BOOK STORE

Team Members:

1. Devanshu Kawad – Backend Developer
2. Devendraa J Sheth – Backend Developer
3. Dhanasingh R – UI/UX Design
4. Vineet A - Frontend
5. Vishaal M- Tester

1.PROJECT OVERVIEW:

Purpose:

The BookStore Application aims to provide a seamless, online platform for book enthusiasts to browse, discover, and purchase books. By leveraging the MERN (MongoDB, Express.js, React, Node.js) stack, the application offers an intuitive, secure, and responsive experience for users to explore a wide variety of books, manage their orders, and enjoy a personalized shopping journey. The goal is to replicate the engaging experience of a physical bookstore while making it accessible from anywhere, at any time.

Goals:

Simplify book discovery through advanced search and filtering.

Provide an easy to navigate UI for browsing and purchasing books.

Enable user registration, secure authentication, and order management.

Implement a smooth, scalable backend for managing books, orders, and inventory.

2. FEATURES

User Registration and Authentication:

Users can sign up, log in, and authenticate their identity securely.

Book Listings & Search:

Display a wide selection of books with sorting and filtering options (by genre, author, price, ratings).

Book Selection:

Allows users to view detailed information about each book, including author, description, price, and ratings.

Shopping Cart and Checkout:

Users can add books to the cart, adjust quantities, and complete purchases securely.

Order Management:

Includes order tracking, history, and the ability to rate books and leave reviews.

Responsive Design:

The application is fully responsive, ensuring an optimal experience across devices (desktop, tablet, mobile).

3. ARCHITECTURE

Frontend (React)

Components: The frontend is developed using React.js, which includes reusable components like book listings, book details, shopping cart, and user profile.

State Management: React's internal state management is used, with tools like Context API or Redux to handle global states (e.g., cart items, user authentication status).

Routing: React Router is used for handling navigation between pages (e.g., home, book details, order history).

Backend (Node.js and Express.js)

Express.js Server: The backend is powered by Node.js and Express.js, which serve the necessary APIs to handle client requests, including fetching book data, processing orders, and handling user authentication.

RESTful API: The backend exposes RESTful APIs for:

Managing books (CRUD operations)

Managing user data (signup, login, authentication)

Handling cart and order processes.

Database (MongoDB)

Schema: The database is designed using MongoDB, which stores:

Books: Title, author, genre, description, price, ratings, and availability.

Users: User profile information, order history, authentication credentials (hashed passwords).

Orders: Order details, payment status, shipping information, and order history.

Relationships:

A User can have multiple Orders.

Each Order can contain multiple Books.

Interactions:

The backend queries and updates the MongoDB database for fetching books, placing orders, and managing user profiles. MongoDB's documentbased structure allows flexibility in storing book metadata and user information.

4. SETUP INSTRUCTIONS

Prerequisites

Node.js (v14 or above)

MongoDB (Local or Remote instance, e.g., MongoDB Atlas)

npm (Node package manager)

Installation

1. Clone the repository:

```bash
git clone https://github.com/vin2004/BookStore.git
```

2. Backend Setup:

Navigate to the `server` directory:

```bash
cd server
```

Install dependencies:

```bash
npm install
```

Start the backend server:

```bash
npm run dev
```

3. Frontend Setup:

Navigate to the `client` directory:

```bash
cd client
```

Install dependencies:

bash

npm install

Start the frontend server:

bash

npm start

5. FOLDER STRUCTURE

Client (Frontend)

`client/`

`src/`

`components/`: React shared components Footer, Home, etc.

`User/`: Different pages like for user Home, Cart, Order History.

`Admin/`: Different pages like for admin Seller, login, users.

`Seller/`: Different pages like for seller add book, my products.

`public/`: Static assets like images, icons.

`App.js`: Main component for routing and rendering the app.

`index.js`: Entry point for React app.

Server (Backend)

`server/`

`db/`: Mongoose models for Admin, User, and Seller schemas.

`routes/`: API route definitions for bookrelated and userrelated requests.

`config.js`: Configuration files for database connection, environment variables.

`server.js`: Main entry point for setting up Express and connecting to MongoDB that has controllers, Services .

6. RUNNING THE APPLICATION

Frontend:

Navigate to the `client` directory and run:

```bash

npm start
```

```

Backend:

Navigate to the `server` directory and run:

```bash

npm start

```

Ensure MongoDB is running locally or use a cloudbased MongoDB instance (e.g., MongoDB Atlas).

7. API DOCUMENTATION

Admin APIs

Admin Login

POST /alogin

Body: { "email": "admin@example.com", "password": "password" }

Response: { "Status": "Success", "user": { "id": "adminId", "name": "Admin Name", "email": "admin@example.com" } }

Admin Register

POST /asignup

Body: { "name": "Admin Name", "email": "admin@example.com", "password": "password" }

Response: "Account Created"

User APIs

User Login

POST /login

Body: { "email": "user@example.com", "password": "password" }
```

Response: { "Status": "Success", "user": { "id": "userId", "name": "User Name", "email": "user@example.com" } }

User Register

POST /signup

Body: { "name": "User Name", "email": "user@example.com", "password": "password" }

Response: "Account Created"

Get All Users

GET /users

Response: [{ "id": "userId", "name": "User Name", "email": "user@example.com" }, ...]

Delete User

DELETE /userdelete/:id

Response:

Status 200: User deleted successfully

Status 500: Internal server error

Seller APIs

Seller Login

POST /slogin

Body: { "email": "seller@example.com", "password": "password" }

Response: { "Status": "Success", "user": { "id": "sellerId", "name": "Seller Name", "email": "seller@example.com" } }

Seller Register

POST /ssignup

Body: { "name": "Seller Name", "email": "seller@example.com", "password": "password" }

Response: "Account Created"

Item APIs

Add Item

POST /items

Body: { "title": "Item Title", "author": "Author Name", "genre": "Genre", "description": "Description", "price": 100, "userId": "userId", "userName": "User Name", "itemImage": "imagePath" }

Response: { "id": "itemId", "title": "Item Title", "author": "Author Name", "genre": "Genre", "description": "Description", "price": 100, "userId": "userId", "userName": "User Name", "itemImage": "imagePath" }

Get Items by User ID

GET /getitem/:userId

Response: [{ "id": "itemId", "title": "Item Title", ... }, ...]

Delete Item

DELETE /itemdelete/:id

Response:

Status 200: Item deleted successfully

Status 500: Internal server error

Create Order

POST /userorder

Body: { "flatno": "Flat No", "city": "City", "state": "State", "pincode": "Pincode", "totalamount": 100, "seller": "Seller Name", "sellerId": "sellerId", "BookingDate": "2023-01-01", "description": "Order Description", "Delivery": "Delivery Info", "userId": "userId", "userName": "User Name", "booktitle": "Book Title", "bookauthor": "Book Author", "bookgenre": "Book Genre", "itemImage": "imagePath" }

Response: { "id": "orderId", "flatno": "Flat No", "city": "City", "state": "State", "pincode": "Pincode", "totalamount": 100, "seller": "Seller Name", "sellerId": "sellerId", "BookingDate": "2023-01-01", "description": "Order Description", "Delivery": "Delivery Info", "userId": "userId", "userName": "User Name", "booktitle": "Book Title", "bookauthor": "Book Author", "bookgenre": "Book Genre", "itemImage": "imagePath" }

Get Orders by User ID

GET /getorders/:userId

Response: [{ "id": "orderId", "flatno": "Flat No", "city": "City", "state": "State", "pincode": "Pincode", "totalamount": 100, "seller": "Seller Name", "BookingDate": "2023-01-01", "description": "Order Description", "Delivery": "Delivery Info", "userId": "userId", "userName": "User Name", "booktitle": "Book Title", "bookauthor": "Book Author", "bookgenre": "Book Genre", "itemImage": "imagePath" }, ...]

Get All Orders

GET /orders

Response: [{ "id": "orderId", "flatno": "Flat No", "city": "City", "state": "State", "pincode": "Pincode", "totalamount": 100, "seller": "Seller Name", "BookingDate": "2023-01-01", "description": "Order Description", "Delivery": "Delivery Info", "userId": "userId", "userName": "User Name", "booktitle": "Book Title", "bookauthor": "Book Author", "bookgenre": "Book Genre", "itemImage": "imagePath" }, ...]

Wishlist APIs

Get All Wishlist Items

GET /wishlist

Response: [{ "itemId": "itemId", "title": "Item Title", "itemImage": "imagePath", "userId": "userId", "userName": "User Name" }, ...]

Get Wishlist by User ID

GET /wishlist/:userId

Response: [{ "itemId": "itemId", "title": "Item Title", "itemImage": "imagePath", "userId": "userId", "userName": "User Name" }, ...]

Add Item to Wishlist

POST /wishlist/add

Body: { "itemId": "itemId", "title": "Item Title", "itemImage": "imagePath", "userId": "userId", "userName": "User Name" }

Response: { "itemId": "itemId", "title": "Item Title", "itemImage": "imagePath", "userId": "userId", "userName": "User Name" }

Remove Item from Wishlist

POST /wishlist/remove

Body: { "itemId": "itemId" }

Response: { "msg": "Item removed from wishlist" }

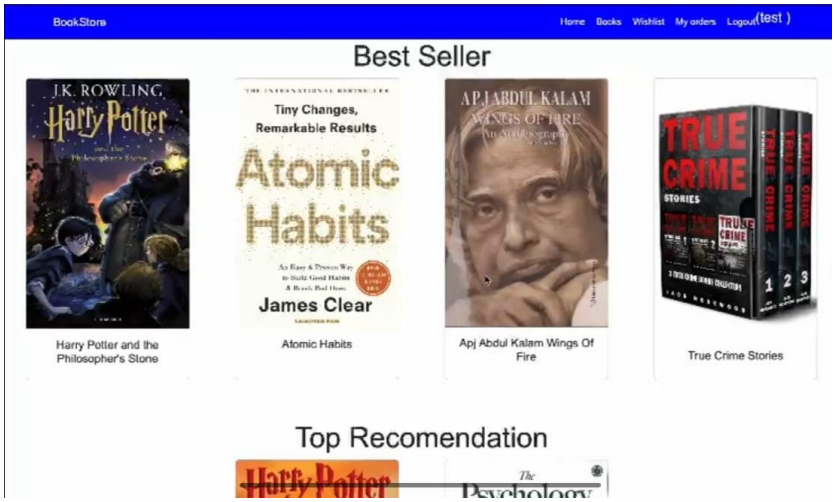
8. AUTHENTICATION

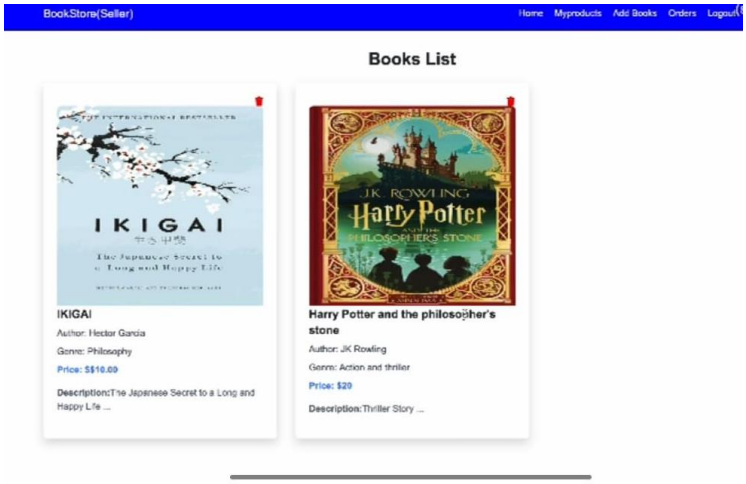
User and Admin Authentication: The code implements authentication through login and registration endpoints for both users and admins, verifying credentials against the database and ensuring unique email addresses during registration.

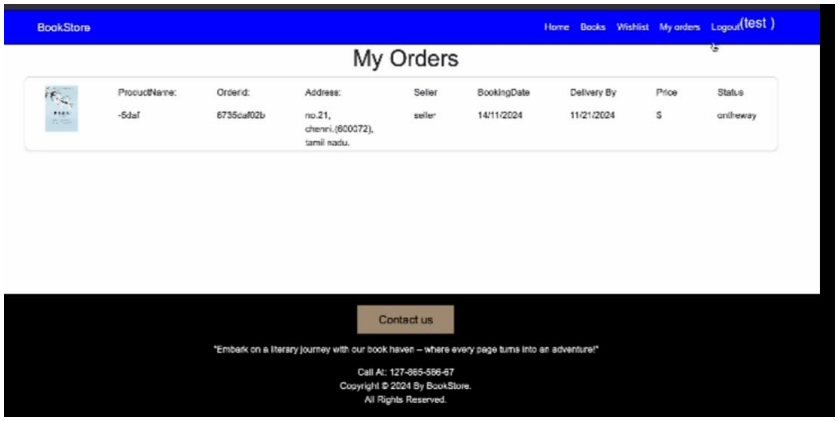
Secure Access Management: Successful authentication provides users with access to their respective accounts, while preventing unauthorized access by validating identities and managing user sessions effectively.

9. USER INTERFACE

Screenshots:


BookStore
Home Books Wishlist My orders Logout(test)
Best Seller
Harry Potter and the Philosopher's Stone
Atomic Habits
Apj Abdul Kalam Wings Of Fire
True Crime Stories
Top Recomendation


BookStore(Seller)
Home Myproducts Add Books Orders Logout(
Books List
IKIGAI
Author: Hector Garcia
Genre: Philosophy
Price: $$10.00
Description:The Japanese Secret to a Long and Happy Life ...
Harry Potter and the philosopher's stone
Author: JK Rowling
Genre: Action and thriller
Price: $20
Description:Thriller Story ...


BookStore
Home Books Wishlist My orders Logout(test)
My Orders
ProductName: -5daf
OrderId: 6735caf02b
Address: no.21, chennai,(600072), tamil nadu.
Seller seller
BookingDate 14/11/2024
Delivery By 11/21/2024
Price $
Status onitheway
Contact us
"Embark on a literary journey with our book haven – where every page turns into an adventure!"
Call At: 127-865-586-67
Copyright © 2024 By BookStore.
All Rights Reserved.

10. TESTING

Tools Used:

Manual testing for verifying application functionality.

User experience testing to ensure usability.

Testing Strategy:

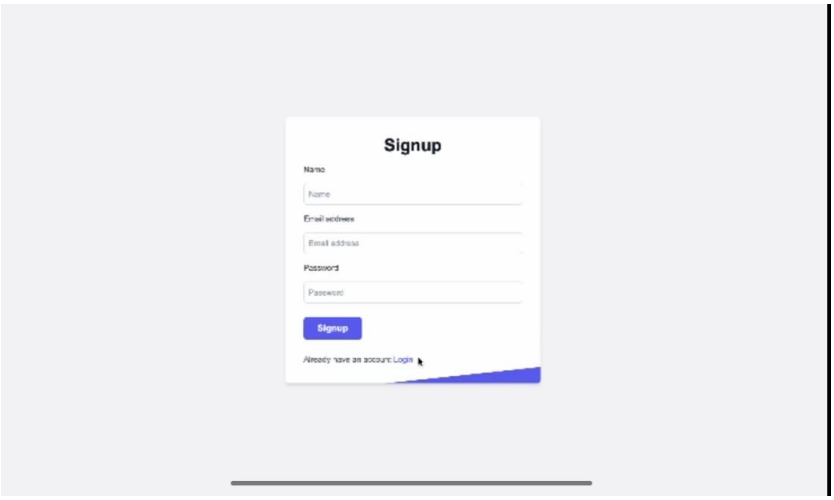
Tests are written to validate:

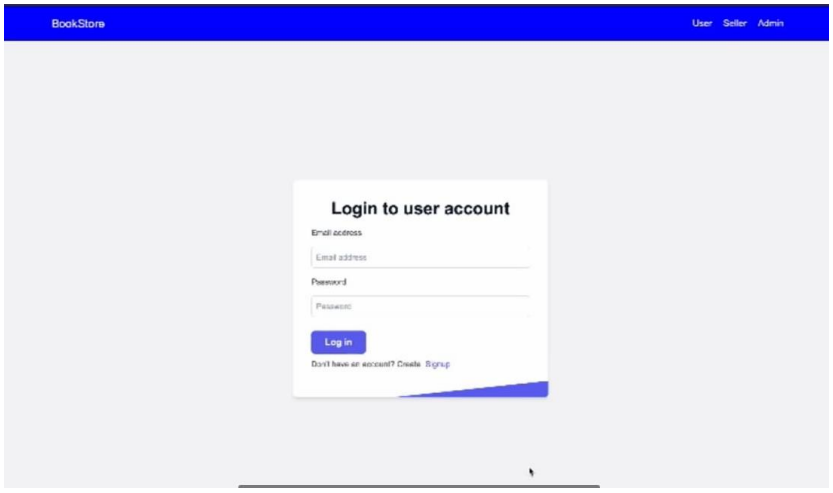
API responses and error handling.

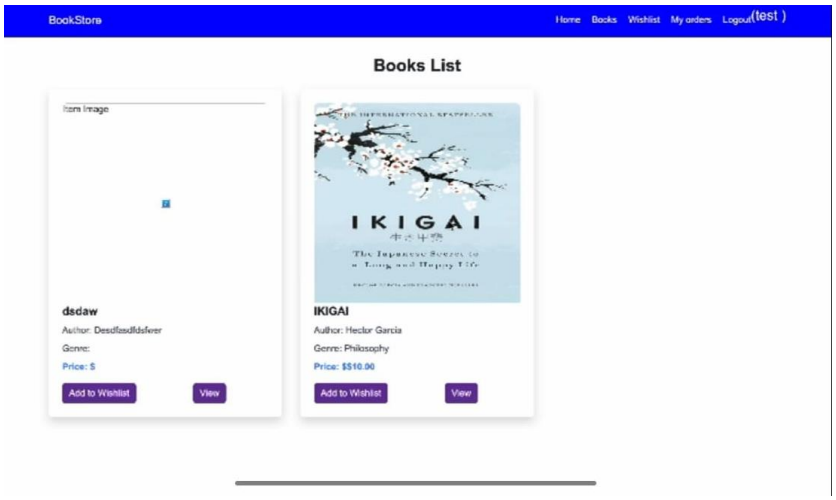
User authentication and authorization.

Frontend component rendering and behaviour.

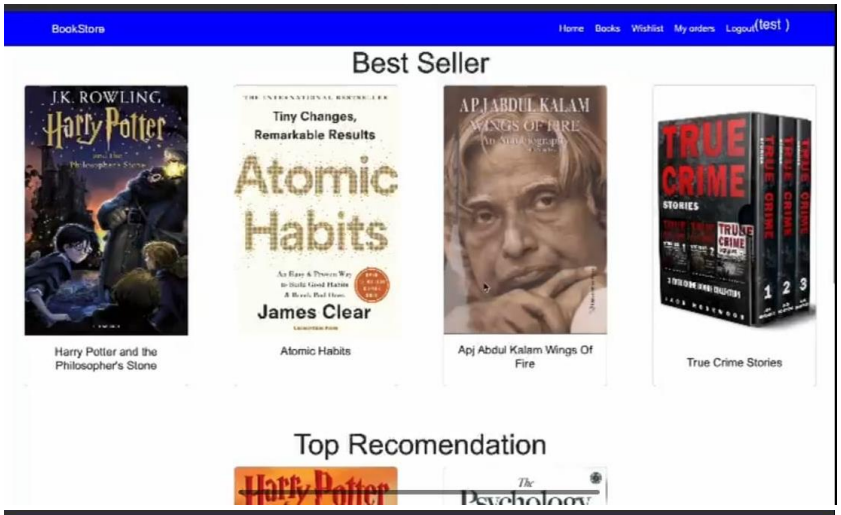
11. SCREENSHOTS OR DEMO

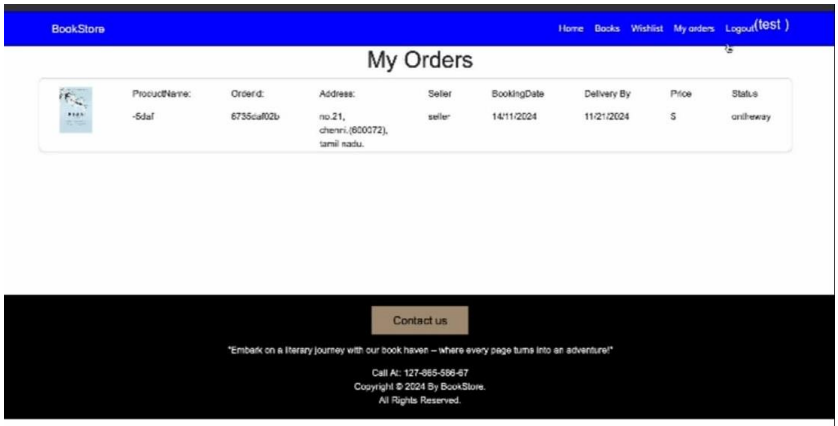


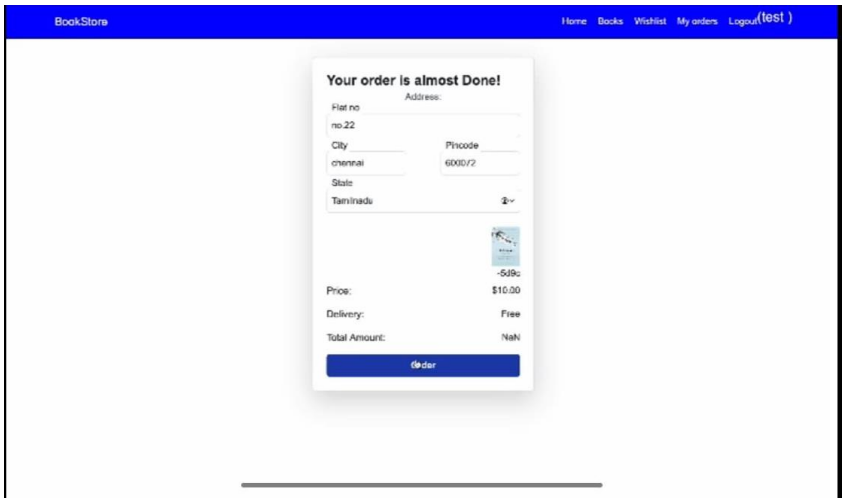

BookStore
User Seller Admin
Login to user account
Email address
Email address
Password
Password
Log in
Don't have an account? Create Signup

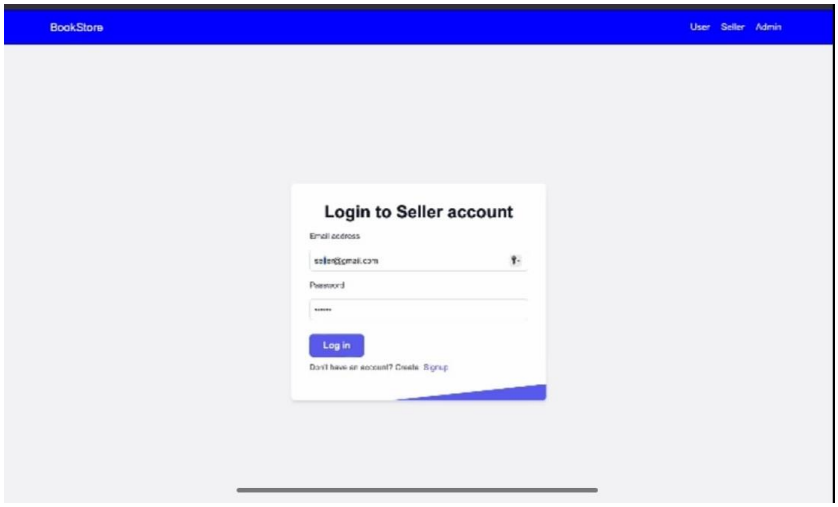

BookStore
Home Books Wishlist My orders Logout(test)
Books List
Item Image
dsdaw
Author: Desdfasdfdsfeer
Genre:
Price: $
Add to Wishlist
View
IKIGAI
Author: Hector Garcia
Genre: Philosophy
Price: $$10.00
Add to Wishlist
View

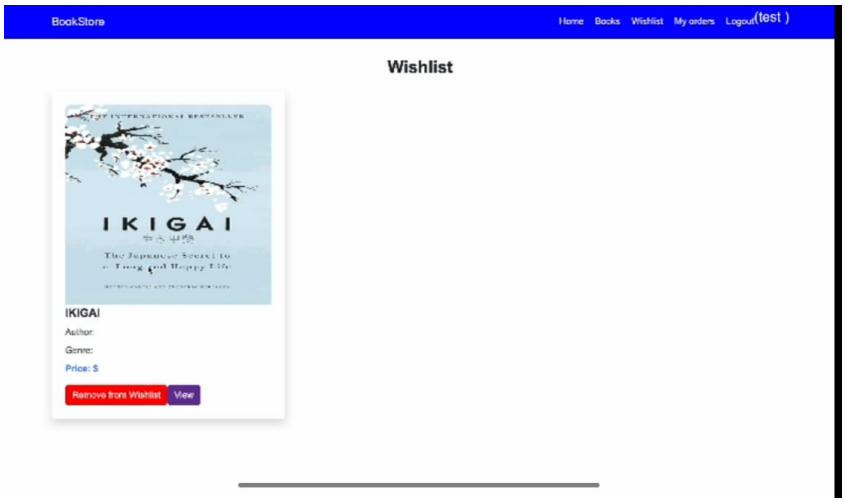

BookStore
Home Books Wishlist My orders Logout(test)
IKIGAI
-5d9c
Description
The Japanese Secret to a Long and Happy Life
Info
Title: IKIGAI
Author: Hector Garcia
Genre: Philosophy
Price: $10.00

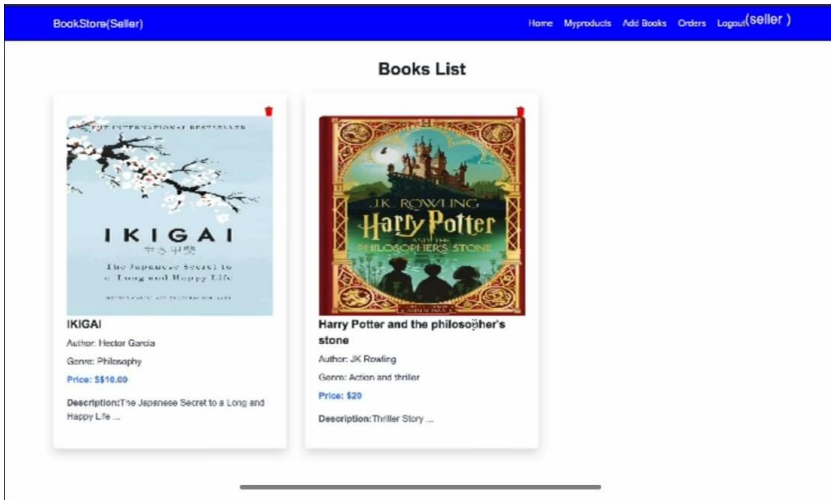

BookStore
Home Books Wishlist My orders Logout(test)
Best Seller
Harry Potter and the Philosopher's Stone
Atomic Habits
Apj Abdul Kalam Wings Of Fire
True Crime Stories
Top Recomendation

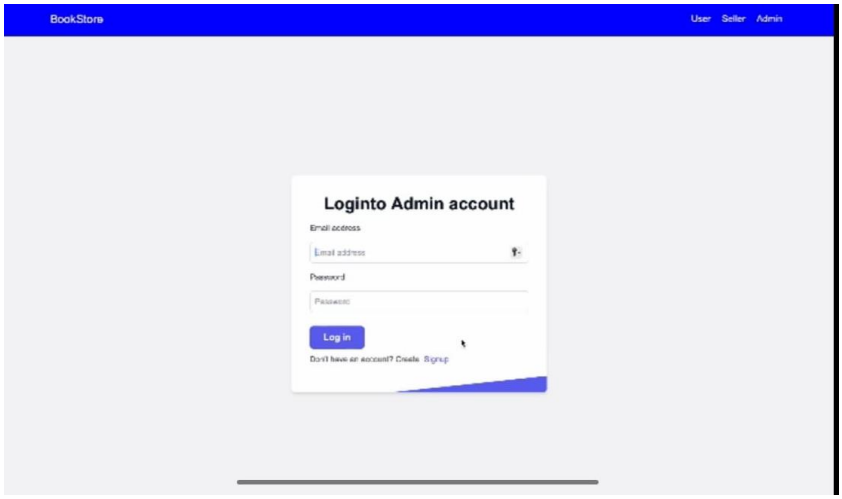

BookStore
Home Books Wishlist My orders Logout(test)
My Orders
ProcuctName: -5daf
Orderd: 6735daf02b
Address: no.21, chenni,(600072), tamil nadu.
Seller seller
BookingDate 14/11/2024
Delivery By 11/21/2024
Price $
Status ontheway
Contact us
"Embark on a literary journey with our book haven – where every page turns into an adventure!"
Call At: 127-865-586-67
Copyright © 2024 By BookStore.
All Rights Reserved.

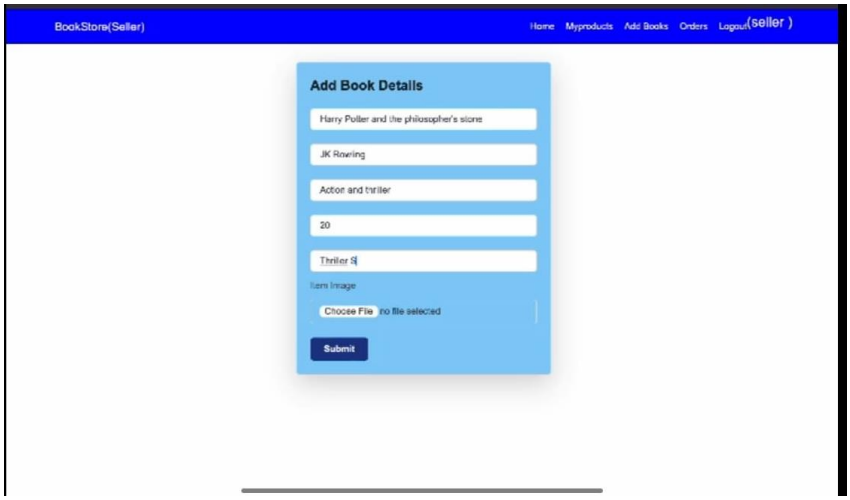

BookStore
Home Books Wishlist My orders Logout(test)
Your order is almost Done!
Address:
Flat no
no.22
City
chennai
Pincode
600072
State
Tamilnadu
-5d9c
Price: $10.00
Delivery: Free
Total Amount: NaN
Order


BookStore
User
Seller
Admin
Login to Seller account
Email address
seller@gmail.com
Password
Log in
Don't have an account? Create Signup


BookStore
Home
Books
Wishlist
My orders
Logout(test)
Wishlist
IKIGAI
IKIGAI
Author:
Genre:
Price: $
Remove from Wishlist
View


BookStore(Seller)
Home
Myproducts
Add Books
Orders
Logout(seller)
Books List
IKIGAI
IKIGAI
Author: Hector Garcia
Genre: Philosophy
Price: S$10.00
Description:The Japanese Secret to a Long and Happy Life ...
J.K. ROWLING
Harry Potter
AND THE PHILOSOPHER'S STONE
Harry Potter and the philosopher's stone
Author: JK Rowling
Genre: Action and thriller
Price: $20
Description:Thriller Story ...


BookStore
User
Seller
Admin
Loginto Admin account
Email address
Email address
Password
Password
Log in
Don't have an account? Create Signup


BookStore(Seller)
Home
Myproducts
Add Books
Orders
Logout(seller)
Add Book Details
Harry Potter and the philosopher's stone
JK Rowling
Action and thriller
20
Thriller S
Item Image
Choose File
no file selected
Submit

12. KNOWN ISSUES

Issue 1: Payment gateway integration may not be fully functional (pending third-party service setup).

Issue 2: Mobile view may have small layout issues with long book descriptions.

13. FUTURE ENHANCEMENTS

Recommendation System: Implement a book recommendation engine based on user preferences and reading history.

Admin Panel: Allow admins to add, edit, or delete books from the catalogue.

Social Login: Enable login with Google, Facebook, or other social platforms for easier registration.

Reviews and Ratings: Implement a more sophisticated review system with comments, images, and ratings for books.