

Full Stack Development with MERN

BookNest: Where Stories Nestle

Introduction :

Project Title: BookNest: Where Stories Nestle

Team Members:

Team ID : LTVIP2025TMID55905

Team Size : 4

Team Leader : Rumesa Khathoon

Team member : Kambham Divya Sudha

Team member : Madamala Nageswararao

Team member : Kuthadi Manasa Venkata Naga Sruthi

Project Overview :

- Purpose:**

The primary goal of BookNest is to create a fully functional, scalable online book-store platform using the MERN stack to cater to the needs of modern readers while ensuring a seamless and secure shopping experience. The platform allows users to register, authenticate, browse a comprehensive catalog of books, add selections to a cart, and complete purchases securely. Each step of the process is designed to prioritize user experience, data integrity, and operational efficiency.

Upon accessing the platform, users can create an account, ensuring a personalized experience with features like saved reading history and personalized recommendations. The authentication system, secured using JWT and bcrypt, protects user data while maintaining ease of login and registration. Once logged in, users can browse books with filtering options for genre, author, and ratings to discover books that align with their preferences.

The cart and purchase management system allows users to add books, adjust quantities, and proceed with secure checkouts, with order confirmations and notifications ensuring transparency. BookNest also maintains an order history for each user, allowing them to track purchases, view past orders, and manage their reading lists effectively.

For administrators and sellers, BookNest provides inventory and order management capabilities, enabling them to add and manage books, monitor stock levels, update book information, and handle orders efficiently. The backend system supports smooth operations by handling all API calls, database interactions, and security processes while ensuring fast responses and data consistency across the platform.

The platform is designed with scalability in mind, supporting a growing user base and book catalog without compromising performance. By leveraging the power of MongoDB, Express.js, React.js, and Node.js, BookNest offers a modern, clean, and efficient system that redefines the digital book-buying experience.

- **Features:**

BookNest offers a comprehensive suite of features tailored to readers, sellers, and administrators, ensuring a seamless, engaging, and secure online book-buying experience:

1. **User Registration & Authentication:** Users can securely register using their email, create profiles, and manage account details. JWT and bcrypt are used to ensure secure authentication, providing a safe environment for user data.
2. **Book Listings with Advanced Filters:** Users can explore a vast library of books categorized by genres, authors, ratings, and price ranges. Advanced search and filter functionalities allow readers to find books according to their preferences easily.
3. **Detailed Book Pages:** Each book listing comes with detailed descriptions, author information, user ratings, and reviews, helping readers make informed choices before adding books to their cart.
4. **Cart & Secure Checkout:** Users can add books to their cart, modify quantities, and proceed to checkout with a streamlined purchase process. The secure checkout system ensures safe transactions while providing a clear summary of the order.
5. **Order Confirmation & Notifications:** Users receive real-time notifications upon successful purchases, including order ID, book details, total price, and estimated delivery timelines.
6. **Order History Tracking:** Users can view past and current orders, track shipments, and review their purchases, allowing them to maintain a structured reading record.
7. **Admin and Seller Panels:** Administrators and sellers can manage the book catalog, update inventory, handle user queries, and track sales data efficiently. They can add or remove book listings, update prices and stock statuses, and manage order processing seamlessly.
8. **Responsive Design:** BookNest is accessible across devices, ensuring users can browse and purchase books from desktops, tablets, and smartphones with consistent performance and user experience.

9. Notifications and Recommendations: Users receive recommendations based on their reading preferences and browsing history, along with timely notifications for order updates, ensuring engagement and convenience.

10. Data Security: BookNest employs secure authentication, encrypted data handling, and role-based access control to protect user data and platform integrity.

By integrating these features, BookNest offers a user-centric, efficient, and enjoyable online book store experience for all stakeholders.

Architecture

- **Frontend:**

The frontend is built using React.js, a component-based JavaScript library that enables building dynamic and responsive user interfaces for browsing and purchasing books. It uses a modular structure where reusable components handle book listings, cart management, order history, and user authentication. React Router manages page navigation, while styling is handled using Bootstrap and Material UI for a consistent look across devices. Axios is used for API requests, allowing smooth interaction with the backend.

- **Backend:**

The backend uses Node.js with the Express.js framework to build RESTful APIs that manage user authentication, book listings, order placement, and inventory management. Express handles routing, middleware, and business logic while maintaining security using JWT for authentication and bcrypt for password hashing. It exposes endpoints for user registration/login, book management, cart operations, and order tracking.

- **Database:**

MongoDB is used as a NoSQL document-oriented database, storing data in flexible JSON-like documents with three primary collections:

- **Users:** Stores user details, login credentials, and user roles (user/admin/seller).
- **Books:** Contains book-specific information including title, author, genre, price, stock quantity, and description.
- **Orders:** Records order details linking users and books, including order status, quantities, and purchase dates.

These collections are interlinked using references, ensuring efficient queries while maintaining data consistency across the BookNest application.

Setup Instructions

Prerequisites:

- Node.js & npm: Required for running the BookNest backend and frontend. [Download here](#)
- MongoDB: Used for storing book store data, can be installed locally or using MongoDB Atlas. [Download here](#)
- Git: For cloning the BookNest project repository. [Download here](#)

Installation:

[1] Clone the Repository:

Open the terminal and run:

```
git clone <repository_link>
```

```
cd <project_folder>
```

[2] Setup Backend:

Navigate to the backend folder:

```
cd backend
```

```
npm install
```

Create a .env file with:

```
PORt=5000
```

```
MONGO_URI=<your_mongo_uri>
```

```
JWT_SECRET=<your_jwt_secret>
```

Start the backend server:

```
npm start
```

The backend will run on <http://localhost:5000>.

[3] Setup Frontend:

In a new terminal:

```
cd frontend
```

```
npm install
```

npm start The frontend will run on <http://localhost:3000>, providing your MERN BookNest application with a complete functional environment for development and testing.

Folder Structure

The frontend structure includes:

- public/: Static files.
- src/components/: Reusable UI components like BookList, Cart, and Navbar.
- src/pages/: Page-level components such as Home, Orders, and BookDetails.
- src/services/: Axios API service files.
- App.js: Routing and state management.

The backend structure includes:

- config/: Database connection and environment configurations.
- controllers/: Business logic for users, books, and orders.
- models/: Mongoose schemas for Users, Books, and Orders.
- routes/: Express routes connecting APIs with controllers.
- middleware/: JWT authentication and error handling.
- server.js: Entry point for the backend server.

This modular structure ensures clean code management and scalability for BookNest.

Running the Application

To run BookNest locally:

1. Ensure Node.js, npm, and MongoDB are installed.

2. Backend: Navigate to the backend folder, run npm install to install dependencies, create a .env with PORT, MONGO_URI, and JWT_SECRET, then run npm start.
3. Frontend: Navigate to the frontend folder, run npm install, then run npm start.

The backend will run on <http://localhost:5000> and the frontend on <http://localhost:3000>, ensuring full-stack functionality for your book store application.

API Documentation

Key APIs include:

- POST /api/users/register: Register a user.
- POST /api/users/login: User login.
- GET /api/books: Fetch all books with filters.
- POST /api/orders: Place an order.
- GET /api/orders/:userId: Retrieve user orders.

Each endpoint is secured with JWT, ensuring authenticated operations while supporting clear CRUD functionalities for user registration, book browsing, and order management.

UI Details

BookNest's UI includes:

- Home Page: Book listings with search and filter options.
- Book Details Page: Detailed view with title, author, genre, price, and description.
- Cart Page: Manage items, view totals, and checkout.
- Orders Page: Track current and past orders.
- Admin Dashboard: Manage books and view orders.

The UI is built using React.js and styled with Bootstrap/Material UI, ensuring a clean, responsive experience for all users, making the book-buying journey seamless and engaging.

Testing

Unit Testing: Tests React components and backend API controllers using Jest and React Testing Library.

Integration Testing: Uses Postman and automated scripts to test API endpoints and data flow.

User Acceptance Testing (UAT): Real users test workflows like browsing, cart management, and order placement.

Performance Testing: Checks how the application performs under concurrent users.

Security Testing: Tests JWT authentication, encrypted data handling, and role-based access control.

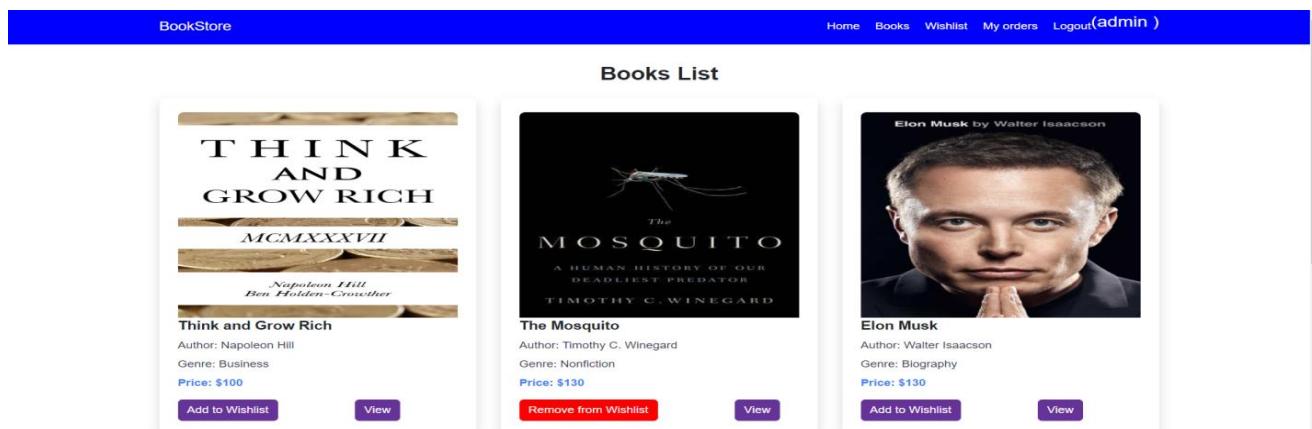
Regression Testing: Ensures new updates do not break existing functionalities.

Automated Testing Pipelines: Future plans to add automated pipelines for continuous testing.

Screenshots or Demo

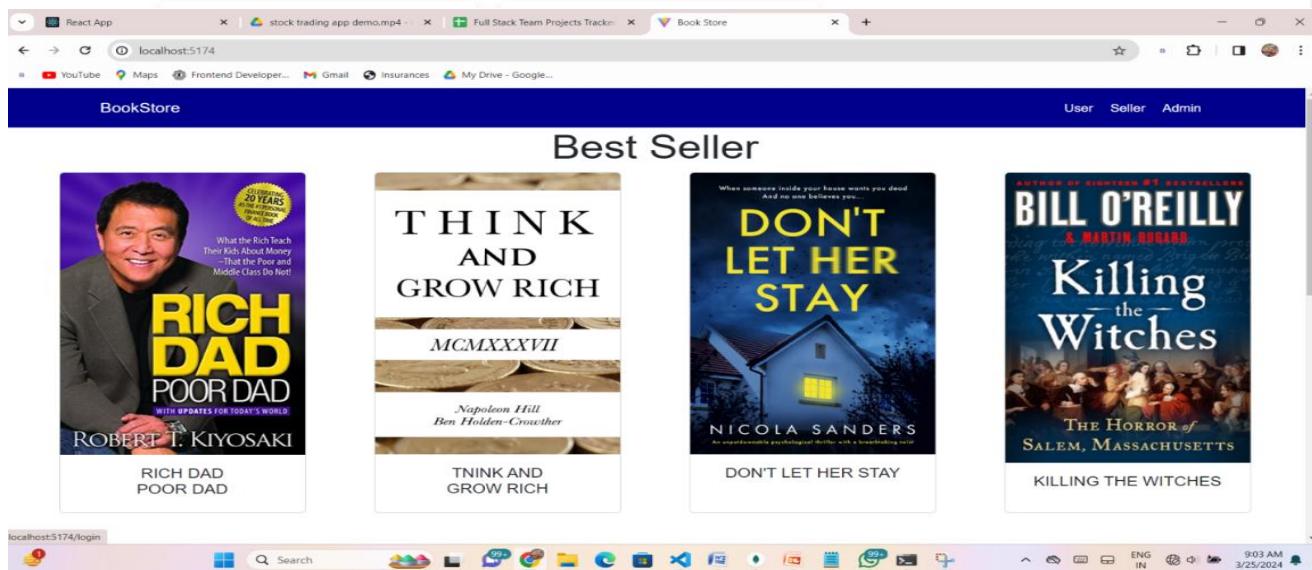
To provide a clear understanding of the Book Nest application, below are screenshots showcasing the key features of the platform:

- Landing page
- Login Page
- My Bookings Page
- Seller Dashboard



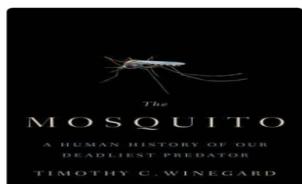
The screenshot shows the 'Books List' section of the Book Store. It displays three book cards:

- THINK AND GROW RICH** by Napoleon Hill and Ben Holden-Crowther. Price: \$100. Buttons: Add to Wishlist, View.
- The Mosquito** by Timothy C. Winegard. Price: \$130. Buttons: Remove from Wishlist, View.
- Elon Musk** by Walter Isaacson. Price: \$130. Buttons: Add to Wishlist, View.



The screenshot shows the 'Best Seller' section of the Book Store. It displays four book cards:

- RICH DAD POOR DAD** by Robert T. Kiyosaki. Price: \$100. Buttons: Add to Wishlist, View.
- THINK AND GROW RICH** by Napoleon Hill and Ben Holden-Crowther. Price: \$100. Buttons: Add to Wishlist, View.
- DON'T LET HER STAY** by Nicola Sanders. Price: \$130. Buttons: Add to Wishlist, View.
- Killing the Witches** by Bill O'Reilly and Martin Dugard. Price: \$130. Buttons: Add to Wishlist, View.

Wishlist**The Mosquito**

Author:

Genre:

Price: \$

[Remove from Wishlist](#) [View](#)**My Orders**

	ProductName:	OrderId:	Address:	Seller	BookingDate	Delivery By	Price	Status
	-0449	6580449015	dsksf, asdas,(fasda), asdasda.	syed	18/12/2023	12/25/2023	\$199	delivered
	-011d	660011d467	122-B, hyderabad,(517994), Telangana.	syed	25/3/2024	4/1/2024	\$229	ontheway
	-0f25	6600125067	,(), ,	syed	25/3/2024	4/1/2024	\$229	ontheway
	-0f25	6600125167	,(), ,	syed	25/3/2024	4/1/2024	\$229	ontheway

Dashboard**USERS**

3

Vendors

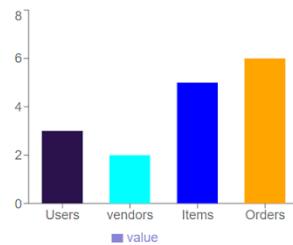
2

Items

5

Total Orders

6

**Login to user account**

Email address

Password

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

Known Issues

BookNest currently has a few known issues under observation. Occasionally, there are minor **inventory sync delays** when multiple users purchase the same book simultaneously. **Notification delays** can occur under heavy load, causing slight lags in sending order confirmations. Some **ui responsiveness issues** may appear on smaller devices, requiring additional testing and fine-tuning. There are also **limitations in image upload size** for book cover uploads. Edge cases in **role-based access control** may occasionally allow visibility of admin controls without functionality. These issues are tracked and progressively improved with feedback from users and testers, ensuring continuous enhancement of BookNest's stability and usability.

Future Enhancements

Future plans for BookNest include adding a **wishlist feature** to allow users to save books for later, **AI-based book recommendations** using user history, and **payment gateway integration** for online transactions. Plans also include **real-time chat support** for user queries, **multi-language support** to enhance accessibility, and **advanced analytics** for admin and sellers to monitor sales and inventory trends. Additionally, developing a **mobile application for Android and iOS** will ensure wider accessibility for users. These enhancements aim to provide a richer, smoother, and more engaging user experience, aligning with BookNest's mission to modernize and simplify book purchasing for readers.