



BookNest: Where Stories Nestle A MERN Stack Full-Stack Book Store Application

SHORT TERM INTERNSHIP PROGRAMME

By **SmartBridge**

Category : Full Stack Development

Team ID : LTVIP2025TMID59020

Team Size : 4

Team Leader : Siddabathula Jayaram-(24485A0521)-C.S.E

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE,GUDLAVALLERU

Team member : Shaik Nouhid-(22481A12F6)-IT

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE , GUDLAVALLERU

Team member : Seshapu Murali-(22481A6098)-IoT

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE , GUDLAVALLERU

Team member : Shaik Jameer Ahmadh-(23485A0519)-CSE

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE, GUDLAVALLERU

project Description:

Welcome to the literary haven of the digital age—introducing our revolutionary 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.

Tailored for the modern book enthusiast, our MERN-based Book-Store Application seamlessly blends robust functionality with an intuitive user interface. From the joy of discovering new releases to the nostalgia of revisiting timeless classics, our platform promises an immersive reading experience customized to cater to your literary preferences.

Fueling the backbone of our application is MongoDB, ensuring a scalable and efficient database infrastructure that facilitates swift access to an extensive collection of literary works. Express.js, with its streamlined web application framework, establishes a responsive and efficient server, while Node.js ensures high-performance, non-blocking I/O operations—resulting in a seamless and enjoyable user experience.

At the heart of our Book-Store Application lies React, a dynamic and feature-rich JavaScript library. Dive into a visually enchanting and interactive interface where every click, search, and book selection feels like a literary journey. Whether you're exploring on a desktop, tablet, or smartphone, our responsive design ensures a consistent and delightful experience across all devices.

Say farewell to the constraints of traditional bookstores and embrace a new era of possibilities with our MERN Stack Book-Store Application. Join us as we transform how you connect with literature, making the discovery of your next favorite read an effortless and enriching experience. Get ready to turn the digital pages of a new chapter in reading, where every book is just a click away, and the literary world is at your fingertips. It's time to open the door to a future where the love for books meets the convenience of modern technology

Key Features

-  **User Authentication** (JWT, Firebase)
 -  **Admin Dashboard** for managing books
 -  Book Listing & Filtering (by genre, rating, etc.)
 -  Cart functionality (Add/Remove/Clear items)
 -  Secure Checkout & Order Management
 -  Order History Tracking
 -  Fully Responsive UI using **Tailwind CSS**
-

Tech Stack

- **Frontend:** React.js, Tailwind CSS, Redux Toolkit, React Router DOM
- **Backend:** Node.js, Express.js
- **Database:** MongoDB, Mongoose
- **Authentication:** Firebase Auth & JWT
- **State Management & API:** Redux Toolkit, RTK Query
- **Deployment:** Vercel (Frontend), Render/Heroku (Backend)

Learning Outcomes

- Implemented full-stack CRUD operations
- Gained hands-on experience with authentication & protected routes
- Developed skills in managing API communication with Redux Toolkit
- Understood and applied real-world eCommerce logic in a book store setting
- Deployed a scalable, production-ready full stack application

 **Topic/Chapter Breakdown****S.no Topic**

- 1 Introduction
- 2 Project Overview
- 3 Project Setup
- 4 Install Tailwind CSS
- 5 Setup React Router DOM
- 6 Customizing the Project
- 7 Creating a Responsive Navbar
- 8 Banner Section
- 9 Top Sellers Section
- 10 Recommend Books
- 11 News Section
- 12 Footer Section
- 13 Login and Register Pages
- 14 Install Redux Toolkit & Cart Logic
- 15 Cart Page: Add/Remove/Clear
- 16 Checkout Page
- 17 Server Setup, Mongoose & MongoDB Integration
- 18 Blog Schema & Model Creation
- 19 Book Routes & Controllers
- 20 Create bookApi with RTK Query & Display Books
- 21 Firebase Setup: Login/Register
- 22 Implement Private Routes
- 23 Create Order Endpoint & Place Order
- 24 Admin Setup & Token Verification
- 25 Admin Router & Dashboard Login
- 26 Admin Panel: Add/Edit/Delete Books

S.no Topic

27 Deployment: How to Deploy MERN Stack Project

Project Introduction: BookNest – Where Stories Nestle

BookNest is a full-stack web application designed to transform the traditional book-buying experience into a seamless digital journey. Developed using the powerful **MERN stack (MongoDB, Express.js, React.js, Node.js)**, this project serves as a modern, scalable, and responsive **online bookstore** tailored for avid readers in the digital age.

The application enables users to **browse a wide variety of books**, filter them by genres, authors, or ratings, and securely **purchase** them through a user-friendly interface. It features **user authentication**, a **real-time shopping cart**, **order management**, and an **admin dashboard** for managing inventory. The front-end is built using **React.js** and styled with **Tailwind CSS** for a responsive experience across all devices, while the back-end ensures secure, efficient, and scalable data handling.

Inspired by real-world scenarios like Sarah's (a reader with a busy schedule), BookNest eliminates the need to visit physical bookstores and offers the same joy of browsing and discovering books from the comfort of one's home. This project reflects the future of eCommerce in the literary domain—where convenience, technology, and passion for reading unite.

Project Overview

BookNest: Where Stories Nestle is a MERN stack-based online bookstore that offers users an engaging, seamless, and responsive digital reading and shopping experience. It supports user registration, book discovery, secure purchasing, and order tracking, along with admin functionalities like book management. The application was developed as part of a full-stack web development internship and simulates a real-world eCommerce bookstore tailored for book enthusiasts.

Project Setup

The setup process involves initializing both the **frontend (React)** and **backend (Node.js + Express)** environments. Key steps include:

1. **Initialize the project** using npx create-react-app booknest-frontend for the front-end and npm init -y for the back-end.
 2. **Set up the folder structure** with separate directories for frontend and backend code to ensure modularity.
 3. **Install essential dependencies** for backend (express, mongoose, cors, dotenv, jsonwebtoken) and frontend (axios, redux, react-router-dom).
 4. **Connect MongoDB Atlas** using Mongoose and environment variables.
 5. **Test both front-end and back-end servers** to ensure they run without conflict using concurrent server setup.
-

Install Tailwind CSS

To style the application with a modern and responsive UI, Tailwind CSS was integrated into the React frontend.

Steps to Install Tailwind CSS:

- **Navigate to the frontend project:**

```
bash  
  
cd booknest-frontend
```

[Copy](#) [Edit](#)

- **Install Tailwind and its dependencies:**

```
csharp  
  
npm install -D tailwindcss postcss autoprefixer  
npx tailwindcss init -p
```

[Copy](#) [Edit](#)

- **Configure tailwind.config.js:**

```
js  
  
content: ["./src/**/*.{js,jsx,ts,tsx}"],
```

[Copy](#) [Edit](#)

- **Add Tailwind directives to src/index.css:**

css

 Copy  Edit

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

- **Start using Tailwind utility classes in your React components for rapid styling and responsiveness.**

Setup React Router DOM

To manage page navigation and routing within the **BookNest** application, we used **React Router DOM**, a popular library for client-side routing in React applications.

Installation:

bash

CopyEdit

```
npm install react-router-dom
```

Configuration:

1. In App.js, wrap the main component tree with <BrowserRouter>.
2. Define routes using <Routes> and <Route> for different pages such as:
 - / → Home
 - /login → Login Page
 - /register → Register Page
 - /books/:id → Book Details Page
 - /cart → Shopping Cart
 - /checkout → Checkout Page
 - /admin → Admin Dashboard

Benefits:

- Enables SPA (Single Page Application) experience.
- Facilitates private and dynamic routing.
- Keeps the URL in sync with the UI, improving navigation clarity.

Customizing the Project

This phase focuses on personalizing the app's design, layout, and structure to align with the **BookNest** theme and improve usability.

Key Customizations:

- Applied **Tailwind CSS** for responsive and clean UI design.
- Created a consistent layout structure including:
 - Header/Navbar
 - Banner & Sections (e.g., Top Sellers, Recommended)
 - Footer
- Customized color palette, spacing, and typography to reflect a professional bookstore look and feel.
- Improved component reusability with modular design patterns.

Tools Used:

- Tailwind Utility Classes
 - Reusable React Components
 - Component-level styling and layout structuring
-

Creating a Responsive Navbar

The **navigation bar** is a crucial component for user interaction, and it was built to be fully responsive and functional across all devices.

Features:

- Brand/logo area
- Navigation links: Home, Categories, Cart, Login/Register
- Mobile menu toggle for smaller screens (hamburger menu)
- Conditional rendering: changes options based on user login status (e.g., Admin Dashboard)

Tech Used:

- Tailwind's flex, grid, hidden, block, and media queries
- React state/hooks for mobile toggle behavior

- React Router's NavLink for active state management

Outcome:

- Consistent navigation across all screen sizes
- Quick access to key sections of the app
- Enhances user experience by providing clarity and structure

Banner Section

The **Banner Section** is the visual introduction of the BookNest homepage, designed to immediately capture the attention of visitors.

Purpose:

- Highlight the platform's theme: reading, discovery, and digital convenience.
- Provide a welcoming message or promotional content.

Implementation:

- Created a full-width banner using Tailwind CSS with **responsive flex and grid layouts**.
- Included:
 - A compelling headline and subtext.
 - A "Shop Now" or "Explore Books" CTA (Call-To-Action) button.
 - Background images or gradients relevant to books/libraries.

Responsive Design:

- Adjusted padding, font sizes, and image positions for different screen sizes to ensure accessibility and visual appeal across devices.
-

Top Sellers Section

The **Top Sellers Section** displays a curated list of the most popular books on the platform, helping users discover trending titles.

Features:

- Pulls data dynamically from the backend using the book API.
- Books displayed with:

- Cover image
- Title & Author
- Rating
- Price
- Optional "Add to Cart" button or "View Details" link.

Tech Used:

- Tailwind for grid layout and responsiveness.
- Fetching book data using RTK Query or Axios.

Goal:

To encourage sales and engagement by showcasing what's trending among readers.

Recommend Books

The **Recommended Books Section** is personalized or curated content designed to engage users with suggestions based on popularity, genre, or admin curation.

Implementation:

- Could be dynamically based on:
 - Genre filtering
 - Top-rated books
 - User preferences (in advanced versions)
- Reused existing **BookCard** components for consistency.

Tools:

- React state or Redux store for dynamic filtering.
 - Tailwind's scrollable container or carousel (optional for mobile sliders).
-

News Section

The **News or Blog Section** shares updates, articles, or literary insights, adding informational and SEO-friendly value to the platform.

 **Features:**

- Blog entries stored in MongoDB and fetched via backend APIs.
- Each post includes:
 - Title
 - Short description
 - Date
 - “Read more” button or link

 **Benefits:**

- Enhances user engagement with reading-related content.
 - Opportunity for admin to update users about offers, releases, or reading tips.
-

 **Footer Section**

The **Footer** provides essential links and reinforces the branding of BookNest while ensuring accessibility to key site resources.

 **Contents:**

- Brand logo or name
- Navigation links (Home, Categories, Contact, About)
- Social media icons (LinkedIn, GitHub, etc.)
- Copyright

 **Layout:**

- Implemented using Tailwind’s grid or flexbox.
- Responsive stacking for mobile views.

 **Extras (Optional):**

- Newsletter signup
- Terms & conditions, privacy policy links

 **Login and Register Pages**

Developed secure **Login and Registration** pages to allow users to create and access their accounts.

- Used **React Forms** with validation.
 - Integrated **Firebase Authentication** and JWT tokens for secure login.
 - Password protection, error handling, and redirection after successful login.
-

Install Redux Toolkit & Cart Logic

Implemented **Redux Toolkit** to manage application-wide state, focusing on cart functionality.

- Installed with:

bash

CopyEdit

```
npm install @reduxjs/toolkit react-redux
```

- Created **slices** for cart management.
- Actions include: addToCart, removeFromCart, clearCart.

Benefits:

- Centralized state management
 - Easy debugging and predictable UI behavior
-

Cart Page: Add/Remove/Clear

Built a complete **shopping cart system**.

- Displays added books with quantities and total cost.
 - Users can:
 - Adjust quantity
 - Remove individual items
 - Clear the entire cart
 - State persisted via **Redux Toolkit**
-

Checkout Page

Designed a responsive **checkout interface** where users finalize their purchases.

- Displays order summary (books, quantities, total cost).
 - Integrates with backend to:
 - Validate user session
 - Store order details
 - Update inventory
-

Server Setup, Mongoose & MongoDB Integration

Set up the **Node.js + Express.js backend server** with connection to **MongoDB Atlas** using **Mongoose**.

- Installed dependencies: express, mongoose, cors, dotenv
 - Connected to MongoDB using a .env secure URI
 - Created schemas for books, users, orders
-

Blog Schema & Model Creation

Added a **Blog** feature for sharing articles and news.

- Created a **Mongoose schema** for blogs:
 - title, content, author, date
 - Used in admin dashboard to post and manage blog entries
-

Book Routes & Controllers

Established **RESTful APIs** for book operations.

- GET /books – fetch all books
- POST /books – add a book (admin only)
- PUT /books/:id – edit book
- DELETE /books/:id – delete book
- Controller logic includes validation and error handling

Create bookApi with RTK Query & Display Books

Used **Redux Toolkit Query (RTK Query)** to fetch and manage API data.

- Created bookApi.js using createApi
- Integrated with Redux store
- Displayed books in components with loading and error states

Benefits:

- Automatic caching, refetching, and state updates
 - Cleaner and scalable code
-

Firebase Setup: Login/Register

Integrated **Firebase Authentication** to manage user sessions securely.

- Configured Firebase project
 - Enabled Email/Password login
 - Connected Firebase to React using firebase/app and firebase/auth
-

Implement Private Routes

Created **Protected Routes** to restrict access to certain pages (e.g., Cart, Checkout, Admin).

- Implemented a PrivateRoute component
 - Verified user authentication status before rendering protected components
-

Create Order Endpoint & Place Order

Implemented server-side **Order API**.

- Endpoint: POST /orders
- Payload: book IDs, quantities, user ID
- On success:
 - Stores order in MongoDB
 - Updates inventory

- Returns confirmation ID
-

Admin Setup & Token Verification

Built token-based **admin verification system**.

- JWT token generated upon login
 - Middleware checks token validity and admin privileges
 - Admin users get access to dashboard and management routes
-

Admin Router & Dashboard Login

Developed **admin-specific routes** and a **separate login panel**.

- /admin login for admins
 - After authentication, admins redirected to dashboard
 - Restricted from user-facing routes using role-based auth
-

Admin Panel: Add/Edit/Delete Books

Functional **Admin Dashboard** for managing the store.

- CRUD operations for books and blogs
 - User-friendly forms for data entry
 - Real-time updates using Redux and API integration
-

Deployment: How to Deploy MERN Stack Project

Deployed frontend and backend separately:

Frontend

- Hosted on **Vercel** using GitHub integration
- Optimized for production with npm run build

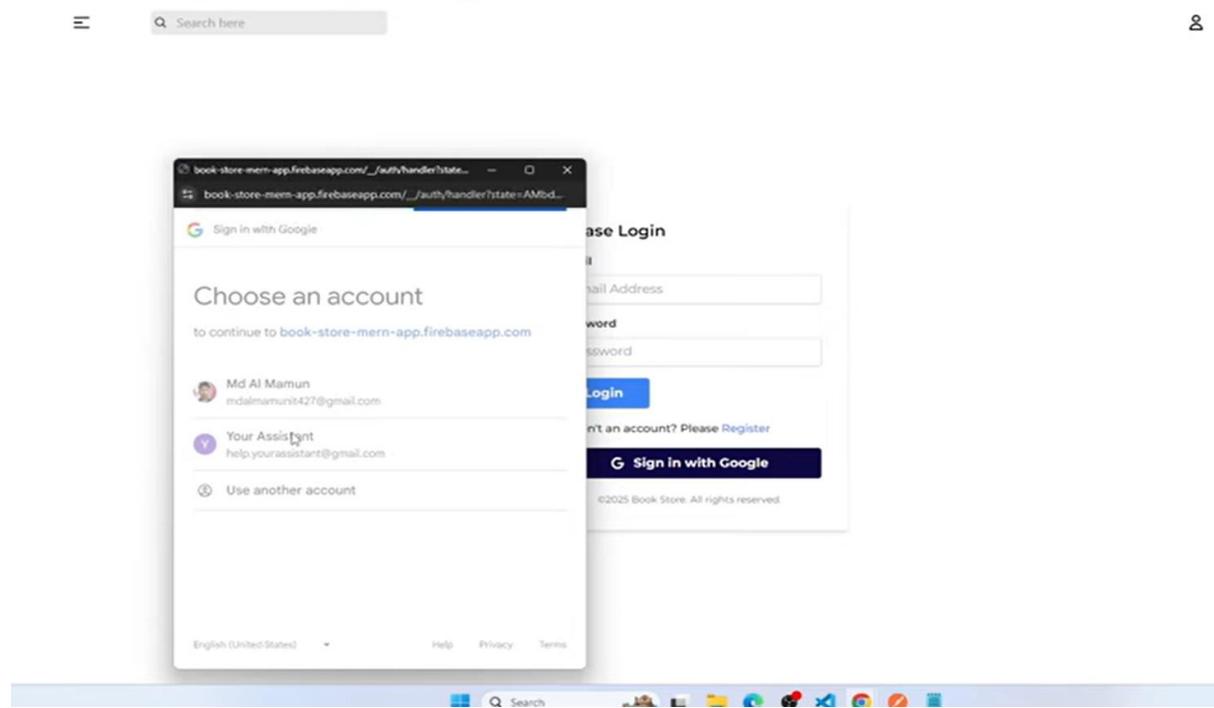
Backend

- Hosted on **Render/Heroku**
- MongoDB database hosted on **MongoDB Atlas**

Environment Variables securely set for API keys and DB URIs.

Project implementation:

Login with gmail:

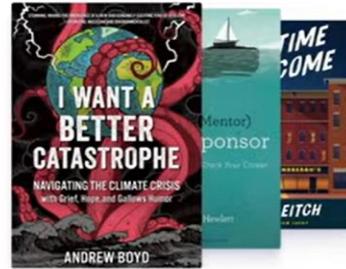


Placing order by adding item in to cart:

New Releases This Week

It's time to update your reading list with some of the latest and greatest releases in the literary world. From heart-pumping thrillers to captivating memoirs, this week's new releases offer something for everyone.

[Subscribe](#)



Top Sellers

Choose a genre ▾



Young Bucks: Killing the Business
The Young Bucks are two...
more when you visit this



The Ride of a Lifetime
A grand vision defined: The
CEO of The Walt Disney...



**Top 10 Fiction Books
This Year 2025 with
Token**
An annual list of the best

Placing order by COD:

Cash On Delevery

Total Price: \$20.00
Items: 1

Personal Details
Please fill out all the fields.

Full Name	Moon		
Email Address	help.yourassistant@gmail.com		
Phone Number	123456		
Address / Street	A	City	B
Country / region	C	State / province	D
		Zipcode	12345

I am agree to the [Terms & Conditions](#) and [Shipping Policy](#).

[Place an Order](#)

Order details in your orders:

Your Orders

#1

Order ID: 66f10c0036f7d6490e426122

Name: Mamun

Email: help.yourassistant@gmail.com

Phone: 8521647893

Total Price: \$29.99

Address:

Kolkata, Abc, India, 7891

Products Id:

66f04d2b8568a4e7c7938c09

2

Order ID: 66f10b7836f7d6490e42611e

Name: Md Al Mamun

Email: help.yourassistant@gmail.com

Phone: 123456789

Total Price: \$34.98

Address:

Dhaka, Dhaka, Bangladesh, 1205

Products Id:

66f04d2b8568a4e7c7938c07

66f04d2b8568a4e7c7938c08



References for BookNest Project:



Technologies & Frameworks

1. React.js Documentation

<https://react.dev/>

Official documentation for React library used in frontend development.

2. Node.js Documentation

<https://nodejs.org/en/docs>

Server-side JavaScript runtime used for backend logic.

3. Express.js Documentation

<https://expressjs.com/>

Web framework for Node.js used to build RESTful APIs.

4. MongoDB Documentation

<https://www.mongodb.com/docs/>

NoSQL database used for storing book, user, and order data.

5. Mongoose.js Documentation

<https://mongoosejs.com/docs/>

MongoDB ODM used to define schemas and interact with the database.

6. **Redux Toolkit Documentation**
<https://redux-toolkit.js.org/>
For managing global state and handling side effects (cart, book data, etc.).
 7. **RTK Query Documentation**
<https://redux-toolkit.js.org/rtk-query/overview>
For efficient data fetching and caching with Redux Toolkit.
 8. **Firebase Authentication**
<https://firebase.google.com/docs/auth>
Used for implementing secure login and registration.
 9. **Tailwind CSS Documentation**
<https://tailwindcss.com/docs>
Utility-first CSS framework used for responsive design.
 10. **React Router DOM**
<https://reactrouter.com/en/main>
Library for client-side routing in React.
-

Video Tutorials / Learning Resources

1. **MERN Stack Full Project Tutorials**
(*YouTube – e.g., Clever Programmer, freeCodeCamp, Code With Antonio*)
Practical guidance for building full-stack applications.
(*Include exact URL if you followed a specific one*)
 2. **Firebase Auth + React Setup Tutorial**
<https://www.youtube.com/watch?v=PKwu15ldZ7k>
Guide for integrating Firebase Auth with React.js.
 3. **Deploying MERN Stack App on Vercel and Render**
<https://www.youtube.com/watch?v=71wSzpLyW9k>
Step-by-step deployment tutorial.
-

Other Supporting Resources:

1. **JWT (JSON Web Tokens) Introduction**
<https://jwt.io/introduction>
For implementing secure token-based authentication.

2. MongoDB Atlas

<https://www.mongodb.com/cloud/atlas>

Cloud-based MongoDB used for deployment.

3. Render Hosting Platform

<https://render.com/>

For backend deployment.

4. Vercel Hosting Platform

<https://vercel.com/>

For frontend (React) deployment.
