

# Full Stack Development with MERN

## 1. Introduction

- **Project Title:** BookNest-Where Stories Nestle
- **Team Members:** Jakke VeeraShankar [Team Leader]  
Dulla SivaKalyani[Team member]  
Duggana MahaLakshmi[Team member]  
Battivalasala Sri Lalitha devi[Team member]

## 2. Project Overview

- **Purpose:** The purpose of the BookNest application is to redefine the traditional reading experience by offering a digital platform where users can explore, discover, and purchase books with ease. The app is tailored for modern readers who prefer a seamless, online experience that doesn't compromise on the joy of browsing and collecting literature.
- **Key Features:**
  - Users can securely register and log in to the platform using their email and password. The system ensures data privacy by encrypting passwords and using token-based authentication to manage sessions. This feature allows users to safely access personalized features like order history and saved preferences..
  - The application provides a comprehensive listing of books available for purchase. Each book entry displays relevant details such as the title, author, genre, description, price, rating, and current stock status. This helps users make informed decisions while browsing the catalog..
  - Books can be added to a shopping cart with specified quantities. Once ready, users can proceed through a secure checkout process to finalize their purchases. This triggers backend operations to generate an order and update the inventory accordingly.
  - After a successful transaction, users receive an order confirmation that includes a unique order ID, the list of books purchased, the total cost, and estimated delivery information. This assures users that their order was successfully placed.
  - Users can view a detailed history of all past and current orders from their profile. Each order record includes purchase details and shipment tracking (if integrated). Users can also review books and rate their shopping experience, contributing to community feedback and recommendations..
  - Admins can manage books, users, and orders through a secure backend interface. They can add/edit/delete book entries, monitor sales, and view analytics.

## 3. System Architecture

- **Client Layer (Frontend)**  
**Technology:** React.js  
**Users:** Readers (Customers), Admin

**Role:**

- Interactive book browsing and purchasing.
- Responsive UI for all devices.
- Axios-based API calls.
- User session management with local storage.

- **Server Layer (Backend API)**

**Technology:** Node.js + Express.js

**Responsibilities:**

- API handling for authentication, book inventory, and orders.
- Middleware for validation and security.
- Separate routes for books, users, and orders.

- **API Gateway**

**Technology:** Node.js with Express.js (or a dedicated gateway service like NGINX or API Gateway libraries for microservices)

**Function:** Acts as a single entry point for all incoming client requests, helping to decouple the frontend from individual backend services.

- **Authentication Service**

**Technology:** MongoDB

**Role:**

- Registers new users and stores hashed passwords.
- Authenticates login attempts using bcrypt and issues JWT tokens.
- Verifies user identity for protected routes.
- Maintains session validity via tokens, enabling secure, stateless access to user

## 4. Setup Instructions

- **Install Prerequisites** : Node.js, npm, MongoDB (local/Atlas)
- **CloneRepository** : git clone <https://github.com/your-username/booknest.git>
- **Start MongoDB** : mongod or connect to MongoDB Atlas
- **Setup Backend:** cd server npm install npm start
- **Setup Frontend:** cd client npm install npm start
- **Frontend API:** <http://localhost:3000>
- **Backend API:** <http://localhost:5000>

## 5. Folder Structure

**Client (React Frontend) :**

/client

/public  
/src  
  /components  
  /pages  
  /context  
  /styles  
App.jsx  
package.json

**Server (Node.js + Express Backend):**

/server  
  /models  
  /routes  
  /controllers  
index.js  
package.json

## 6. Running the Application

- **Frontend:** cd client  
    npm start  
    # Runs on <http://localhost:3000>
- **Backend:** cd server  
    npm start  
    # Runs on <http://localhost:5000>

## 7. API Documentation

- **User APIs:**
  - POST /api/regist   Register new user
  - POST /api/login – Authenticate user
- **Book APIs:**
  - GET /api/books – Fetch all books
  - GET /api/books/:id – Fetch single book details
  - POST /api/books – (Admin) Add new book
  - PUT /api/books/:id – (Admin) Update book
  - DELETE /api/books/:id – (Admin) Delete book
- **Order APIs:**
  - POST /api/orders – Place an order
  - GET /api/orders – View all orders for user
  - GET /api/orders/:id – View order details

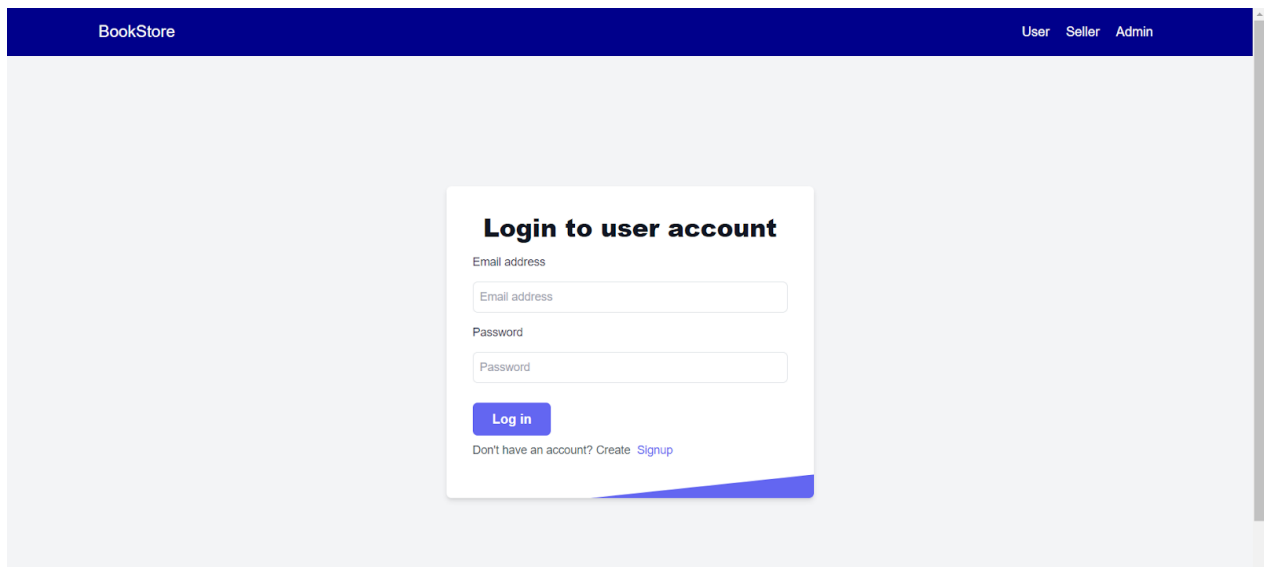
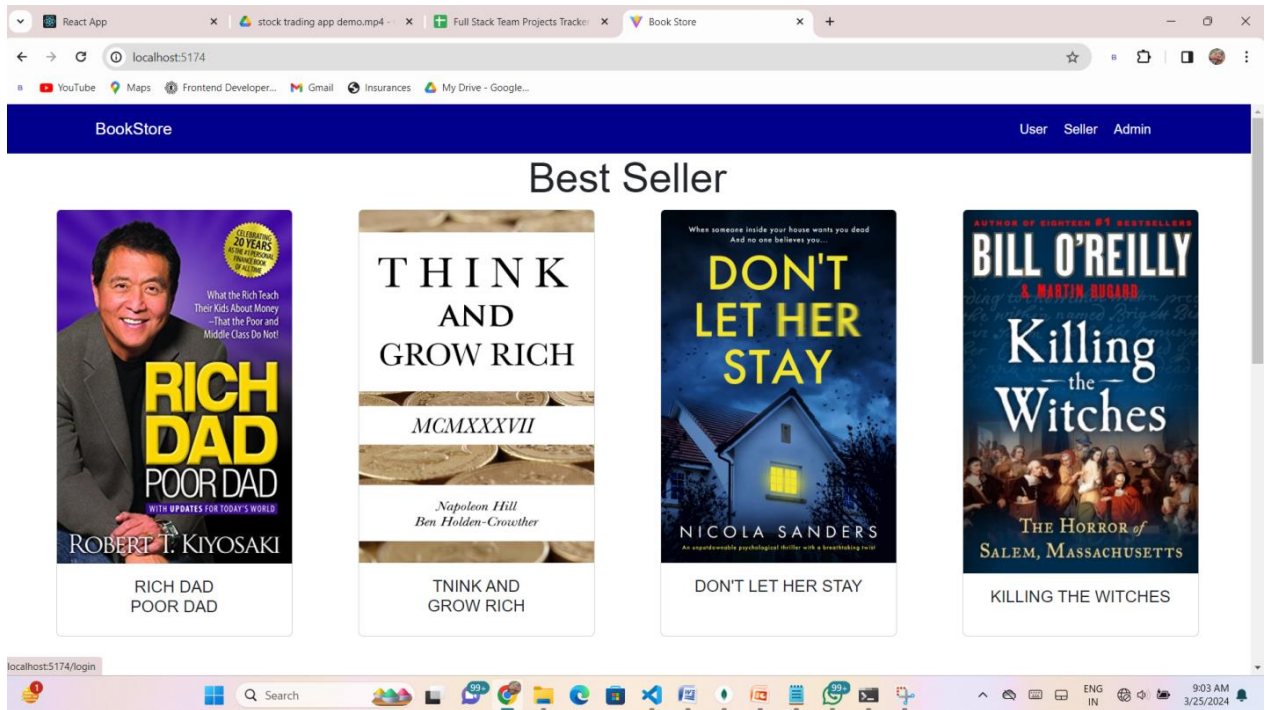
## 8. Authentication Flow

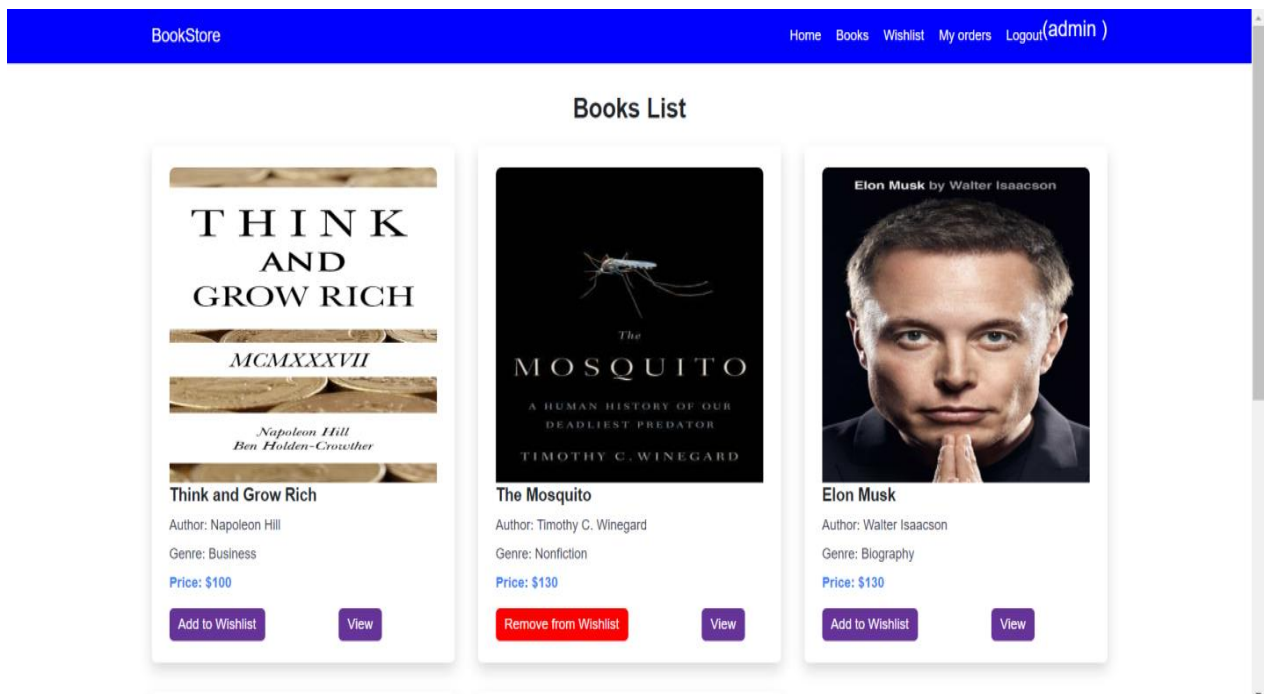
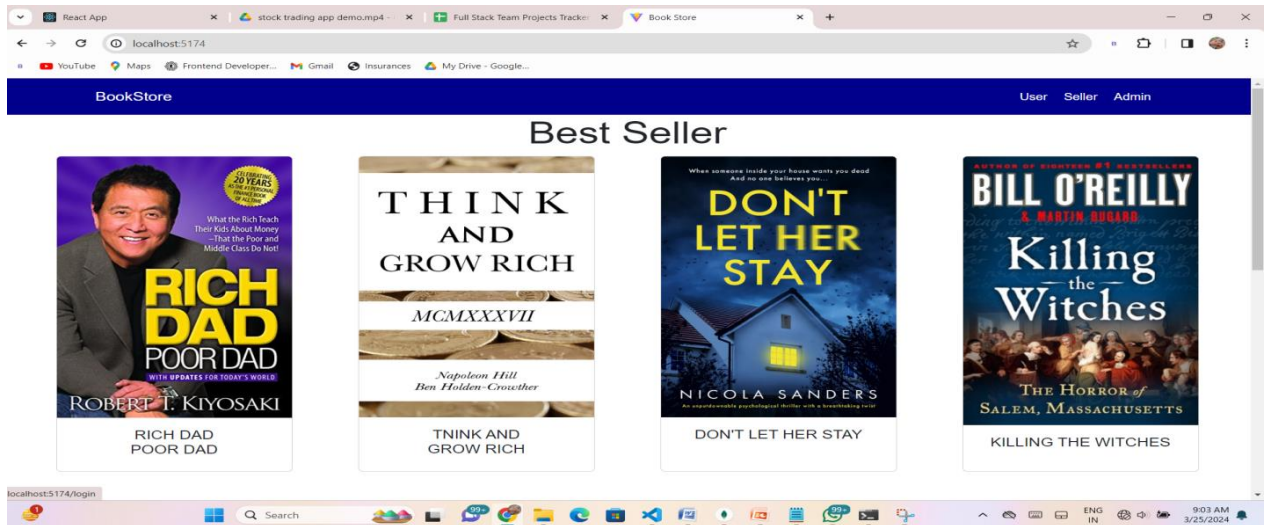
- **User Registration (POST /register):** Registers name, email, password (hashed), saved in DB.
- **User Login (POST /login):** Authenticates credentials → Returns JWT → Stored on client.

## 9. Testing

- **Unit Testing:** API functions and logic verified.
- **Integration Testing:** End-to-end testing for cart → order → confirmation.
- **UI Testing:** Responsive design tested across devices.
- **Security Testing:** JWT authentication and protected routes.

## 10. Screenshots / Demo





Wishlist



**The Mosquito**


Author:


Genre:


Price: \$


[Remove from Wishlist](#) [View](#)

My Orders

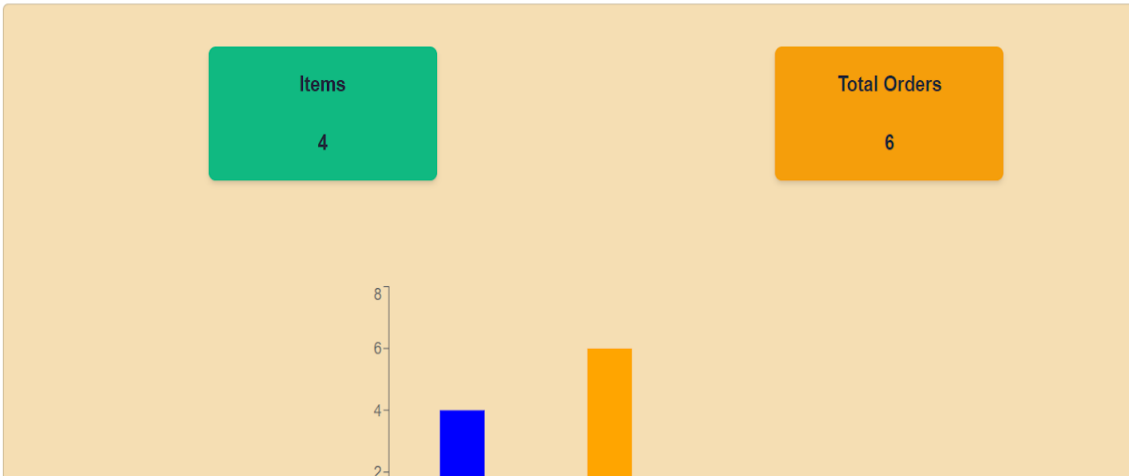
	ProductName:	Orderid:	Address:	Seller	BookingDate	Delivery By	Price	Status
	-0449	6580449015	dsfksf, asdas,(fasda), asdasda.	syed	18/12/2023	12/25/2023	\$199	delivered

	ProductName:	Orderid:	Address:	Seller	BookingDate	Delivery By	Price	Status
	-0f1d	6600f1d467	122-B, hyderabad,(517994), Telangana.	syed	25/3/2024	4/1/2024	\$229	ontheway








	ProductName:	Orderid:	Address:	Seller	BookingDate	Delivery By	Price	Status
	-0f25	6600f25067	, ,(), .	syed	25/3/2024	4/1/2024	\$229	ontheway

	ProductName:	Orderid:	Address:	Seller	BookingDate	Delivery By	Price	Status
	-0f25	6600f25f67	, ,(), .	syed	25/3/2024	4/1/2024	\$229	ontheway

Dashboard

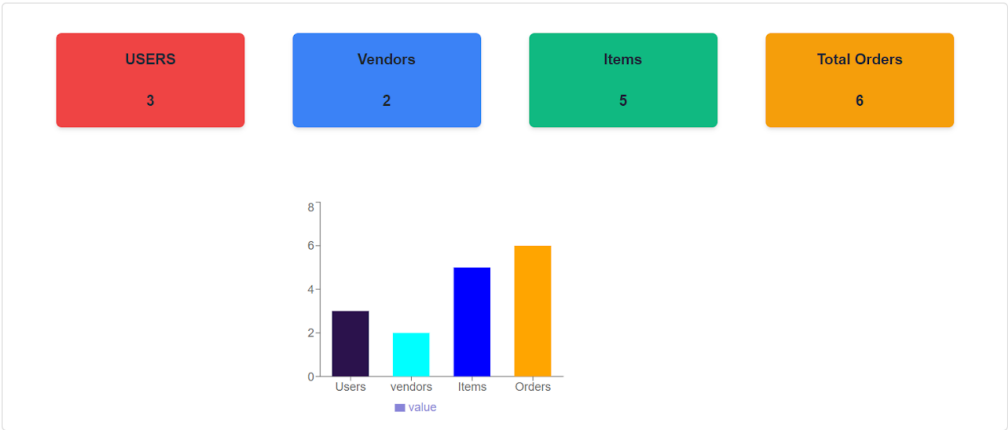


Vendor Products

	Product Name: -d98a	Orderid: 655d98a04f	Warranty: 1 year	Price: 100	
	Product Name: -d9a1	Orderid: 655d9a184f	Warranty: 1 year	Price: 130	
	Product Name: -d9d0	Orderid: 655d9d0d4f	Warranty: 1 year	Price: 130	
	Product Name:	Orderid:	Warranty:	Price:	



DashBoard



localhost:5174/ahome

Users

sl/no	Userld	User name	Email	Operation
1	655d9f4c4f4ace5f198b9abf	arshad	arshad@gmail.com	<div><div></div><div>view</div></div>
2	655e571f62e8144c8a9cb27f	shivani	shivani@gmail.com	<div><div></div><div>view</div></div>
3	6580442915b2ba8ff503a659	syed	syed@gmail.com	<div><div></div><div>view</div></div>


BookStore(Admin)

HomeUsersSellersLogout (syed )

Users

sl/no	Userid	User name	Email	Operation
-------	--------	-----------	-------	-----------

User Orders

	Product Name:	Orderid:	Address:	Buyer	Seller	BookingDate	Delivery By	Warranty	Price	Status
	-da8a	655da8aa49	22 ,sri apartments, banglore,(516269), karnataka	arshad	syed	22/11/2023	11/29/2023	1 year	189	delivered

Close

BookStore(Admin)

HomeUsersSellersLogout (syed )

Vendors

sl/no	Userid	User name	Email	Operation
1	655da3154992a4960bffc489	elf	elf@gmail.com	<div><div></div><div>view</div></div>
2	655c4b32b451e85b2daade96	syed	syed@gmail.com	<div><div></div><div>view</div></div>

The demo of the app is available

at: <https://drive.google.com/file/d/1y7TPSSUC5krDNSxNI66VvYI4MTxPy8XV/view?usp=sharing>

## **11. Known Issues**

- No Payment Gateway Yet – Purchases are simulated.
- Role-based admin access pending frontend enforcement.
- UI needs improvement on small screens.
- Real-time order tracking not yet implemented

## **12. Future Enhancements**

- Integrate payment gateways (Stripe, Razorpay)
- Add analytics dashboard for admin
- Allow users to upload book reviews
- Mobile App version for Android/iOS
- Real-time in-app notifications
- Implement recommendation engine (based on reading behavior)
- Real-time stock alert system