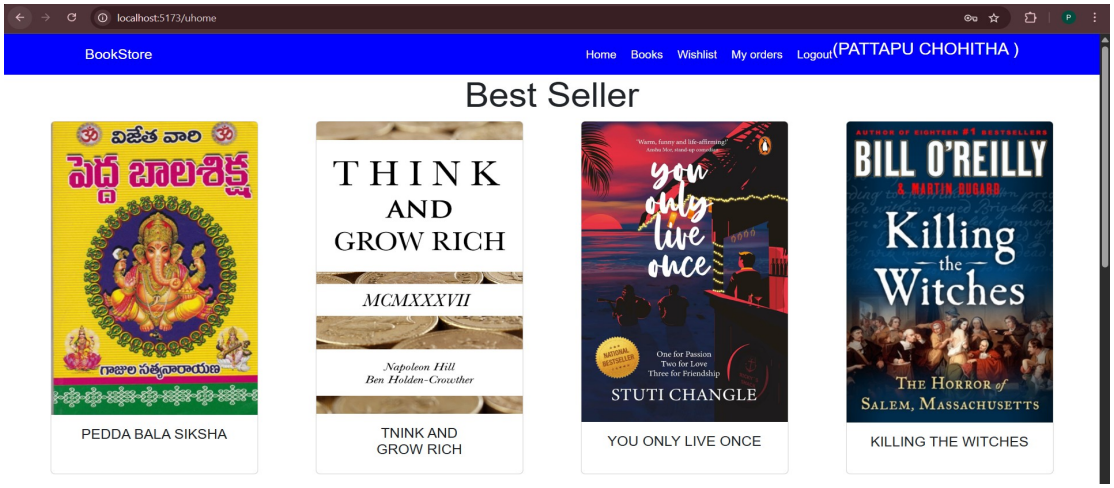


- **User Login Page**

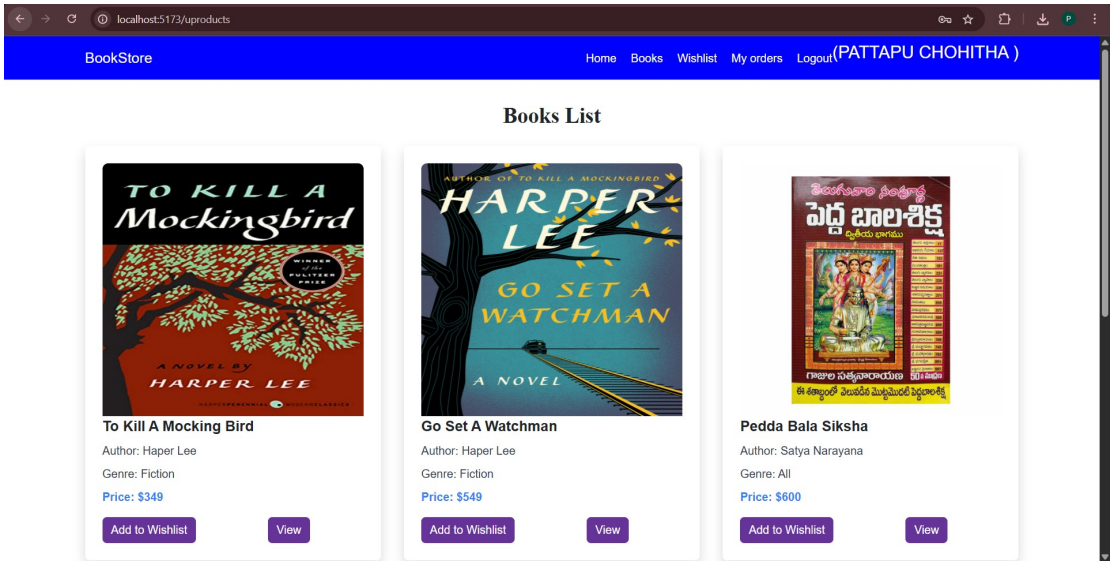




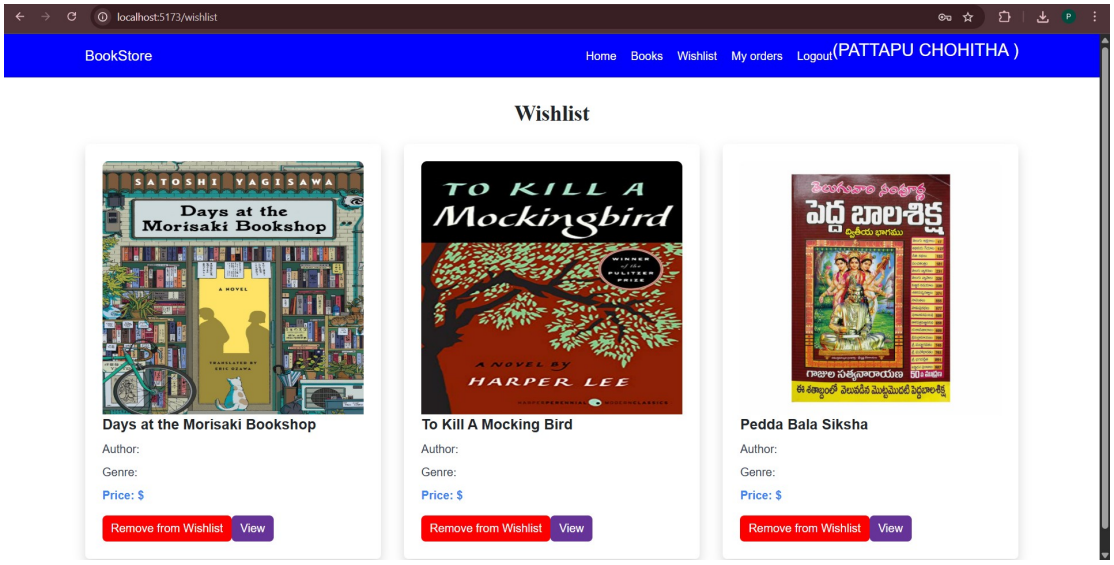
- # User Dashboard



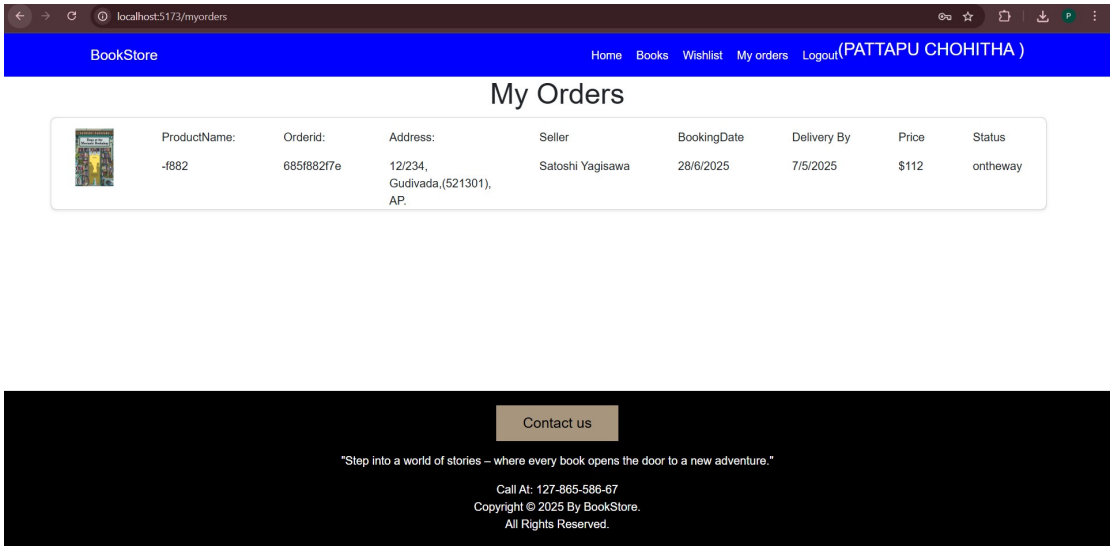
- # Book List Page



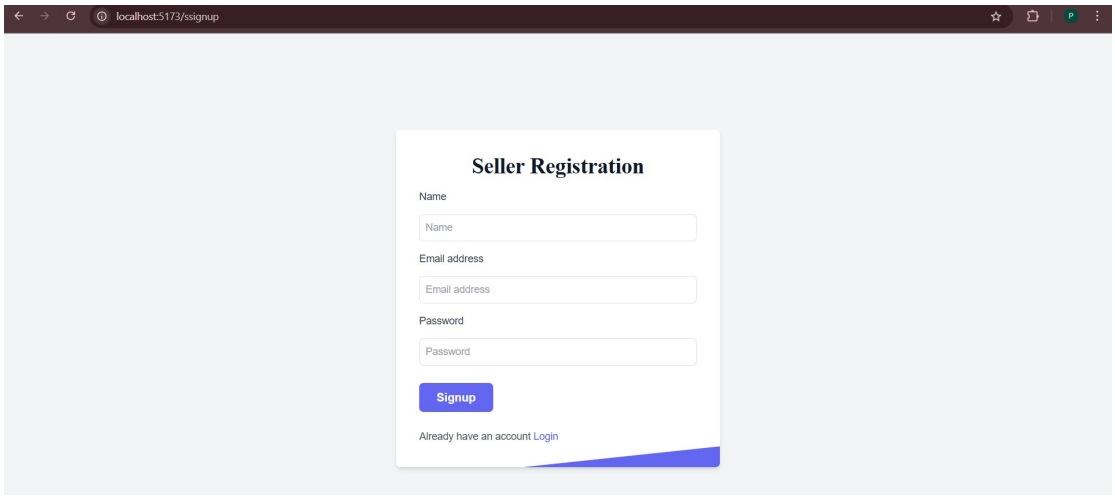
- # User Wishlist Page



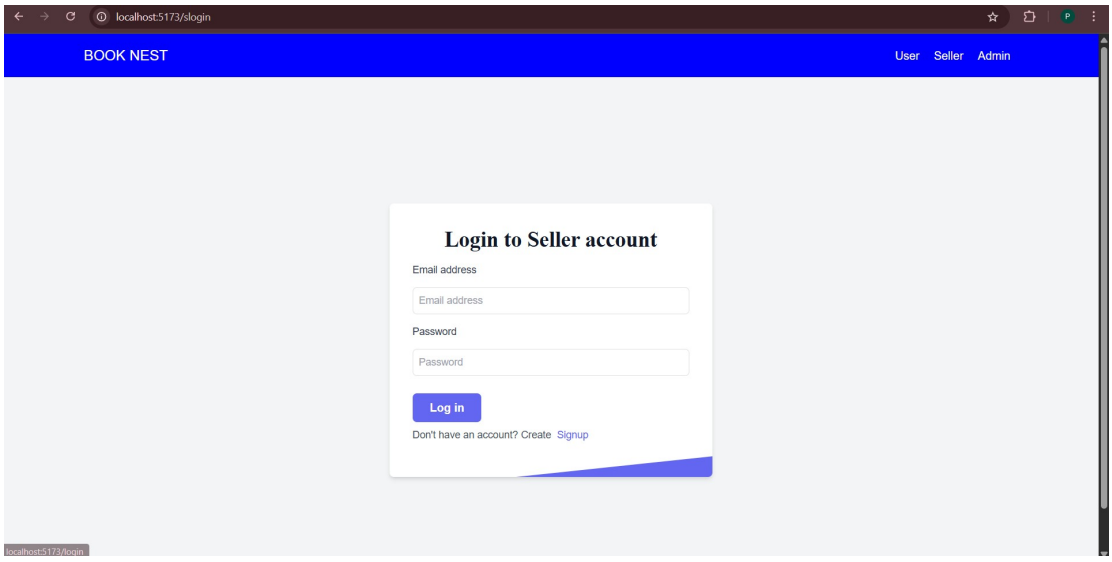
- # User Orders Page

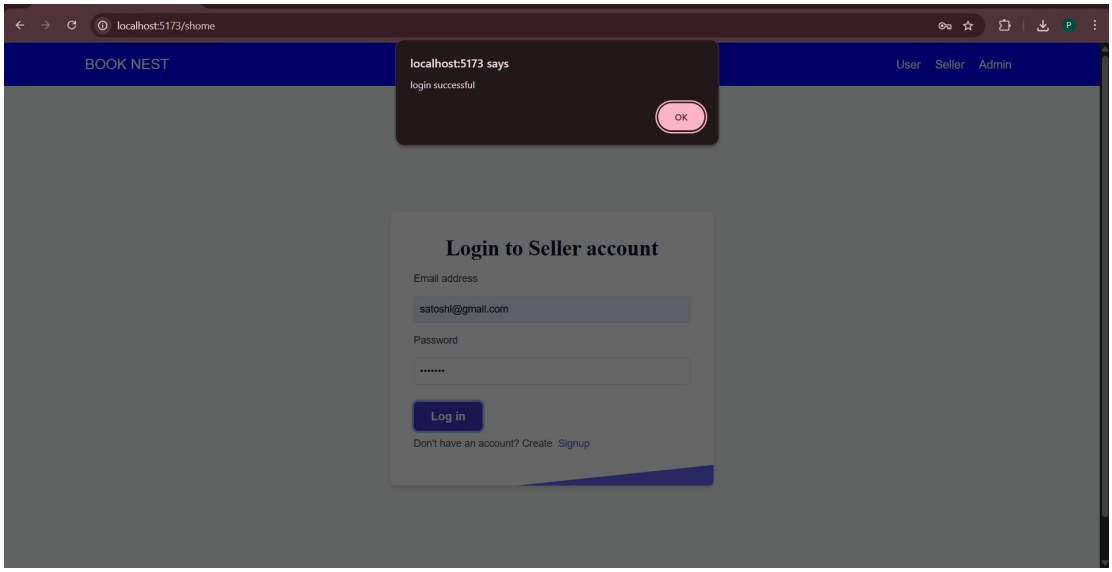


- # Seller Registartion Page



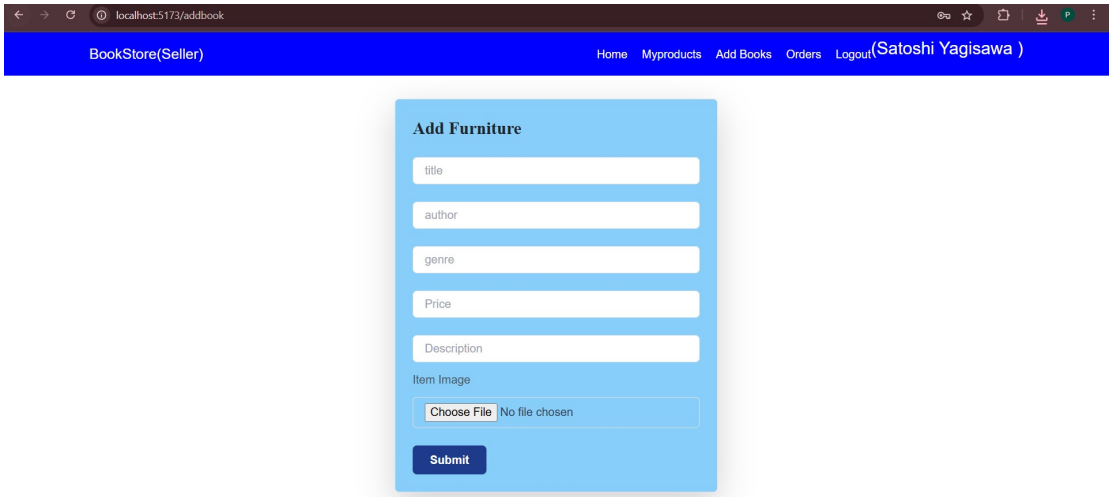
- # Seller Login Page

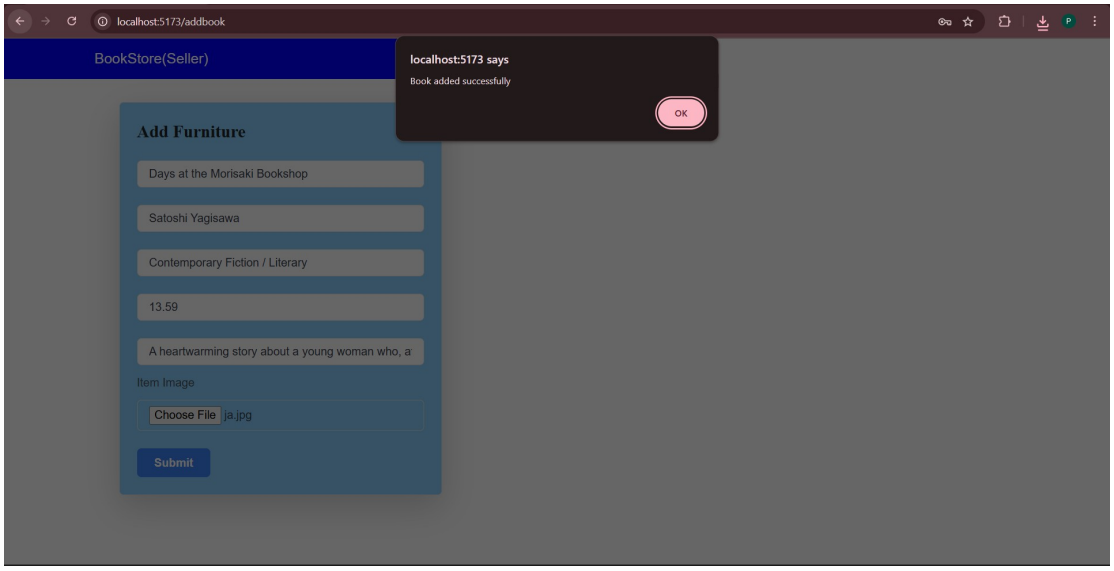




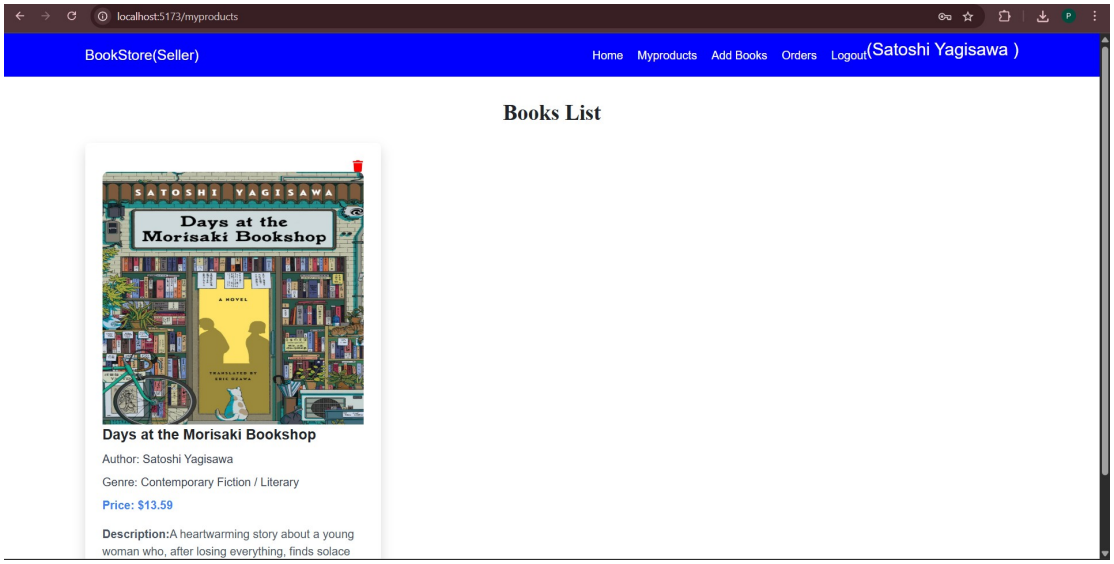
- **Seller Dashboard**

- **Seller Add Book Page**

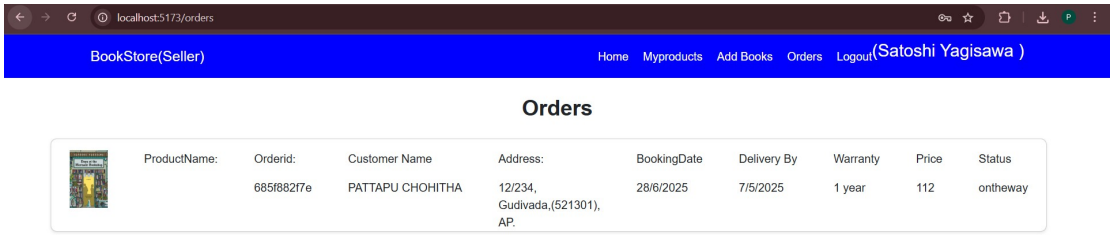




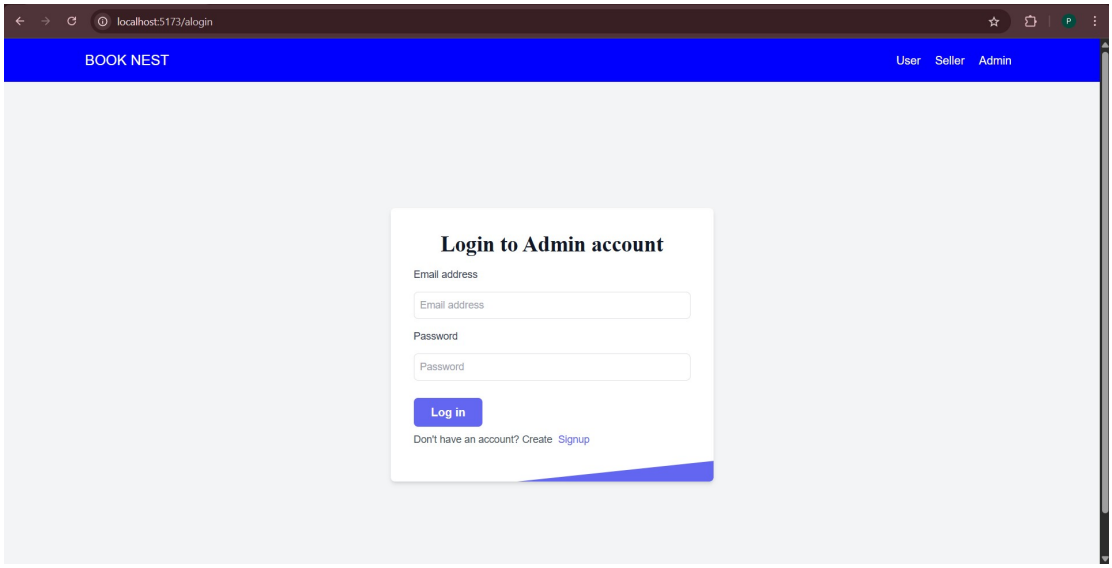
- **Seller Book List Page**



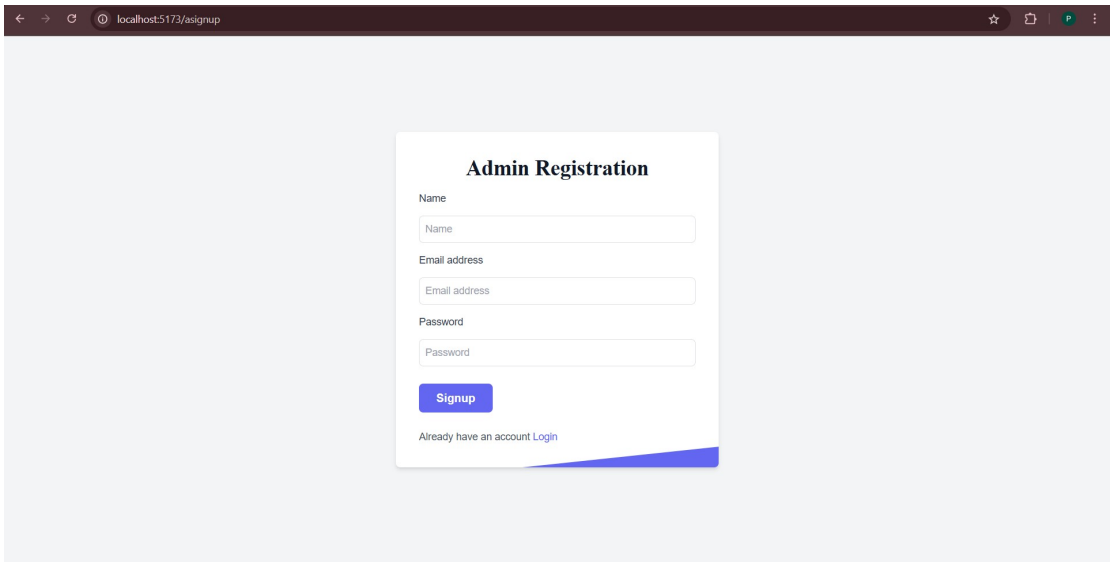
- **Seller Order List Page**

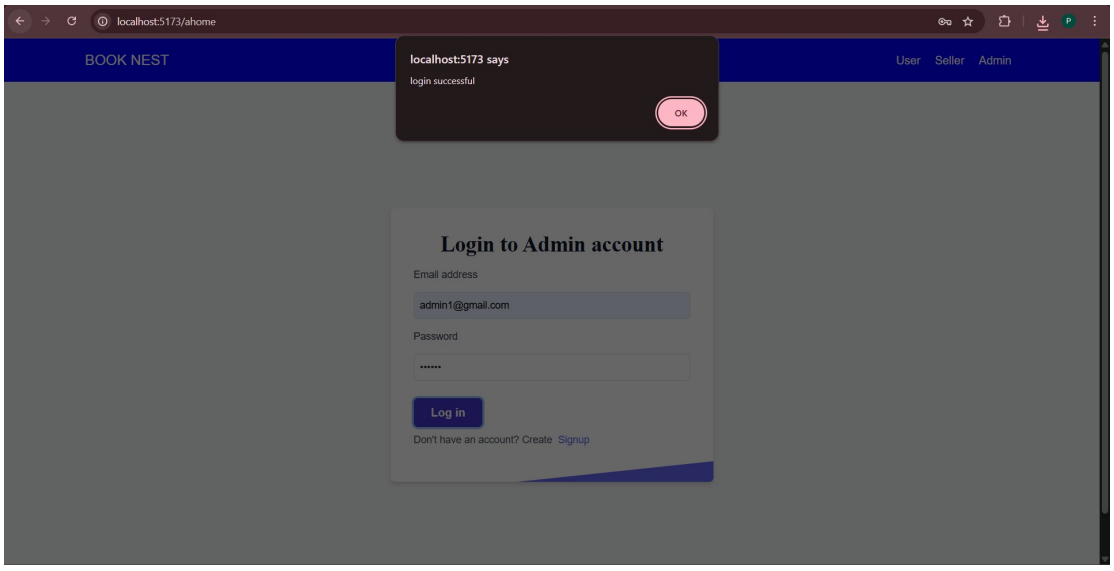


- **Admin Login Page**

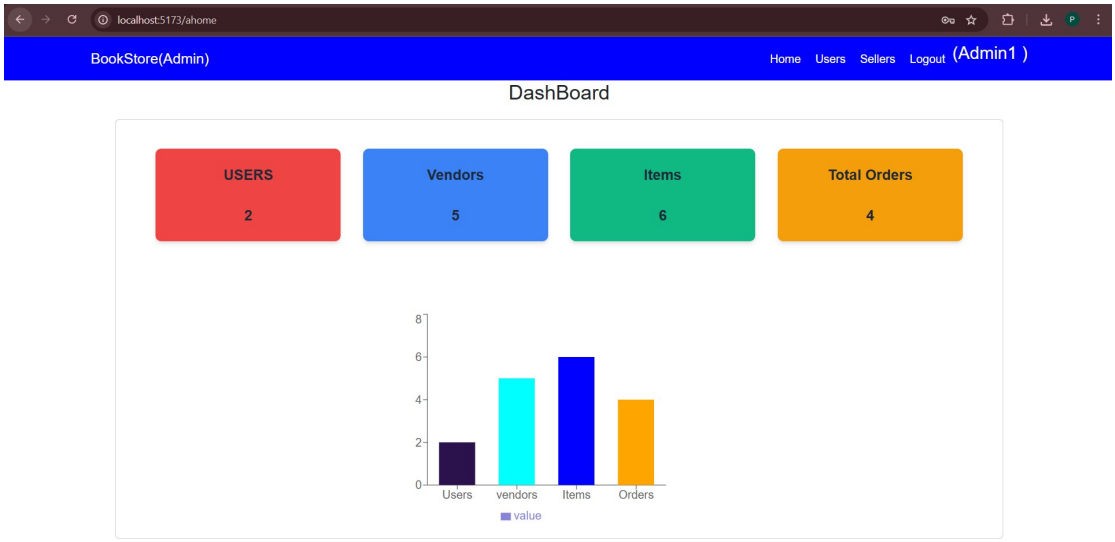


- **Admin Registration Page**

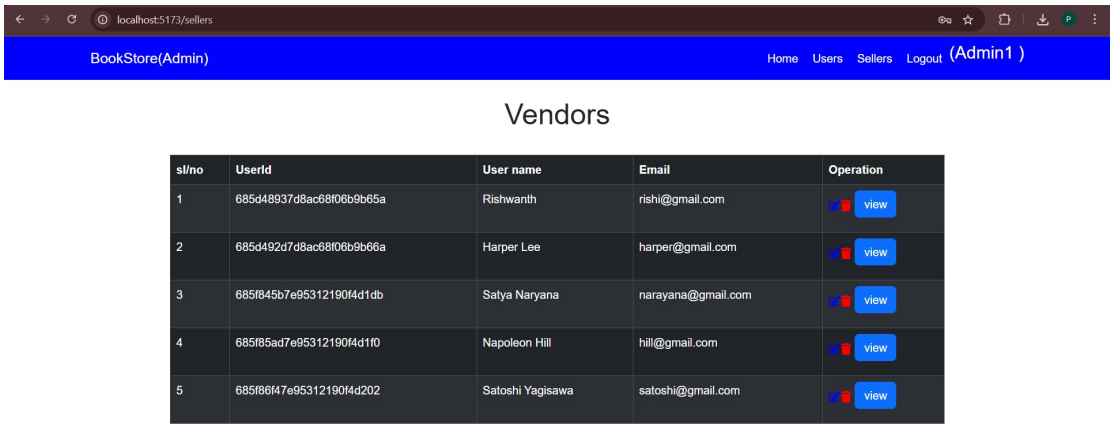




- # Admin Dashboard

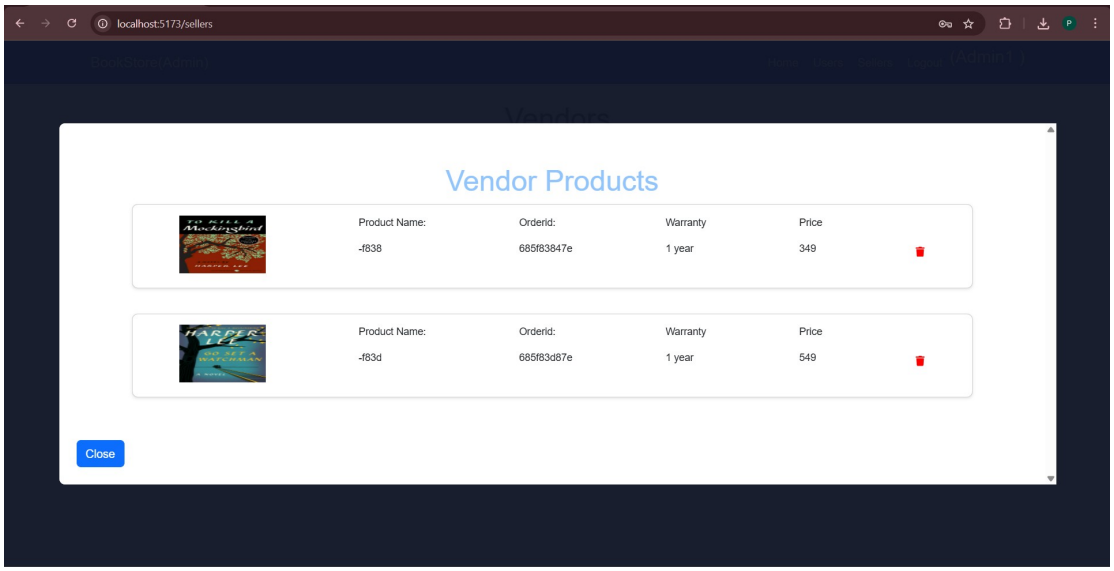


- # Vendors List Page

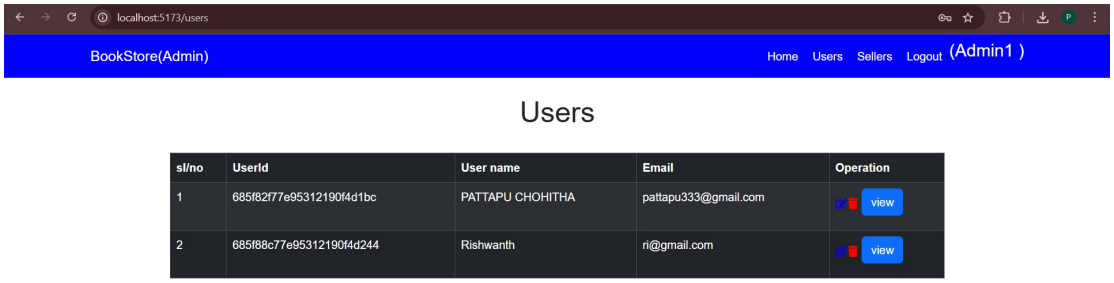


sl/no	UserId	User name	Email	Operation
1	685d48937d8ac68f06b9b65a	Rishwanth	rishi@gmail.com	view
2	685d492d7d8ac68f06b9b66a	Harper Lee	harper@gmail.com	view
3	685f845b7e95312190f4d1db	Satya Naryana	narayana@gmail.com	view
4	685f85ad7e95312190f4d1f0	Napoleon Hill	hill@gmail.com	view
5	685f86f47e95312190f4d202	Satoshi Yagisawa	satoshi@gmail.com	view

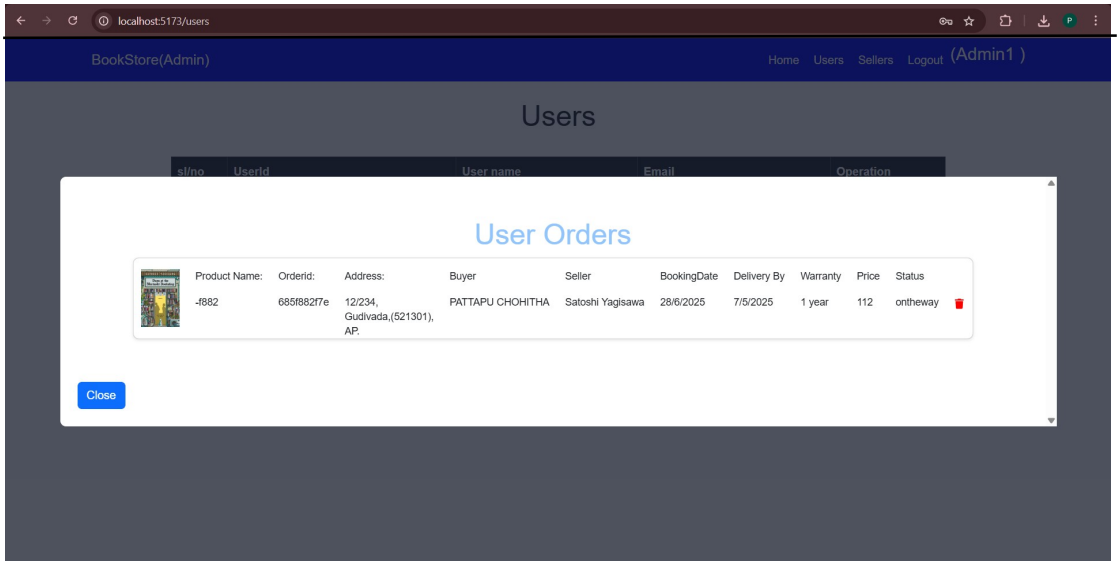
- # Vendor Products Page



- **Users List Page**



- **User Orders Page**



8. ADVANTAGES & DISADVANTAGES

Advantages:

- Simple, elegant UI/UX
- Secure and scalable architecture
- Multi-role functionality

Disadvantages:

- Lacks recommendation engine (for now)
- No integrated payment gateway.

9. CONCLUSION

BookNest brings together the love for reading and the power of modern web technologies. It provides an immersive book browsing and buying experience backed by a scalable and secure MERN stack. The system is fully functional and easily extendable to include more advanced features like personalized recommendations and integrated payments. The project not only meets its initial objectives but lays a strong foundation for future innovations in the digital bookstore domain.

.10. FUTURE SCOPE

- **Personalized Book Recommendations**: Integrate collaborative filtering or content-based ML models.
- **Payment Integration**: Add Razorpay or Stripe for online transactions.
- **Community Building**: Enable reviews, ratings, and book discussion forums.
- **AI-Powered Chatbot**: For user assistance in finding books or order help.
- **Mobile App Version**: To increase accessibility and user engagement.

---

11. APPENDIX

- **Source Code:** https://github.com/PattapuChohitha/BookNest
- **DemoVideo:** https://drive.google.com/file/d/1IV60sPoNLlNH6ZdNAh3YPdjUqR8SD4d1/view?usp=sharing