PROJECT DOCUMENTATION

BookNest: Where stories nestle(MERN Stack)

Technology Stack: React.js | Node.js | Express.js | MongoDB

Modules: User | Seller | Admin | Cart | Orders | Wishlist | Feedback

Project Title: BookNest: Where stories nestle

Team ID : LTVIP2026TMIDS24176

Team Size: 4

Team members and Roles:

Team Leader: Bellamkonda Jaswan (Frontend Developer)

Team Member: Dangeti Siva Akshith(Frontend Support)

Team Member: Devanaboyina Charan Chaitanya(UI/UX Designer)

Team Member: Panchakarla Harshitha (Full Stack Developer, Backend Developer)

Table of Contents

BookNest (BookStore)
Full Stack MERN Project Documentation

1. Introduction

Project Title

BookNest (BookStore): Where stories nestle

About the Project

BookNest is a full-stack web application developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). It acts as an online book store where users can browse books, view details, add books to cart, place orders, and maintain a wishlist. The platform also provides role-based dashboards for Users, Sellers, and Admins.

The system is designed to simulate a real-world e-commerce bookstore. Users can register and login, explore books, manage their cart, and track orders. Sellers can add and manage their books and view incoming orders. Admins can manage books, users, orders, and wishlist data from an admin panel.

Objective

- To develop a responsive and scalable web-based book store solution using the MERN stack.
- To implement secure authentication with role-based access (User / Seller / Admin).
- To allow users to browse books, manage wishlist and cart, and place orders.
- To allow sellers to manage their products and track orders.

- To provide an admin panel to manage users, books, orders, and platform activity.

2. Project Overview

Purpose

The purpose of BookNest is to design and implement a fully functional, user-friendly, and responsive online bookstore platform. In modern digital commerce, users prefer browsing and purchasing products online due to convenience and speed. BookNest meets this need by offering a centralized platform where books are listed with details such as title, author, price, and cover image.

The platform supports three roles: User, Seller, and Admin. Each role has its own dashboard and permitted actions. This role-based system reflects a real-world e-commerce workflow.

Goals

- Build a complete MERN stack application with frontend, backend, and database integration.
- Implement role-based dashboards and secure route protection.
- Provide core e-commerce features: cart, wishlist, order placement, and order tracking.
- Provide seller features: add books, manage products, and view orders.
- Provide admin features: manage users, books, orders, and wishlist.

Key Features / Modules

The BookNest platform includes the following key modules:

- Authentication Module (Register/Login with role selection)
- Book Browsing and Book Details Module
- Cart Module (Add to cart, view cart, place order)
- Order Module (User orders, Seller orders, Admin all orders)

- Wishlist Module (Add/Remove wishlist items)
- Feedback Module (User feedback)
- Seller Dashboard (MyProducts, AddBooks, Orders)
- Admin Panel (Dashboard, Users, Books, Orders, Wishlist)

3. Architecture

BookNest follows a modern 3-tier MERN architecture:
• MongoDB (Database Layer)
• Express.js + Node.js (Backend/API Layer)
• React.js (Frontend/Presentation Layer)

This architecture ensures scalability, maintainability, and clear separation of concerns.

Frontend Architecture – React.js

The frontend is built using React.js, enabling dynamic and responsive user interfaces through reusable components and page-based routing.

- React Router DOM is used for navigation between pages (Home, About, Login, Signup, Dashboards).
- Conditional rendering is used to show features based on role (User/Seller/Admin).
- Axios is used to communicate with backend APIs.
- The UI is styled using Bootstrap and custom CSS for a clean, modern layout.

Backend Architecture – Node.js & Express.js

The backend is developed using Node.js and Express.js. It exposes REST APIs for authentication, book management, cart management, wishlist, and order processing.

- Routes are modularized by feature (auth, books, users, cart, orders, wishlist).
- Controllers contain business logic for each API endpoint.

- Middleware verifies JWT tokens and protects routes based on role.
- Environment variables are managed using dotenv.

Database Architecture – MongoDB

MongoDB is used as the NoSQL database for storing users, books, cart, orders, wishlist, and feedback data. MongoDB provides flexible schema design and scalability for web applications.

- Users collection stores user credentials and role information.
- Books collection stores book product details.
- Orders collection stores order details and delivery status.
- Cart and Wishlist collections store user-specific items.

4. Setup Instructions

Prerequisites

Tool	Purpose	Recommended Version
Node.js	JavaScript runtime environment	v18+
npm	Node package manager	v9+
MongoDB (Local)	Database for storing data	Any stable version
Git	Version control and cloning repository	Latest
VS Code	Code editor	Optional
Postman/Thunder Client	Testing backend APIs	Optional

Installation Guide

Step 1: Clone the Project Repository

Commands:git clone https://github.com/your-username/booknest.git
cd booknest

Step 2: Install Dependencies

Install frontend packages:

```
cd client
npm install
```

Install backend packages:

```
cd ../server
npm install
```

Step 3: Set Up Environment Variables

Inside the /server folder, create a .env file and add the following variables:

```
MONGO_URI=mongodb://127.0.0.1:27017/booknest
JWT_SECRET=your_secret_key
PORT=5000
```

Step 4: Run the Application

Start backend server:

```
cd server
npm start
```

Start frontend React app:

```
cd ../client
npm start
```

Once started:

- Frontend runs on: http://localhost:3000
- Backend runs on: http://localhost:5000

5. Folder Structure

Client Folder Structure (React Frontend)

Folder / File	Purpose
/client	Root folder for the React app
/public	Static assets (images, icons, etc.)
/src/components	Reusable UI components (Navbar, Cards, Forms, etc.)
/src/pages	Screens such as Home, About, Login, Signup, Dashboards
App.js	Central router and layout orchestrator
index.js	React entry point
package.json	Frontend dependencies and scripts

Server Folder Structure (Node.js + Express Backend)

Folder / File	Purpose
/server	Root folder for backend
/models	Mongoose schemas (User, Book, Order, Cart, Wishlist, Feedback)
/routes	Express routes grouped by feature
/controllers	Business logic for APIs
/middleware	JWT auth middleware and role checks
server.js	Backend entry point and Express server setup
.env	Environment variables (not committed to GitHub)
package.json	Backend dependencies and scripts

6. Running the Application

Start the frontend:

```
cd client
npm start
```

Start the backend:

```
cd server
npm start
```

Expected URLs:

- Frontend UI: http://localhost:3000
- Backend API: http://localhost:5000/api

7. API Documentation

The BookNest backend exposes REST-style APIs for all core operations. All routes return JSON and follow standard HTTP verbs (GET, POST, PUT, DELETE).

Authentication Routes

Method	Endpoint	Description	Payload
POST	/api/register	Register new user/seller/admin	{ "name", "email", "password", "role" }
POST	/api/login	Login user	{ "email", "password", "role" }

Book Routes

Method	Endpoint	Description	Access
GET	/api/books	Fetch all books	Public
GET	/api/books/:id	Fetch single book	Public
POST	/api/books	Add new book	Seller/Admin
PUT	/api/books/:id	Update book	Seller/Admin
DELETE	/api/books/:id	Delete book	Seller/Admin

Cart Routes

Method	Endpoint	Description
POST	/api/cart/add	Add a book to cart
GET	/api/cart	Get user cart items
DELETE	/api/cart/remove/:id	Remove item from cart

Order Routes

Method	Endpoint	Description	Access
POST	/api/orders	Place an order	User
GET	/api/orders/user	Get logged-in user orders	User
GET	/api/orders/seller	Get seller orders	Seller
GET	/api/orders/all	Get all orders	Admin
PUT	/api/orders/:id/status	Update delivery status	Seller/Admin

Wishlist Routes

Method	Endpoint	Description
POST	/api/wishlist/:bookId	Add/Remove book from wishlist
GET	/api/wishlist	Get wishlist items

User Management Routes (Admin)

Method	Endpoint	Description
GET	/api/admin/users	View all users
DELETE	/api/admin/users/:id	Delete a user

Feedback Routes

Method	Endpoint	Description
POST	/api/feedback	Submit feedback
GET	/api/feedback	View feedback (Admin)

8. Authentication

BookNest uses a secure JWT-based authentication system. It supports role-based access control for Users, Sellers, and Admins.

Authentication Flow

Step	Action	What Happens
1	Register	User submits details. Password is hashed with bcrypt and stored in MongoDB.
2	Login	Server validates credentials and returns a JWT token on success.
3	Protected Requests	Frontend sends token in headers: Authorization: Bearer <token>.

JWT Token Details

Field	Purpose
userId	MongoDB ID of the logged-in user
email	Email address
role	user / seller / admin
iat / exp	Issued-at and expiry timestamps

9. User Interface

The BookNest user interface is designed for clarity, responsiveness, and easy navigation. The UI supports three main roles: User, Seller, and Admin.

UI Screens / Pages

- Home Page: Landing page with welcome banner and navigation.
- About Page: Short description about the platform.
- Login Page: Role-based login form.
- Signup Page: Register account with role selection.
- User Dashboard: Profile, Browse books, Purchase, Feedback, Logout.
- Admin Panel: Dashboard, Users, Books, Orders, Wishlist.
- Seller Panel: MyProducts, Add Books, Orders, Dashboard.
- Users Management Page: Admin can view users and their orders.
- Orders Pages: User orders and admin all orders listing.
- Wishlist Page: Wishlist items displayed as cards.

10. Testing

Manual testing was performed to ensure correct functionality, stability, and role-based access across all modules. API endpoints were tested using Postman/Thunder Client.

Frontend Testing

- Validated Login and Signup forms.
- Verified role-based navigation after login.
- Checked cart, wishlist, and order pages for correct rendering.
- Verified responsive layout using browser dev tools.

Backend Testing

- Tested Register and Login endpoints.
- Verified JWT-protected routes with and without token.
- Tested CRUD operations for books.
- Tested order placement and order status updates.
- Tested admin-only routes such as user management.

11. Screenshots / Demo

The following screenshots demonstrate the key pages and modules of the BookNest platform. Screenshots include Home, About, Login, Signup, User Dashboard, Users Management, Orders, Seller Products, Seller Add Book, Admin Books, Admin Wishlist, Admin Orders, and Seller Dashboard.

12. Known Issues

- No email verification during registration (users can register directly).
- No online payment gateway integration (orders are placed without payment).
- Limited search and filter options for books (basic listing only).
- No pagination (all books load at once).
- No automated testing framework implemented (manual testing only).

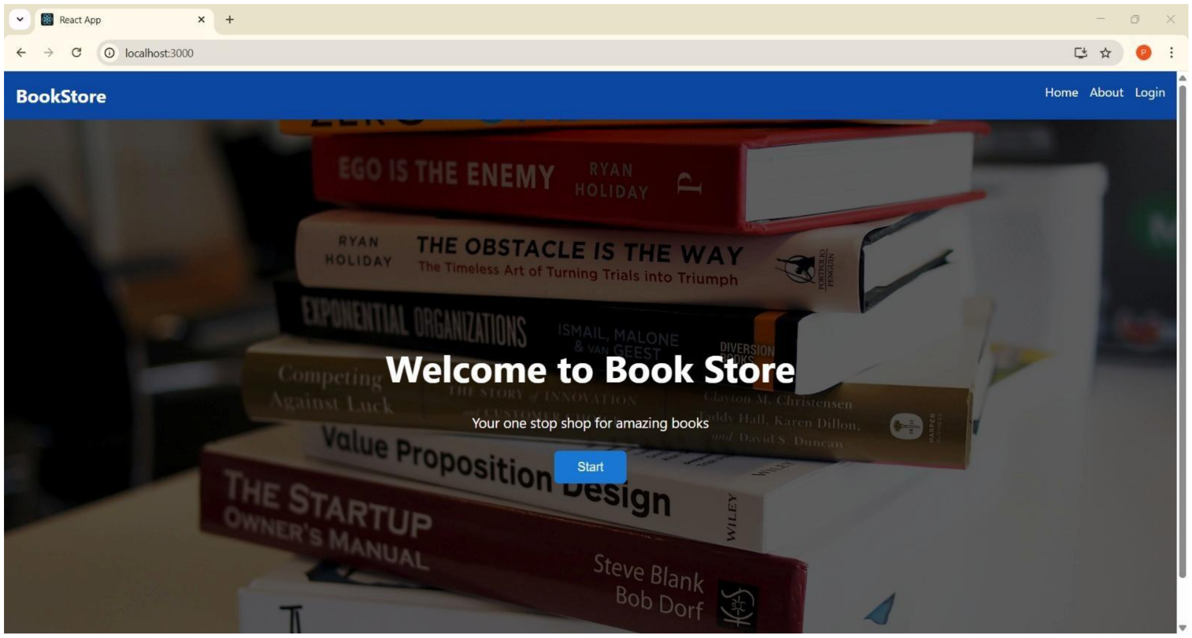
13. Future Enhancements

- Add advanced search and filters (category, price range, author).
- Integrate payment gateway (Razorpay/Stripe).
- Add email verification and forgot password functionality.
- Add product ratings and reviews.
- Implement pagination or infinite scroll for performance.
- Deploy frontend and backend to cloud (Vercel/Netlify + Render/Railway).
- Add analytics dashboards for sellers and admins.

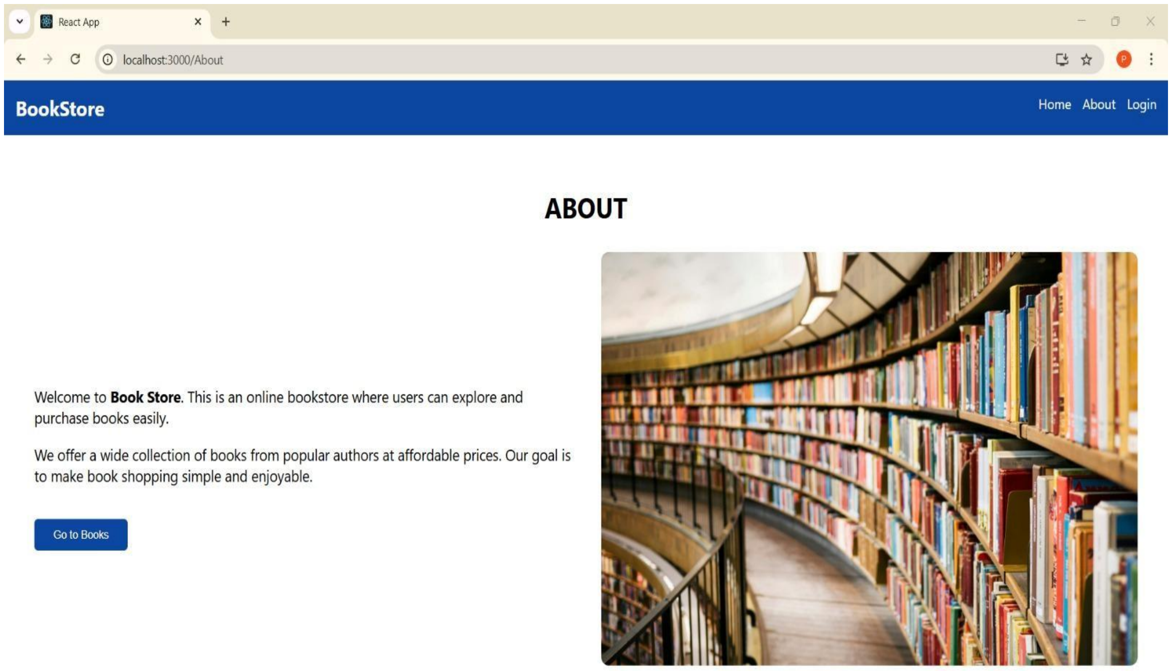
Appendix A: UI Screenshots

-The following screenshots are captured from the running application on localhost. They demonstrate the main pages and role-based dashboards of BookNest.

Home Page (Landing Screen):



About Page:

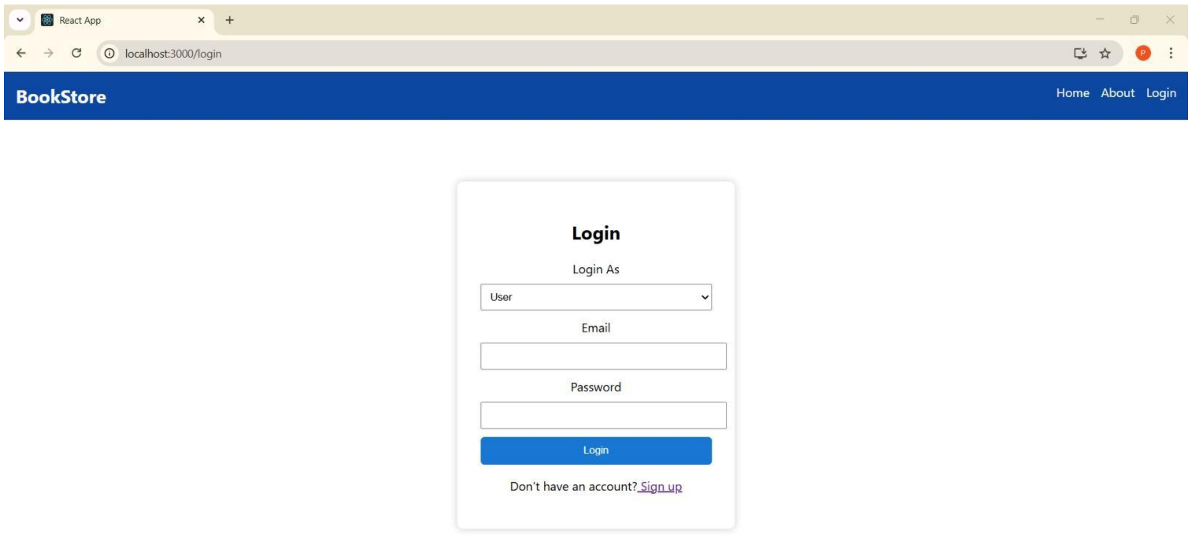


ABOUT

Welcome to **Book Store**. This is an online bookstore where users can explore and purchase books easily.

We offer a wide collection of books from popular authors at affordable prices. Our goal is to make book shopping simple and enjoyable.

Login Page (Role-based Login)



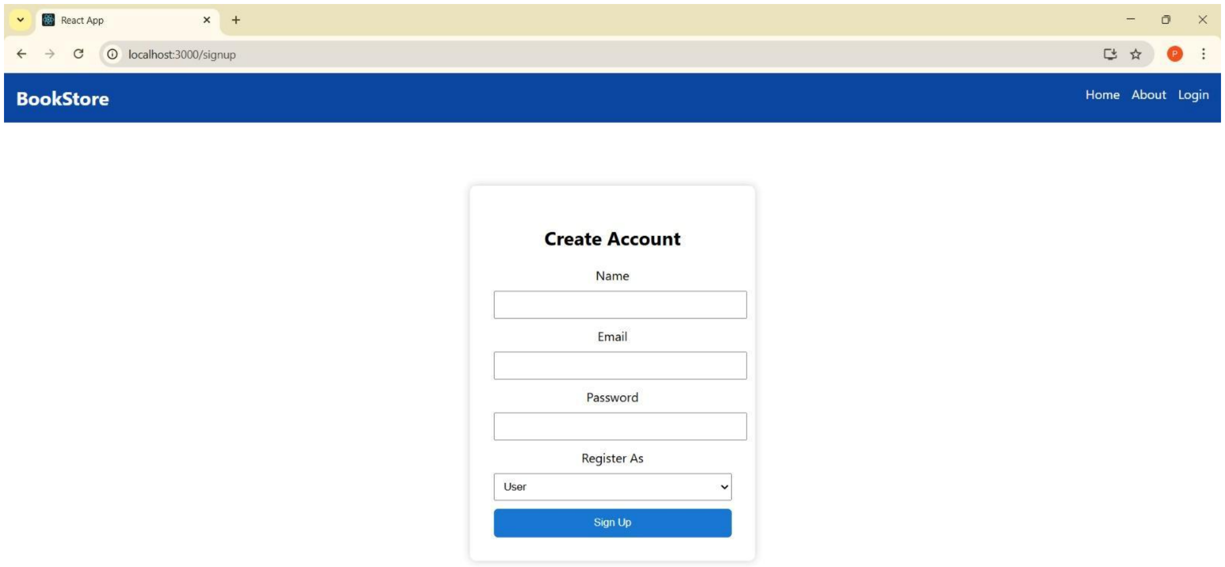
Login

Login As: User

Email

Password

Signup Page (Role-based Registration)



Create Account

Name

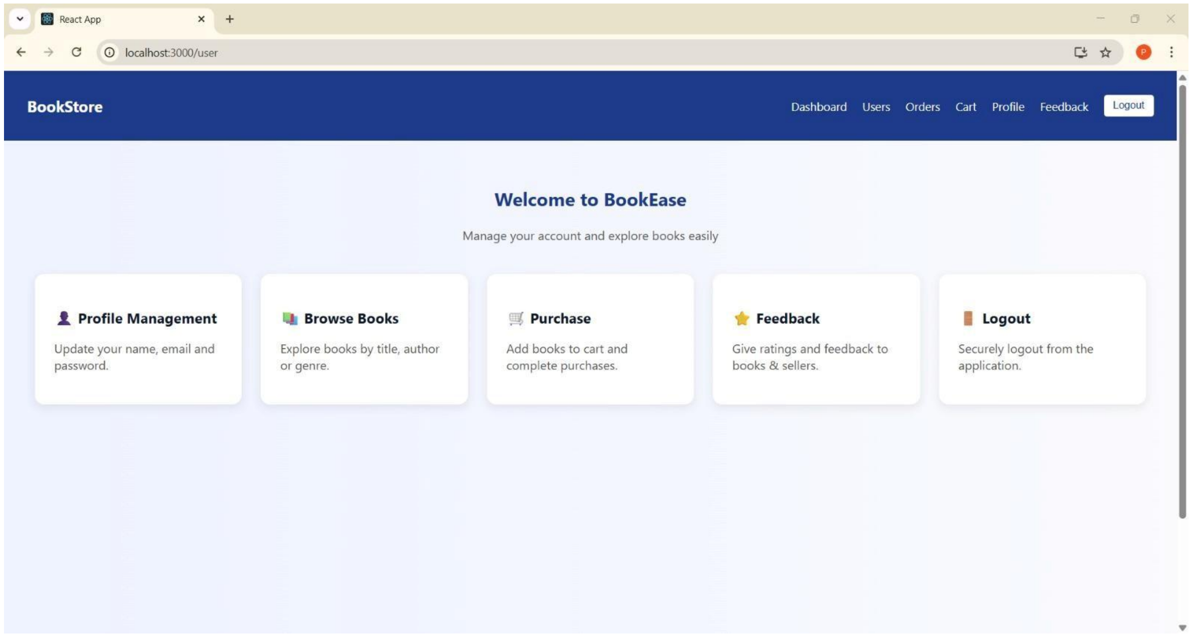
Email

Password

Register As

User

User Dashboard:



Welcome to BookEase

Manage your account and explore books easily

Profile Management
Update your name, email and password.

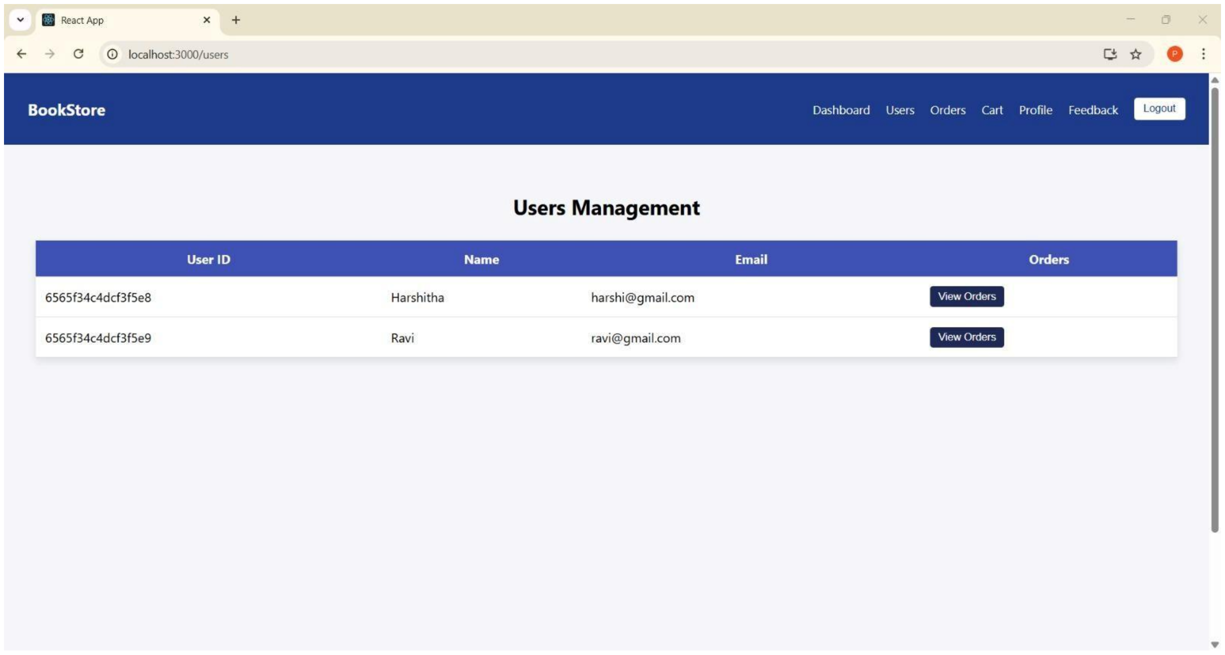
Browse Books
Explore books by title, author or genre.

Purchase
Add books to cart and complete purchases.

Feedback
Give ratings and feedback to books & sellers.

Logout
Securely logout from the application.

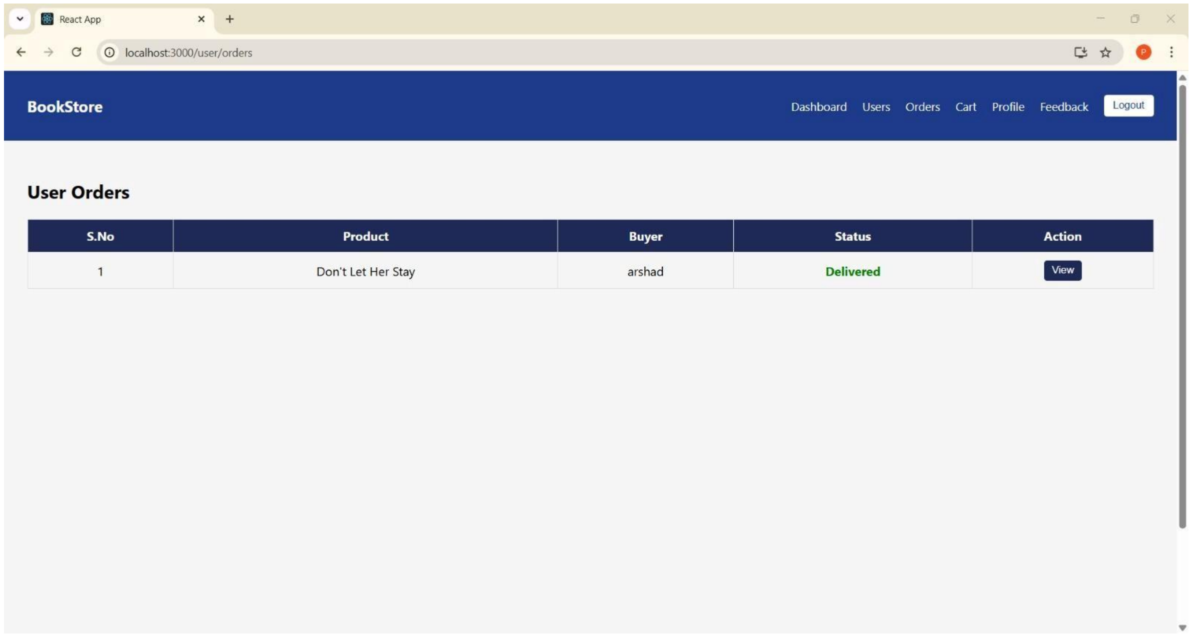
Users Management:



Users Management

User ID	Name	Email	Orders
6565f34c4dcf3f5e8	Harshitha	harshi@gmail.com	View Orders
6565f34c4dcf3f5e9	Ravi	ravi@gmail.com	View Orders

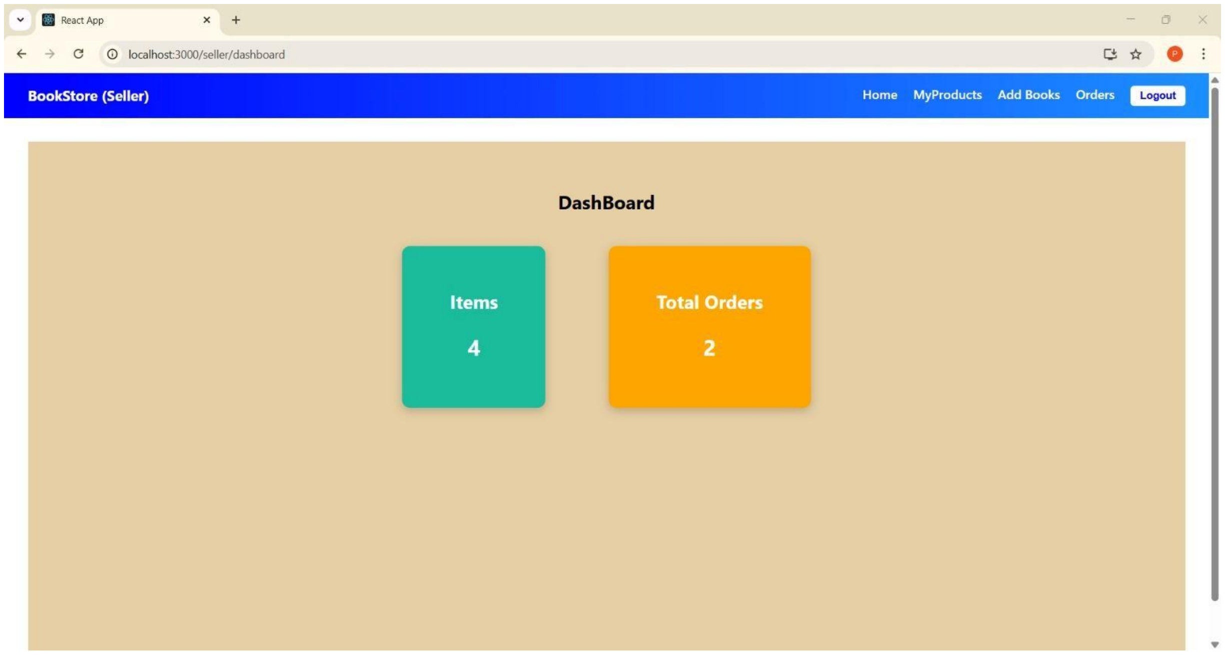
User Orders Page:



User Orders

S.No	Product	Buyer	Status	Action
1	Don't Let Her Stay	arshad	Delivered	View

Seller: Dashboard:



DashBoard

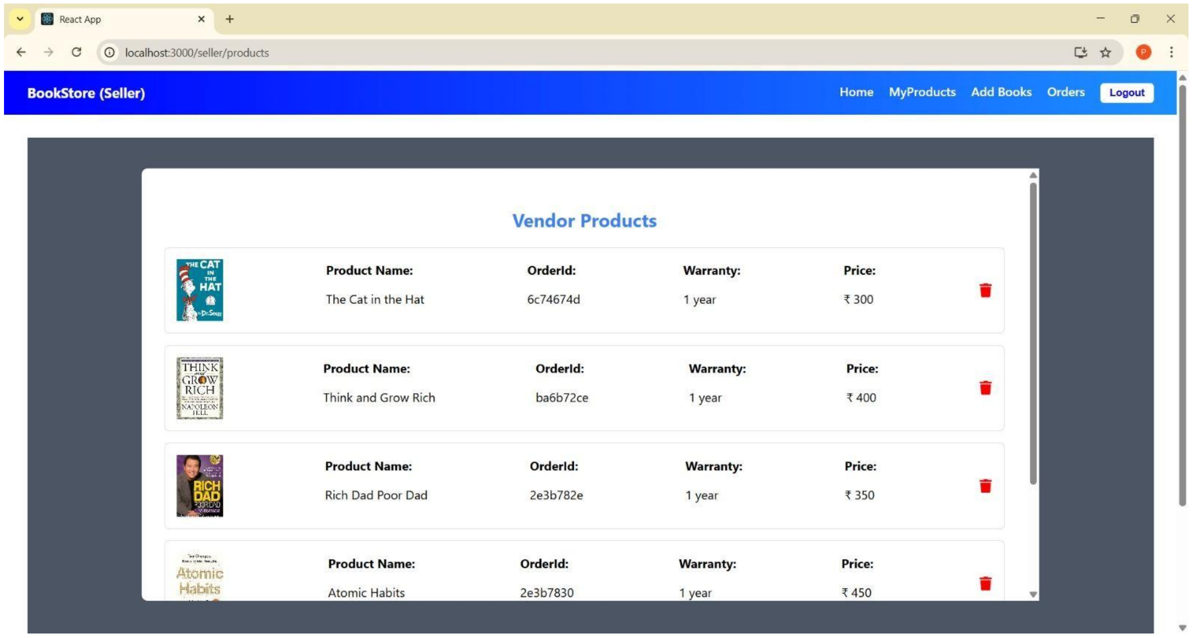
Items

4

Total Orders

2

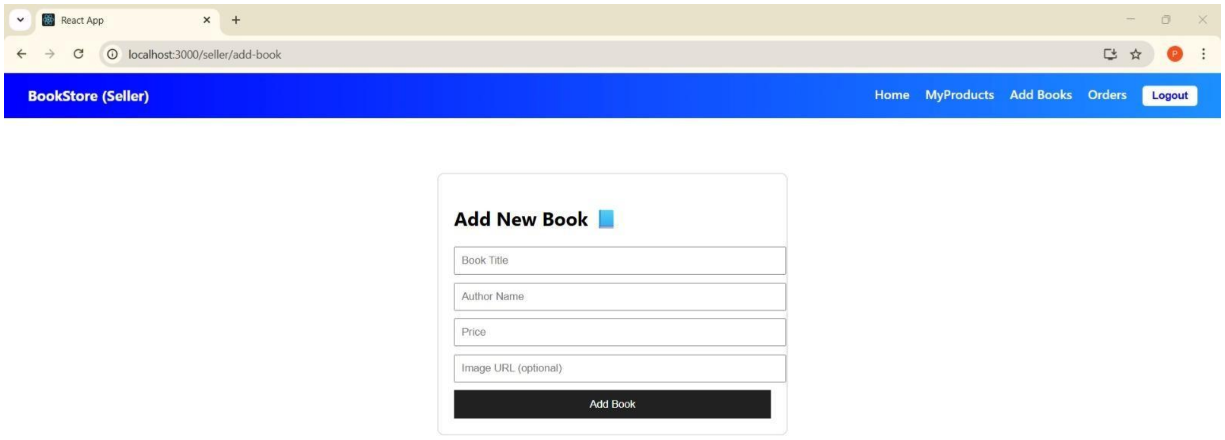
Seller: Vendor Products:



Vendor Products

Product Name:	OrderId:	Warranty:	Price:
The Cat in the Hat	6c74674d	1 year	₹ 300
Think and Grow Rich	ba6b72ce	1 year	₹ 400
Rich Dad Poor Dad	2e3b782e	1 year	₹ 350
Atomic Habits	2e3b7830	1 year	₹ 450

Seller:Add Books:



React App

localhost:3000/seller/add-book

BookStore (Seller)

Home MyProducts Add Books Orders Logout

Add New Book 📘

Book Title

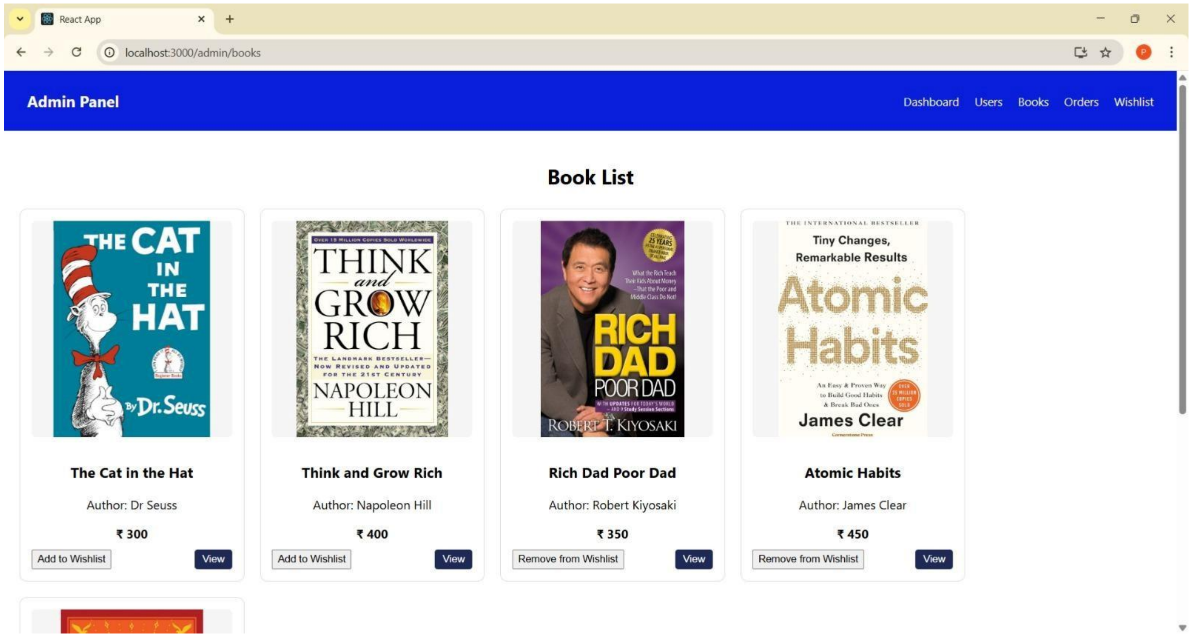
Author Name

Price

Image URL (optional)

Add Book

Admin: BookList:



React App

localhost:3000/admin/books

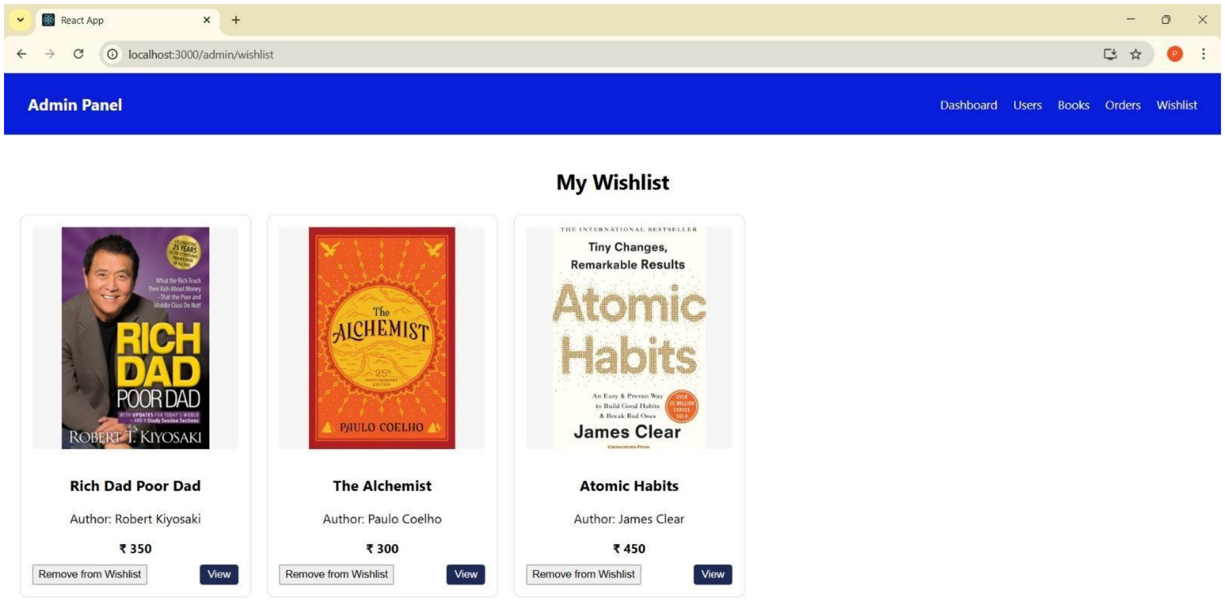
Admin Panel

Dashboard Users Books Orders Wishlist

Book List

The Cat in the Hat	Think and Grow Rich	Rich Dad Poor Dad	Atomic Habits
Author: Dr Seuss	Author: Napoleon Hill	Author: Robert Kiyosaki	Author: James Clear
₹ 300	₹ 400	₹ 350	₹ 450
Add to Wishlist View	Add to Wishlist View	Remove from Wishlist View	Remove from Wishlist View

Admin: Wishlist


React App
localhost:3000/admin/wishlist
Admin Panel
Dashboard Users Books Orders Wishlist

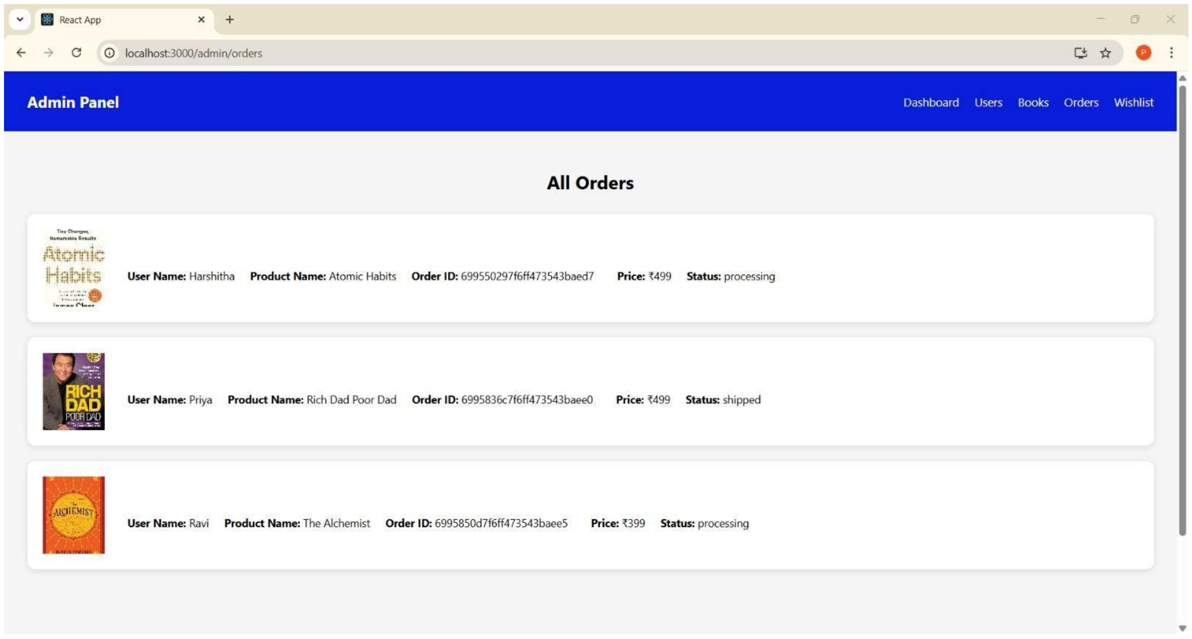
My Wishlist

Rich Dad Poor Dad
Author: Robert Kiyosaki
₹ 350
Remove from Wishlist View

The Alchemist
Author: Paulo Coelho
₹ 300
Remove from Wishlist View

Atomic Habits
Author: James Clear
₹ 450
Remove from Wishlist View

Admin:Orders:


React App
localhost:3000/admin/orders
Admin Panel
Dashboard Users Books Orders Wishlist

All Orders

User Name: Harshitha **Product Name:** Atomic Habits **Order ID:** 699550297f6ff473543baed7 **Price:** ₹499 **Status:** processing

User Name: Priya **Product Name:** Rich Dad Poor Dad **Order ID:** 6995836c7f6ff473543baee0 **Price:** ₹499 **Status:** shipped

User Name: Ravi **Product Name:** The Alchemist **Order ID:** 6995850d7f6ff473543baee5 **Price:** ₹399 **Status:** processing