

By

Varapula Umadevi

Rachana palo

Gummella Hema Lakshmi

Praveen kumar Gorrela

ID:LTVIP2026TMIDS86553



# BOOKNEST: WHERE STORIES NESTLE

# INTRODUCTION



BOOK NEST is an app for book lovers, a charming community for those who love to read e-books. Readers can find favourite book. Also, authors can share their stories with the book nest community.

BookNest is a comprehensive full-stack web application developed using the PERN stack (PostgreSQL, Express.js, React.js, and Node.js). It functions as an online book shop where users can effortlessly browse, search, and purchase a wide array of books. Additionally, BookNest features a dynamic blogging platform that allows users to read and create posts on various literary topics, enhancing the community aspect of the platform.

# PROJECT OVERVIEW

## FEATURES:

**User Authentication:** Secure user registration and login using JWT.

Bookstore: Browse, search, and purchase books with ease.

**Blog Section:** Read and create blog posts about books, authors, and literary trends.

**User Profiles:** Personalized user dashboard displaying user information and activity.

**Shopping Cart:** Add books to the cart and manage purchases seamlessly.

**Reviews and Ratings:** Users can submit reviews and ratings for books.

**Admin Dashboard:** Manage books, posts, and user data efficiently

# ARCHITECTURE

## TECHNOLOGIES USED

### Frontend:

React.js  
React Router  
Axios for API requests  
Tailwind for styling

### Backend:

Node.js  
Express.js  
JWT for authentication  
Multer for file uploads

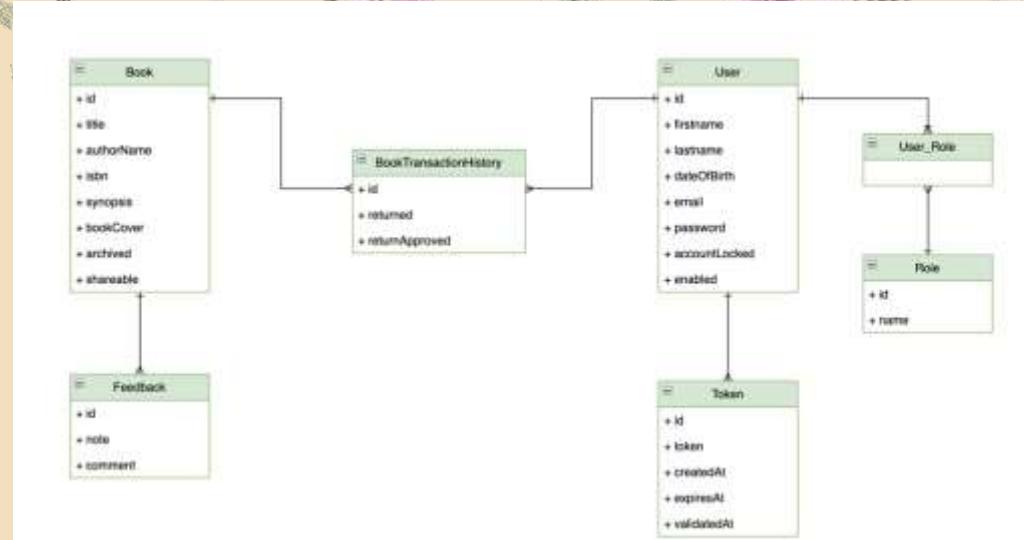
### Database:

PostgreSQL

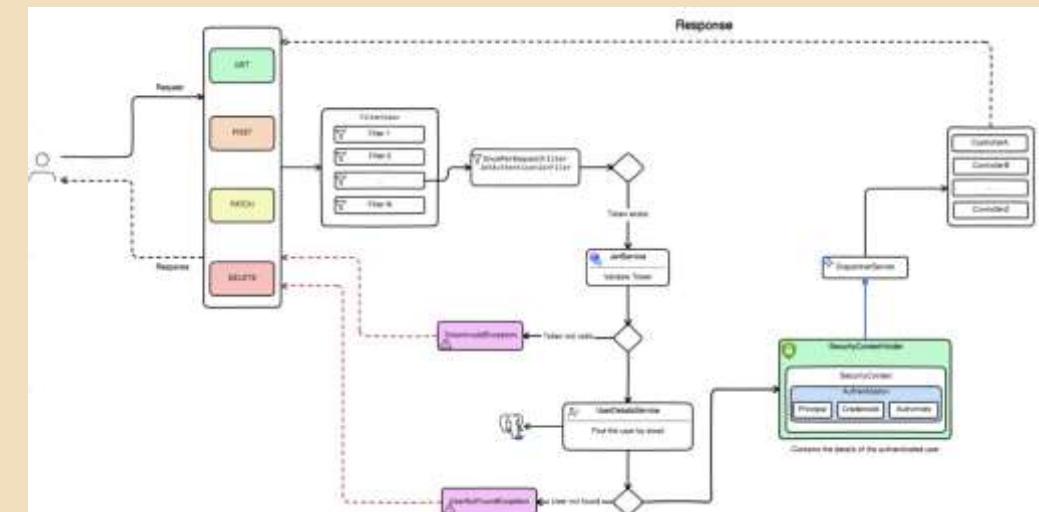
### Environment Variables:

dotenv for managing sensitive data

Spring security diagram



Class diagram



# INSTALLATION

Follow these steps to setup the BookNest application on your local machine:

**1. Clone the repository:** `git clone https://github.com/Manish312002/-BOOKNEST-Ultimate-Online-Bookstore-and-Literary-Community.git` cd -BOOKNEST-Ultimate-Online-Bookstore-and-Literary-Community.

**2. Install backend dependencies:** `cd server npm i`

**3. Install frontend dependencies:** `cd ..\client npm i`

**4. Set up environment variables:** Create a `.env` file in the backend directory with the following content:

`PG_USER=your_username`

`PG_PASSWORD=your_password`

`PG_DATABASE=your_database`

`PG_HOST=localhost`

`PG_PORT=5432`

`SECRET=your_jwt_secret`

**5. Run the backend server:** `cd ..\server nodemon server.js`

**6. Run frontend server:** `cd ..\client npm run dev`

# FOLDER STRUCTURE



## 1. Client (React frontend):

Typical React folder layout:  
client directory

`src/` → components, pages, assets, App.js, index.js.  
\*\*npm start\*\*

`public/` → index.html, favicon.

`package.json` for dependencies & scripts.  
directory

## 2. Server (Node.js backend):

time)then npm start(or

- Typical Node.js folder layout:

`routes/` → API endpoint handlers.

`controllers/` → business logic.

`models/` → database schemas (Mongoose/Sequelize).

`app.js` or `server.js` → main server file.

# RUNNING APPLICATION

## 1.Frontend:

open a terminal in the

Run npm install(first time) then

## 2.Backend:

open a terminal in the client

Run npm install(first

```
# Start backend  
cd BookNest/backend  
npm start
```

if configured).

```
# Start frontend  
cd BookNest/frontend  
npm run dev
```

# API DOCUMENTATION

## User Authentication:

POST /register: Register a new user, POST /login: Log in an existing user , GET /profile: Get user profile information

## Books:

GET /books: Get a list of books , POST /upload-book: Add a new book , PATCH /book/update/:id: Update book details , DELETE /book/delete/:id: Remove a book

## Blog Posts:

GET /posts: Get all blog posts , POST /dashboard/upload-post: Create a new blog post , PATCH /post/update/:id: Update a blog post , DELETE /post/delete/:id: Delete a blog post

## Cart Management:

GET /shop/cart: Get the user's shopping cart , POST /shop/cart: Add an item to the cart , DELETE /cart/delete/:id: Remove an item from the cart

## Reviews:

GET /reviews: Get all reviews , POST /review: Submit a review for a book

## Dashboard:

GET /dashboard/user: Get the user , GET /dashboard/books: Get user uploaded books , GET /dashboard/posts: Get user uploaded posts

POST /dashboard/logout: Log out user

# AUTHENTICATION

- Secure user registration and login using JWT.

**Bookstore:** Browse, search, and purchase books with ease.

**Blog Section:** Read and create blog posts about books, authors, and literary trends.

**User Profiles:** Personalized user dashboard displaying user information and activity.

**Shopping Cart:** Add books to the cart and manage purchases seamlessly.

**Reviews and Ratings:** Users can submit reviews and ratings for books.

**Admin Dashboard:** Manage books, posts, and user data efficiently

# TESTING

## 1. Frontend (React)

- \*Unit tests\*: Jest + React Testing Library (test components, hooks).
- \*Integration tests\*: Test API calls, interactions with Cypress/Playwright.
- \*Example\*: Test book search, add-to-favorites flow.

## 2. Backend (Node.js + DB)

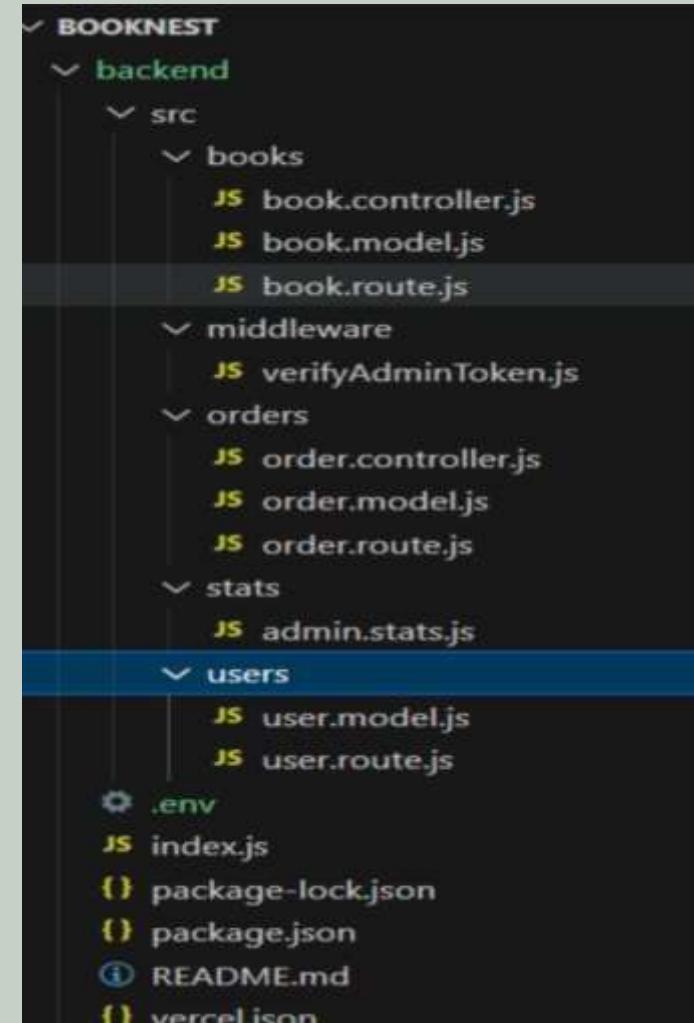
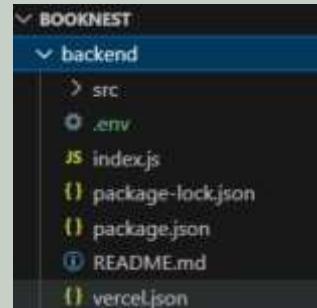
- \*Unit tests\*: Jest (test services, models, utils).
- \*Integration tests\*: Supertest (test API endpoints).
- \*Example\*: Test `POST /books`, `GET /books/:id`.

## 3. E2E Testing

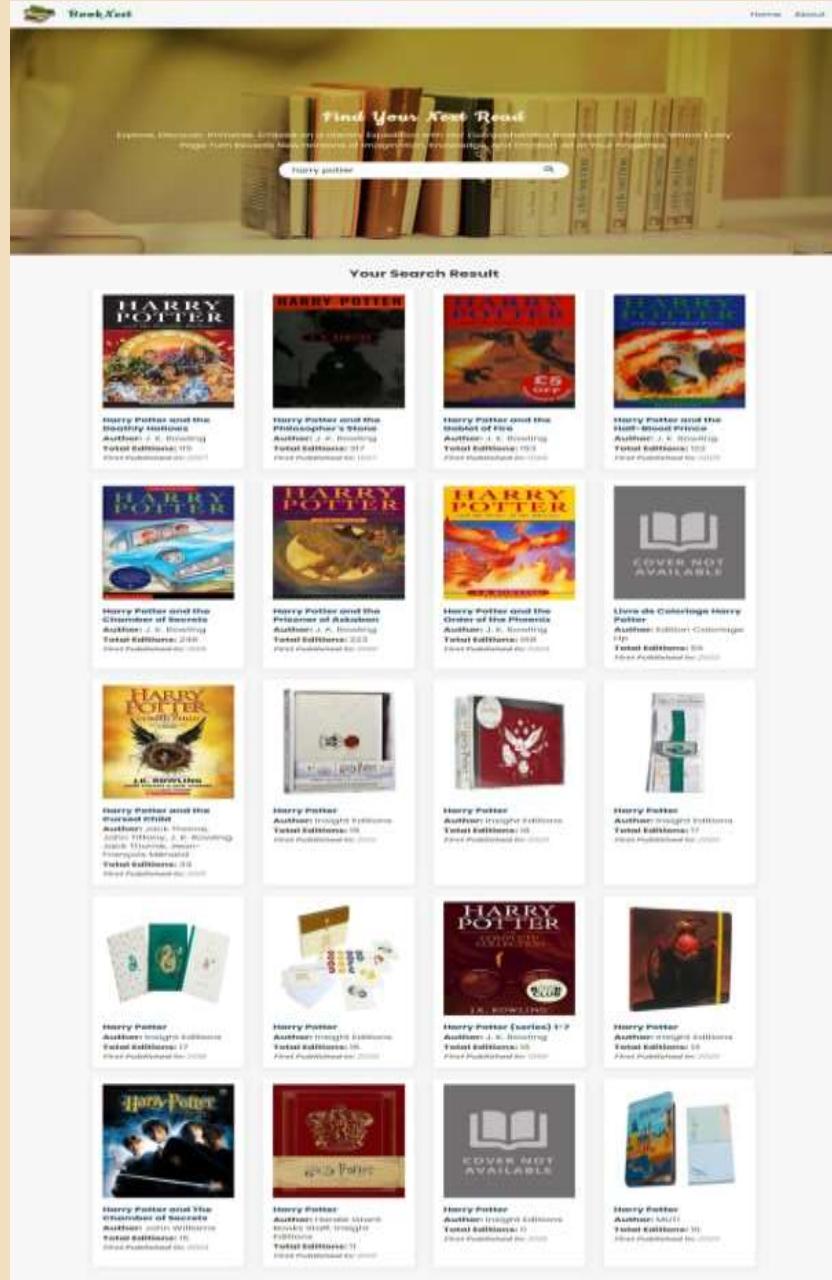
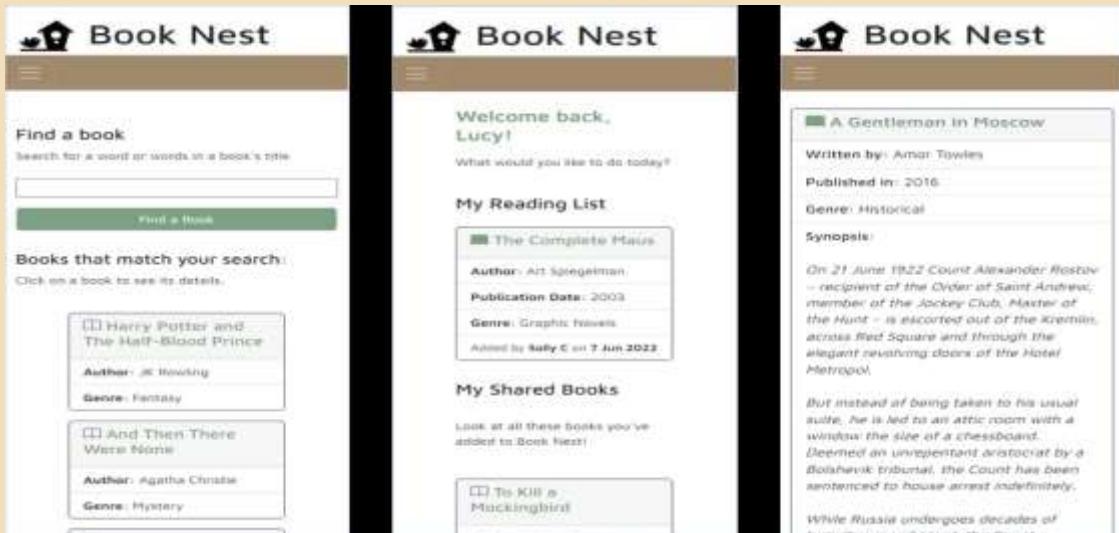
- Cypress: Test full flows (login, add book, checkout).

## Tools Used

- \*Frontend\*: Jest, React Testing Library, Cypress.
- \*Backend\*: Jest, Supertest.
- \*DB\*: Mock data or in-memory DB (e.g., Jest mocks for DB).
- \*CI/CD\*: GitHub Actions for running tests on PRs.



# SCREENSHOTS AND DEMO



# FUTURE SCOPE



**User interface:** margins on nav buttons to stop pushing out, more colour contrast

Manage when user tries to add a book already in their reading list (e.g. no "add to reading list button")

Manage when user tries to remove a book already in their reading list (e.g. no "remove from reading list" button)

Retain text in search so user sees their typed search

Link book record on reading list to its ebook

User checkout to read or reserve an ebook

Delete a book record

Edit a book record

Add images to a book record

Enhanced context-sensitive navigation

Notifications (e.g. when a new book in a favourite genre is shared)

Login with social media account



# CONCLUSION



BookNest provides a robust platform for book lovers to explore literature and engage with a community of like-minded individuals. With user-friendly features and a modern tech stack, it's the perfect place to buy books and share literary insights. Enjoy your reading journey!

# DEMONSTRATION

[https://drive.google.com/file/d/1YxCG12jB3P65poJ1TCLed2GKD0H2ao7r/view?usp=drive\\_link](https://drive.google.com/file/d/1YxCG12jB3P65poJ1TCLed2GKD0H2ao7r/view?usp=drive_link)

THANK YOU

