

BOOKNEST: Where Stories Nestle

1. Introduction

Project Title: Booknest

Team Members:

- N.Sai harshitha
- P.Rajitha
- M.Sowjanya
- K.Srilekha
- Afreen Shaik

2. Project Overview

Purpose:

Book-Store Application, a masterpiece crafted with precision using the powerful **MERN(MongoDB, Express.js, React, Node.js) Stack**. Immerse yourself in a world where the love for reading converges seamlessly with cutting-edge technology, redefining the way bibliophiles explore, discover, and indulge in their literary pursuits.

Features:

1. **Seamless Checkout:** Secure and smooth payment process with instant order confirmations and email notifications.
2. **Effortless Product Discovery:** Advanced search capabilities, intuitive category navigation, and powerful filters to help users find exactly what they need.
3. **Personalized Recommendations:** AI-driven algorithms analyze user behavior to provide curated product suggestions.
4. **Seller Dashboard:** Comprehensive tools for inventory tracking, order processing, and analytics to monitor performance metrics.
5. **Real-time Analytics:** Data-driven insights for sellers, highlighting sales trends, customer preferences, and product performance.

3. Architecture

Frontend:

- Developed using **React.js** for its component-based architecture and state management capabilities.
- Features dynamic components like:
 - **Product Listings:** Displays products with sorting and filtering options.
 - **Cart Management:** Allows users to add, update, or remove items in their cart.
 - **User Authentication:** Login and registration pages with secure validation.
 - **Admin Panel:** Provides sellers with tools to manage inventory and view analytics.

Backend:

- Built with **Node.js** and **Express.js**, ensuring scalability and high performance.
- Features include:
 - **API Endpoints:**
 - `/products` for fetching product data.
 - `/orders` for processing customer orders.
 - `/users` for managing user authentication.
 - **Middleware** for error handling and authentication using **JWT**.

Database:

- **MongoDB** serves as the database, storing collections for:
 - **Users:** Authentication credentials, profiles, and purchase history.
 - **Products:** Information on inventory, prices, categories, and descriptions.
 - **Orders:** Details about placed orders, delivery status, and payment.

This architecture ensures modularity, scalability, and efficient data management.

4. Setup Instructions

Prerequisites:

- **Node.js:** v14 or later
- **MongoDB:** Installed locally or set up using a cloud provider like MongoDB Atlas
- **npm:** Package manager for installing dependencies

Installation:

Clone the repository:

```
https://github.com/stinastanley/stina.git
```

Navigate to the project directory:

```
cd Booknest
```

Install dependencies for the backend and frontend:

Backend:

```
cd server  
npm install
```

Frontend:

```
cd client  
npm install
```

Run the servers:

Backend:

```
node index.js
```

Frontend:

```
npm start
```

5. Folder Structure

Client:

- `src/components`: Contains reusable components like **Navbar**, **ProductCard**, **CartItem**, etc.
- `src/pages`: Holds page components such as **Home**, **Cart**, **Checkout**, and **AdminPanel**.
- `src/services`: Manages API interactions for fetching and posting data.
- `src/redux`: Implements state management for cart items, user authentication, and order status.

Server:

- `routes`: Defines all RESTful API routes for users, products, and orders.
- `controllers`: Contains the logic for handling API requests and responses.
- `models`: Defines database schemas for **Users**, **Products**, and **Orders**.
- `middleware`: Handles authentication (JWT) and error management.

6. Running the Application

Frontend:

Navigate to the client directory:

```
cd client
```

Start the React application:

```
npm start
```

Backend:

Navigate to the server directory:

```
cd server
```

Start the Node.js server:

```
node index.js
```

7. API Documentation

Endpoints:

1. GET /products

- Fetches the list of all products.
- Parameters: Optional category and price filters.

Response:

json

```
[  
  {  
    "id": "123",  
    "name": "Gold Bracelet",  
    "price": 50,  
    "category": "Accessories"  
  }  
]
```

2. POST /orders

- Places a new order.
- Parameters: User ID, product details, and quantity.

Response:

json

```
{  
  "message": "Order placed successfully",  
  "orderId": "456"  
}
```

8. Authentication

JWT-based Authentication:

- **Login:** Issues a JWT token upon successful authentication.
- **Token Validation:** Protects private routes like `/orders` and `/admin`.
- **Logout:** Invalidates the token on the client side.

9. User Interface

Screens:

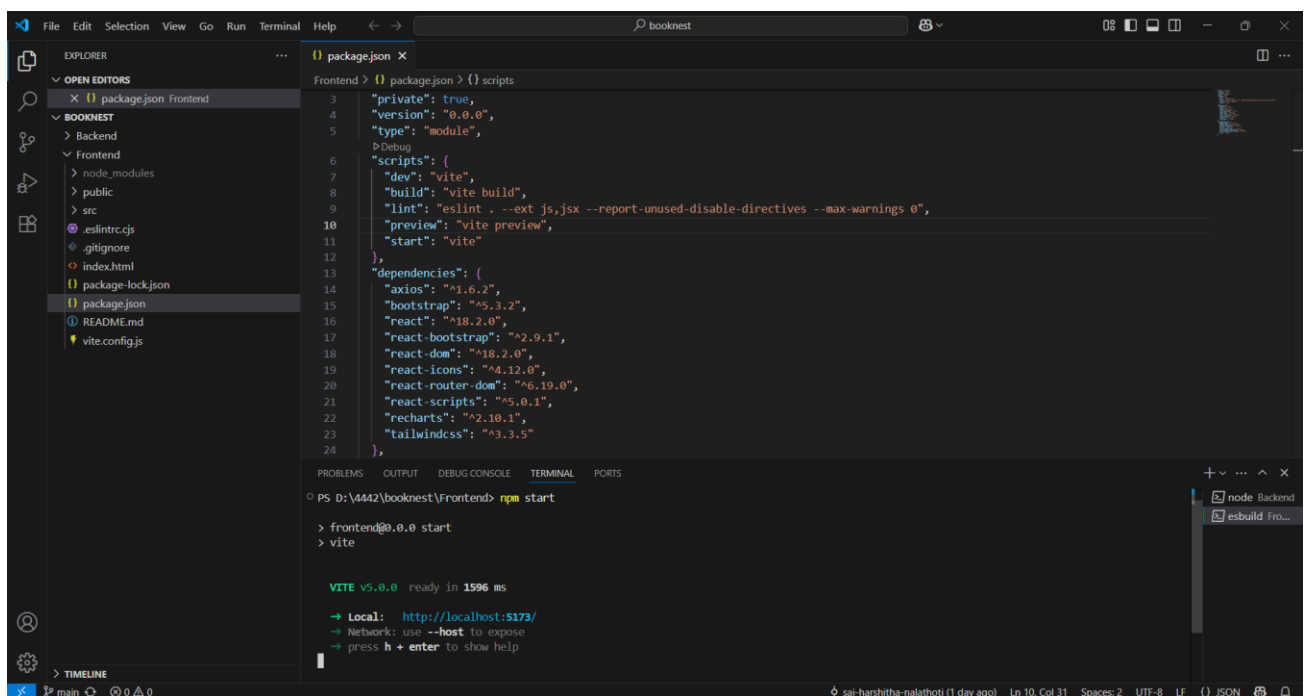
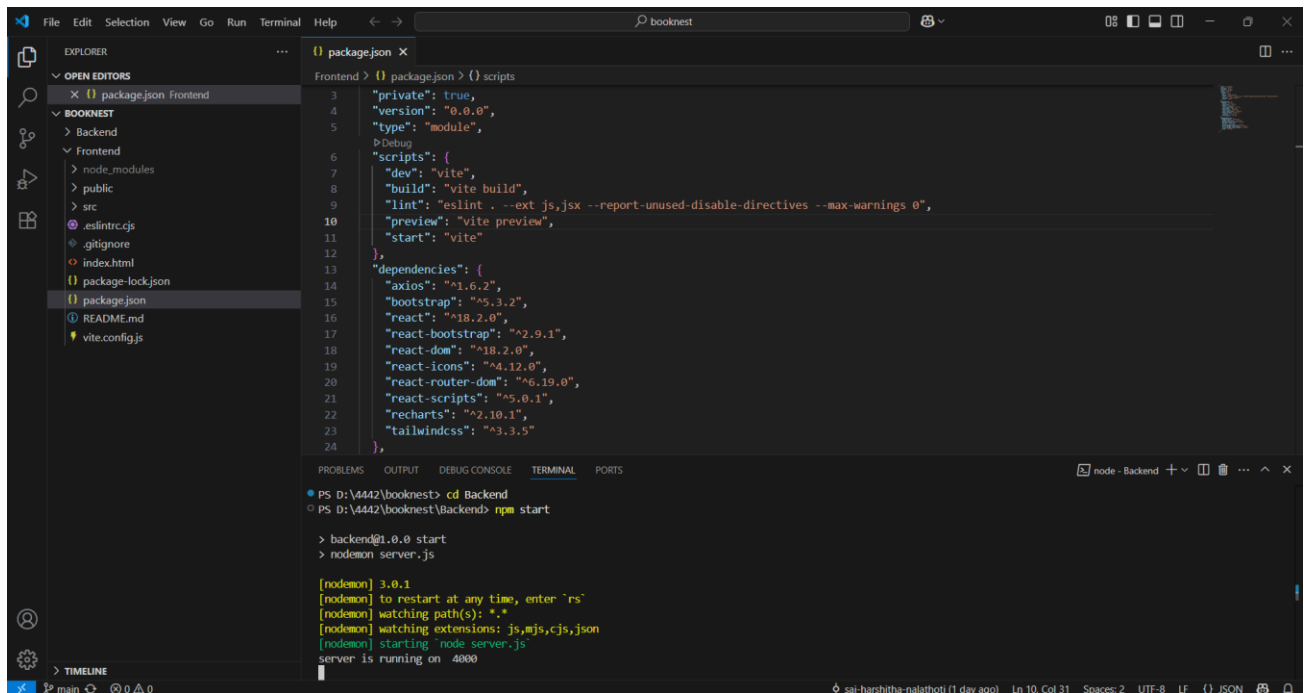
- **Home:** Displays trending products and categories.
- **Product Details:** Shows detailed information about a selected item.
- **Cart:** Summarizes selected products and their quantities.
- **Admin Dashboard:** Offers order status updates and analytics for sellers.

10. Testing

- **Unit Testing:** Conducted using **Jest** for components and backend logic.
- **API Testing:** Performed using **Postman** to validate endpoints.

11. Screenshots or Demo

Screenshots of key UI components and admin panel.



localhost:5173/login

BookStore

User Seller Admin

Login to user account

Email address

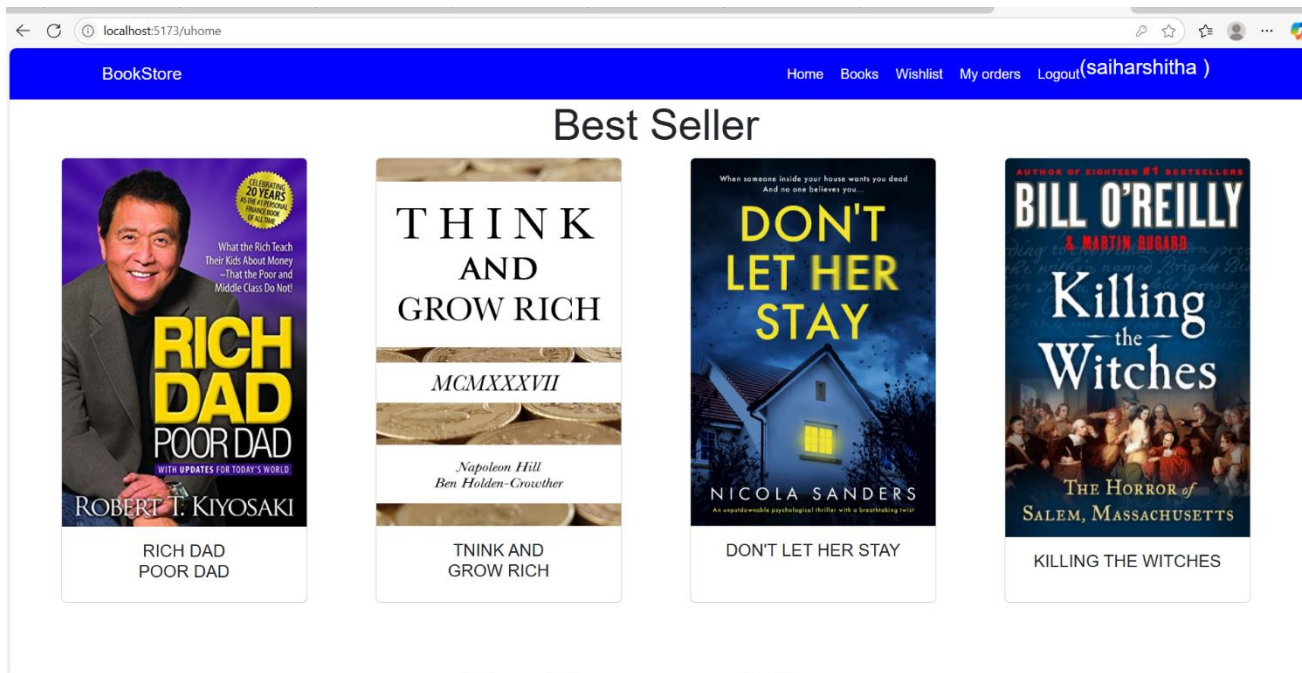
Saiharshithanallathoti@Gmail.Com

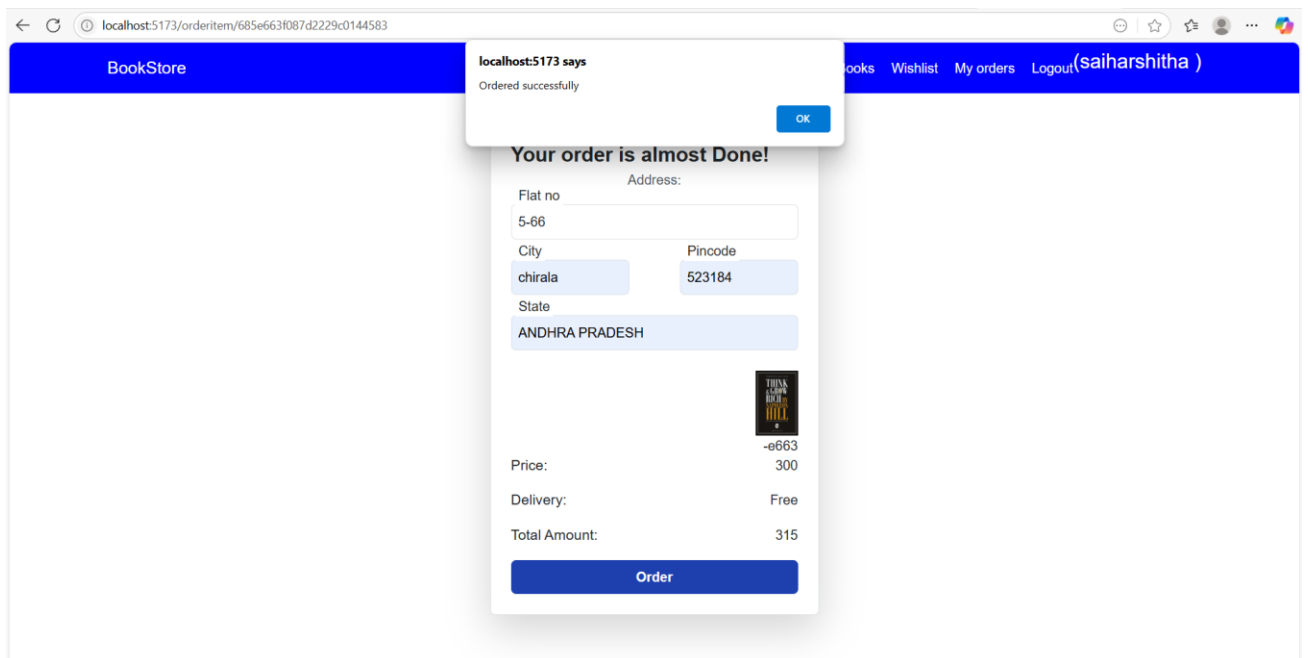
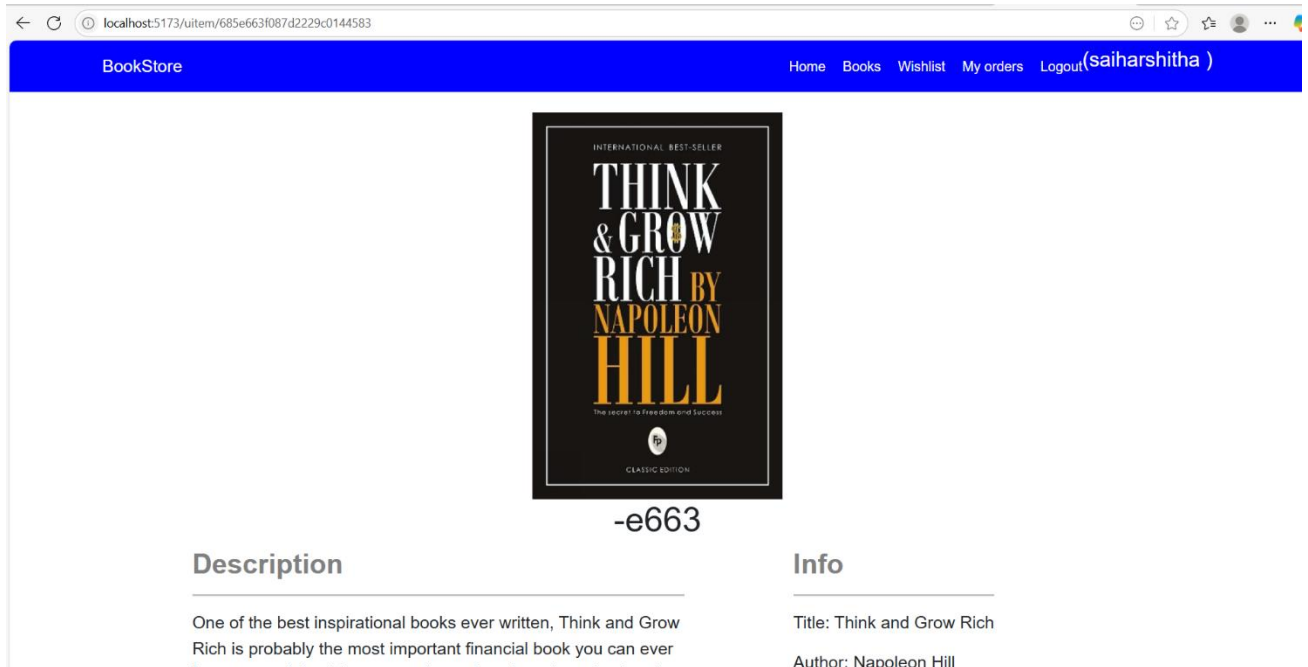
Password

...

Log in

Don't have an account? Create [Signup](#)






←↻📍 localhost:5173/myorders

🔍🏠👤⋮

BookStore

HomeBooksWishlistMy ordersLogout(saiharshitha)

My Orders



ProductName:

Orderid:

Address:

Seller

BookingDate

Delivery By

Price

Status

-e634

685e634c08

19-4-12,
chirala,(523184),
ANDHRA PRADESH.


rajitha

27/6/2025

7/4/2025

\$2599

ontheway



ProductName:

Orderid:

Address:

Seller

BookingDate

Delivery By

Price

Status

-fd1b

685fd1b8dc

5-66,
chirala,(523184),
ANDHRA PRADESH.

shanmukhi

28/6/2025

7/5/2025

\$399

ontheway

Contact us


←↻📍 localhost:5173/orders

🔍🏠👤⋮

BookStore(Seller

HomeMyproductsAdd BooksOrdersLogout(rajitha)

Orders



ProductName:

Orderid:

Customer Name

Address:

BookingDate

Delivery By

Warranty

Price

Status

685e634c08

saiharshitha

19-4-12,
chirala,(523184),
ANDHRA PRADESH.

27/6/2025

7/4/2025

1 year



2599

ontheway

← ↻ 📄 localhost:5173/sellers

BookStore(Admin) Home Users Sellers Logout (harshitha)

Vendors

sl/no	Userid	User name	Email	Operation
1	685e621e087d2229c01444fa	rajitha	rajithapenumudi44@gmail.com	 view
2	685e6432087d2229c0144578	shanmukhi	shanmukhi31@gmail.com	 view

localhost:5173/sellers

Demo Video.mp4 /

Link to live demo: https://drive.google.com/file/d/1H7EaQ-L6HsU2_FUDNMfqIU5_Y7qKBMh/view?usp=sharing

Link to GitHub: [sai-harshitha-nalathoti/booknest](https://github.com/sai-harshitha-nalathoti/booknest)

12. Known Issues

- **Slow Search Performance:** Optimization is required for large product datasets.
- **UI Bugs:** Minor alignment issues on smaller screens.

13. Future Enhancements

- **Voice Search:** Enable users to search using voice commands.
- **Mobile App:** Create a cross-platform app using React Native.
- **Multi-language Support:** Expand accessibility for gl

