

# Full Stack Development with MERN

## Project Documentation format

### 1. Introduction

- **Project Title:** “BookEase: A Book Store Application”
- **Team Members:**
  - Adelin Brianna A (au311121104001)
  - Amala Swetah Louis Kishore (au311121104005)
  - Leena Bency A (au311121104032)
  - Piyo Licira J (au311121104044)

### 2. Project Overview

- **Purpose:** BookEase provides readers with an intuitive platform to explore and buy books online. It bridges the gap between digital convenience and the charm of traditional book shopping by offering curated recommendations, personalized services, and secure transactions.
- **Features:**
  - ★ **User Accounts:** Allow users to register, log in, and manage their profiles.
  - ★ **Search Functionality:** Users can search for books by title, author, genre, or keyword.
  - ★ **Cart System:** Enables users to add books to their cart for easy checkout.
  - ★ **Order Management:** Tracks the purchase history and provides status updates for orders.
  - ★ **Admin Dashboard:** Allows admin users to manage inventory, view sales analytics, and handle customer queries.

### 3. Architecture

- **Frontend:**
  - ★ **Technology Used:** React.js for building reusable components and dynamic UI.
  - ★ **State Management:** React Context API or Redux for managing application state.
  - ★ **Routing:** React Router for navigating between pages (e.g., Home, Book Details, Cart, etc.).
  - ★ **Styling:** Styled-components or CSS frameworks like Bootstrap or Material-UI
- **Backend:**
  - ★ **Node.js:** JavaScript runtime for server-side logic.
  - ★ **Express.js:** Lightweight framework to create RESTful APIs.
  - ★ **Middleware:**
    - Authentication middleware (JWT verification).
    - Error-handling middleware to catch and display errors gracefully.

- **Database:**

#### **Schema Design:**

**Users Collection:** Stores user profiles, hashed passwords, and purchase history.

**Books Collection:** Contains book details such as title, author, price, stock, and genre.

**Orders Collection:** Tracks user orders with timestamps and status.

#### **Third-Party Tools and Libraries:**

- **Frontend:** Axios for API calls, React Query for caching data.
- **Backend:** Mongoose for MongoDB interaction, bcrypt for password hashing.
- **Others:** Payment gateways like Stripe for secure transactions.

## **4. Setup Instructions**

- **Prerequisites:**
  - **Node.js and npm:** Ensure you have Node.js (v16+) installed.
  - **MongoDB:** Set up a local or cloud-based MongoDB database.
  - **Git:** Required for version control and cloning the repository.
- **Installation:** Step-by-step guide to clone, install dependencies, and set up the environment variables.

### **1) Clone the repository:**

```
git clone https://github.com/your-repo/bookease.git
```

```
cd bookease
```

### **2) Backend Setup:**

i) Navigate to the server directory.

ii) Install dependencies:

```
npm install
```

iii) Create `.env` file for environment variables:

```
PORT=5000
```

```
MONGO_URI=<your_mongodb_connection_string>
```

```
JWT_SECRET=<your_jwt_secret_key>
```

iv) Start the backend server:

```
npm start
```

### 3) Frontend Setup:

i) Navigate to the `client` directory.

ii) Install dependencies:

**`npm install`**

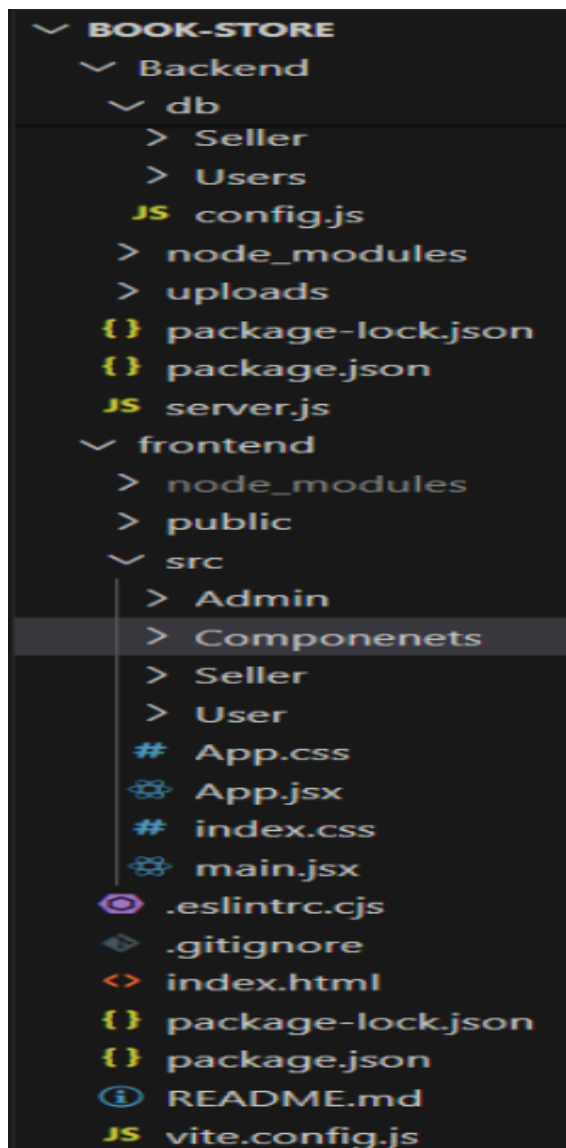
iii) Create `.env` file:

**`REACT_APP_API_URL=http://localhost:5000/api`**

iv) Start the frontend development server:

**`npm start`**

## 5. Folder Structure



## 6. Running the Application

- Run **backend**:  
`cd server`  
`npm start`
- Run **frontend**:  
`cd client`  
`npm start`
- Open the application in your browser at **`http://localhost:3000`**.

## 7. API Documentation

**POST /api/users/login**

**Request Body:json**

```
{  
  
  "email": "user@example.com",  
  
  "password": "password123"  
}
```

**Response:json**

```
{  
  
  "token": "jwt_token",  
  
  "user": {  
  
    "id": "user_id",  
  
    "name": "John Doe"  
  
  }  
}
```

## 8. Authentication

**Signup:** Securely stores hashed passwords.

**Login:** Issues a JWT for authenticated sessions.

**Middleware:** Protects routes like **`/api/orders`** and **`/api/books`**.

## 9. User Interface

**Homepage:** Featured books and recommendations.

**Book Details Page:** Information about a specific book with an "Add to Cart" button.

**Cart Page:** Displays items added for checkout.

**Profile Page:** Allows users to view and edit account details.

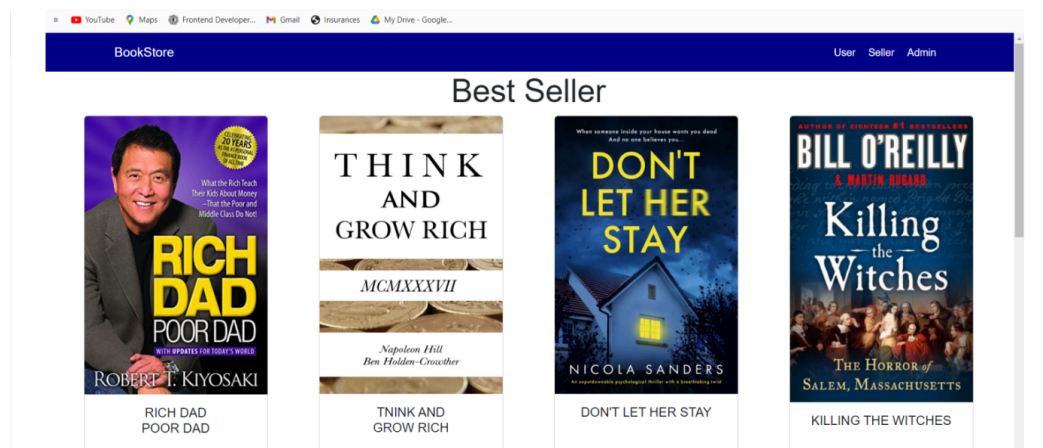
## 10. Testing

**Unit Testing:**

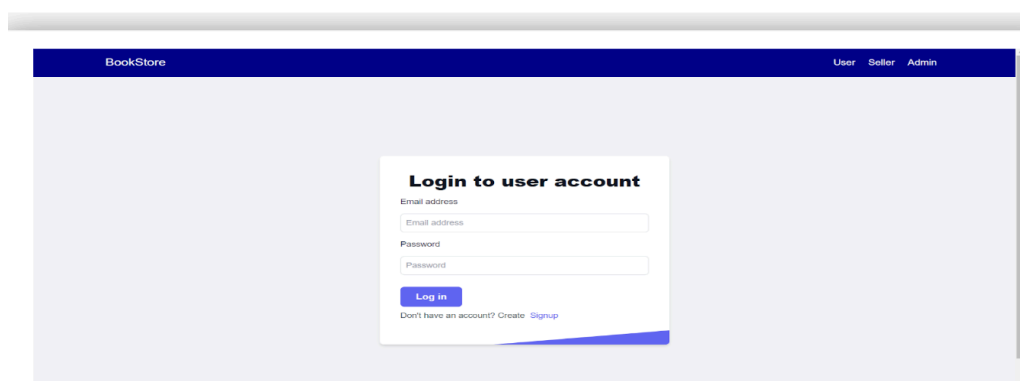
- Use Jest for testing React components and business logic.
- Use Postman for API testing (GET, POST, PUT, DELETE requests).

## 11. Screenshots or Demo

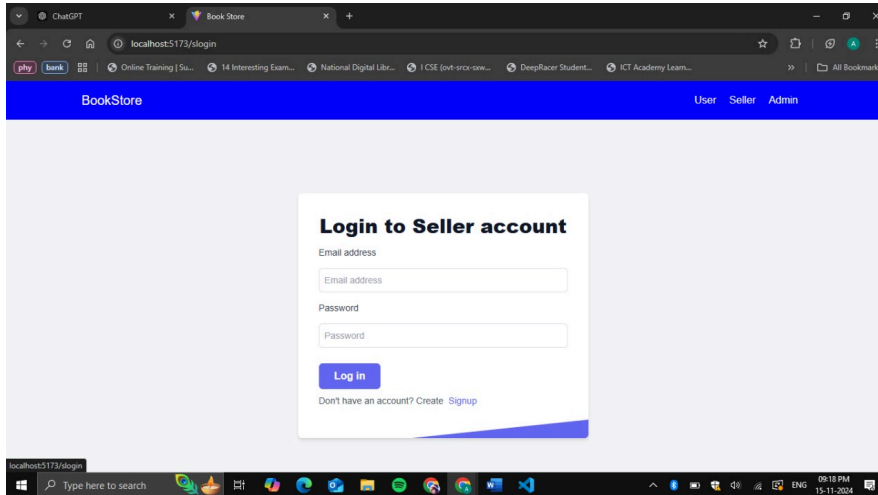
**Landing Page :**



**User's Login Page :**

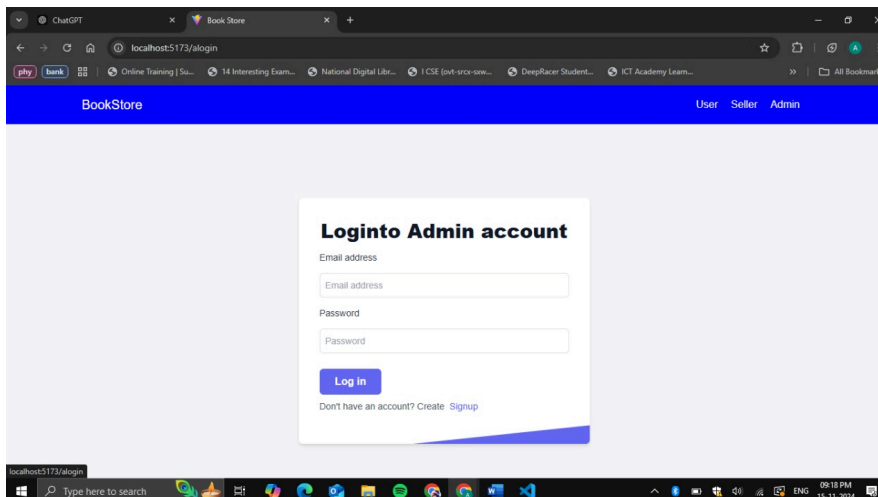


## Seller's Login Page :



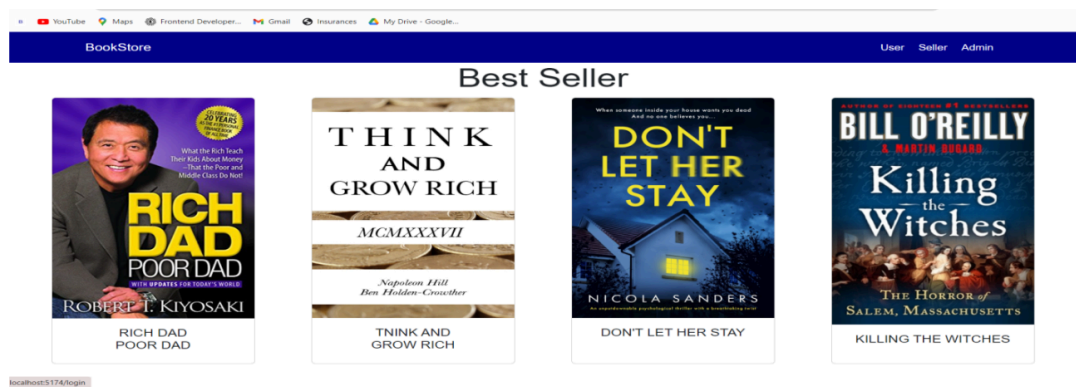
The screenshot shows a web browser window with the URL `localhost:5173/login`. The page has a blue header with the text "BookStore" and navigation links "User", "Seller", and "Admin". The main content area is light blue and features a white login form titled "Login to Seller account". The form includes two input fields: "Email address" and "Password", followed by a blue "Log in" button. Below the button, there is a link that says "Don't have an account? Create [Signup](#)". The browser's taskbar at the bottom shows the time as 09:18 PM on 15-11-2024.

## Admin's Login Page :



The screenshot shows a web browser window with the URL `localhost:5173/login`. The page has a blue header with the text "BookStore" and navigation links "User", "Seller", and "Admin". The main content area is light blue and features a white login form titled "Loginto Admin account". The form includes two input fields: "Email address" and "Password", followed by a blue "Log in" button. Below the button, there is a link that says "Don't have an account? Create [Signup](#)". The browser's taskbar at the bottom shows the time as 09:18 PM on 15-11-2024.

## Home Page :




Books Page :

BookStore

HomeBooksWishlistMy ordersLogout(admin )

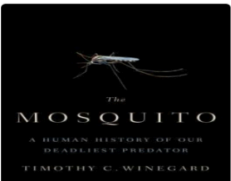
Books List



**Think and Grow Rich**  
MCMXXXVII  
Napoleon Hill  
*Ben Holden-Croscher*

**Think and Grow Rich**  
Author: Napoleon Hill  
Genre: Business  
Price: \$100


Add to WishlistView



**The Mosquito**  
A HUMAN HISTORY OF OUR DEADLIEST PREDATOR  
TIMOTHY C. WINEGARD

**The Mosquito**  
Author: Timothy C. Winegard  
Genre: Nonfiction  
Price: \$130

Remove from WishlistView



**Elon Musk**  
by Walter Isaacson

**Elon Musk**  
Author: Walter Isaacson  
Genre: Biography  
Price: \$130

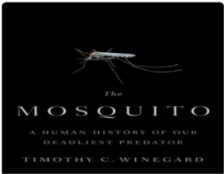
Add to WishlistView

WishList Page :

BookStore

HomeBooksWishlistMy ordersLogout(admin )

Wishlist



**The Mosquito**  
A HUMAN HISTORY OF OUR DEADLIEST PREDATOR  
TIMOTHY C. WINEGARD

**The Mosquito**  
Author:  
Genre:  
Price: \$


Remove from WishlistView

My Bookings Page :

BookStore

HomeBooksWishlistMy ordersLogout(syed )

My Orders



ProductName: -0449

OrderId: 6580448015

Address: dsfsd, asdas, (fada), asdasda.


Seller: syed

BookingDate: 18/12/2023

Delivery By: 12/25/2023

Price: \$199

Status: delivered



ProductName: -011d

OrderId: 660011467

Address: 122-B, hyderabad,(517994), Telangana.


Seller: syed

BookingDate: 25/3/2024

Delivery By: 4/1/2024

Price: \$229

Status: ontheway



ProductName: -025

OrderId: 660025067

Address: . , .


Seller: syed

BookingDate: 25/3/2024

Delivery By: 4/1/2024

Price: \$229

Status: ontheway



ProductName: -025

OrderId: 660025667

Address: . , .

Seller: syed

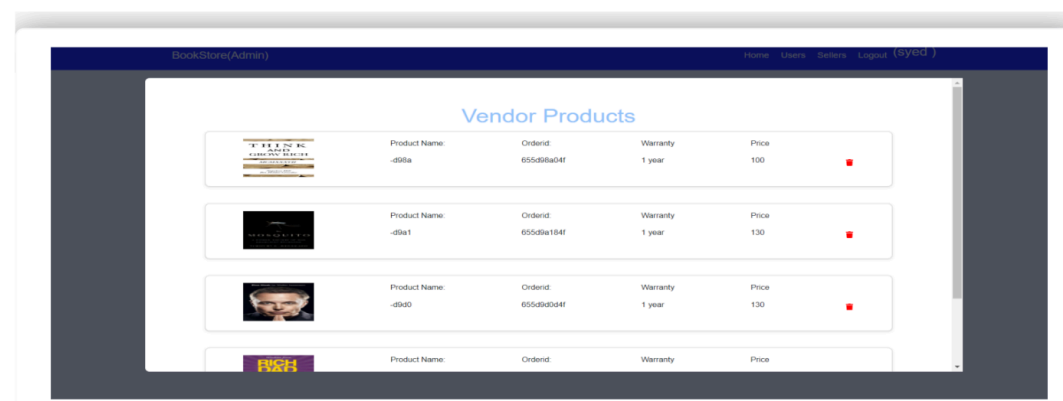
BookingDate: 25/3/2024

Delivery By: 4/1/2024

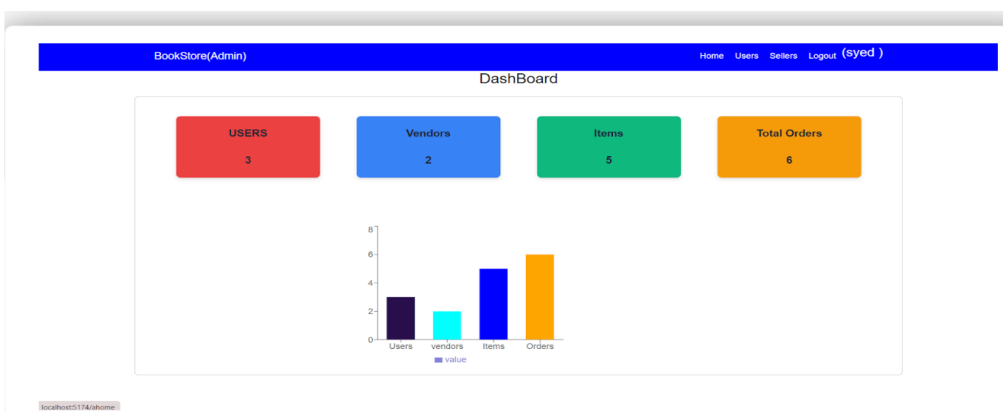
Price: \$229

Status: ontheway

Seller Items :



Admin Dashboard :

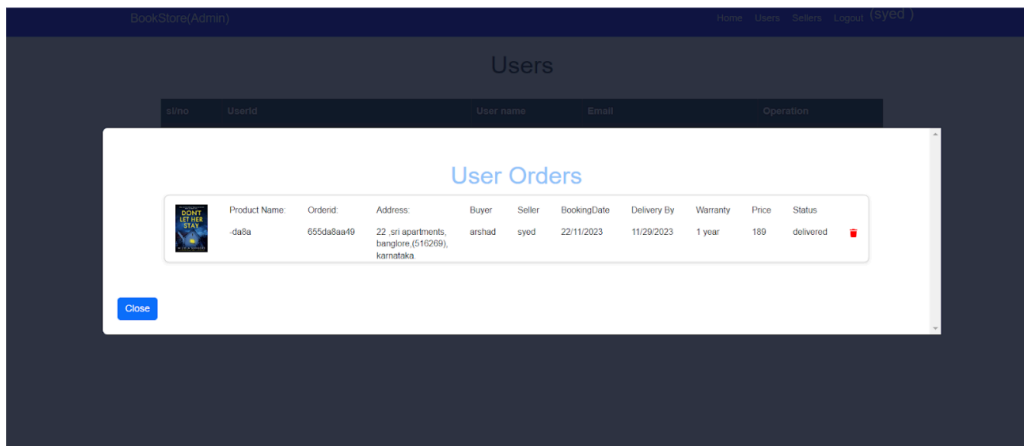


Users Page :

BookStore(Admin)					Home	Users	Sellers	Logout (syed )
Users								
s/no	Userid	User name	Email	Operation				
1	650d9f4c4f4ace5f198b9abf	arshad	arshad@gmail.com	<a href="#">view</a>				
2	650e571f62e8144c8a9cb277	shivani	shivani@gmail.com	<a href="#">view</a>				
3	6500442915b2ba8ff503a669	syed	syed@gmail.com	<a href="#">view</a>				



## User Orders :



## Sellers Page :

BookStore(Admin)

HomeUsersSellersLogout (syed )

Vendors

sl/no	Userid	User name	Email	Operation
1	655da3154992a4960b06489	elf	elf@gmail.com	<div><div></div><div></div><div>view</div></div>
2	655c4b32b451e85b2daade96	syed	syed@gmail.com	<div><div></div><div></div><div>view</div></div>

## 12. Known Issues

- Pagination may be slow for a large number of books.
- Sorting functionality does not persist across sessions.

## 13. Future Enhancements

### ★ Features:

- Add wishlist functionality.
- Enable real-time chat for customer support.

### ★ Performance:

- Implement caching mechanisms using Redis.
- Optimize database queries with indexes.