Project Documentation of BookNest: Where Stories Nestle

Project Documentation of BookNest: Where Stories Nestle

Overview

BookNest is a full-stack book store application that enables users to browse, select, and purchase books in an online environment. Built using the powerful MERN stack (MongoDB, Express.js, React.js, Node.js), it combines seamless UX, authentication, and efficient inventory and order management. Designed for book lovers like Sarah—who need fast, personalized access to literature without visiting a physical store—BookNest transforms traditional book buying into a digital delight.

Tech Stack

Layer	**Technology**
Frontend	React.js, Axios, Vite, Bootstrap
Backend	Node.js, Express.js
Database	MongoDB with Mongoose
Auth	JWT, bcrypt
Environment	.env variables for secure configs

Project Structure booknest-app/

```
├── backend/
|   ├── server.js
|   ├── routes/
|   ├── models/
|   └── .env
└── frontend/
    ├── public/
    └── src/
        ├── components/
        ├── pages/
```

└── main.jsx

MongoDB Models

User Model

{

name: String, email: String,

password: String, role: String

// 'user' or 'admin'

}

Book Model

{

title: String, author:

String, genre: String,

description: String,

price: Number,

availability: Boolean,

rating: Number

}

Order Model

{

userId: ObjectId, items: [{ bookId: ObjectId,

quantity: Number }], totalPrice: Number,

status: String, // 'Pending', 'Completed'

createdAt: Date

}

Environment Variables

.env file in /backend/

MONGO_URI=mongodb+srv://<username>:<password>@cluster0.mongodb.net/booknestDB

JWT_SECRET=yourJWTSecret

PORT=4000

REST API Endpoints

Auth

- POST /api/register – Register user
- POST /api/login – Authenticate

Books

- GET /api/books – List all books
- GET /api/books/:id – View book details
- GET /api/books?genre=:name – Filter by genre

Orders

- POST /api/orders – Place an order
- GET /api/orders/user/:id – View user's order history
- PUT /api/orders/:id – Update status (admin)

All sensitive routes require JWT authentication.

How to Run

Backend cd

npm install

npm start

Frontend cd

npm install

npm run dev

Completed Features

- User registration and login
- Browse books by category, author, and rating
- Add to cart and purchase
- View order history
- MongoDB integration with secure CRUD
- JWT authentication
- Responsive UI using Vite + Bootstrap

- Protected routes with middleware

Future Features

- Wishlist integration
- Rating and review system
- Admin book upload dashboard
- Real-time stock alerts
- Search with auto-suggestions
- Email notifications on purchase

🧪 Testing Strategy

✅ Manual Testing

- **Frontend**:
  - Verified component rendering across pages (Book listing, Cart, Login, etc.)
  - Checked responsiveness on desktop, tablet, and mobile devices
  - Validated error states (e.g., login failure, out-of-stock items)
- **Backend**:
  - Used Postman to test API endpoints individually
  - Verified JWT protection by accessing restricted endpoints with and without tokens
  - Monitored console logs and server responses for consistency

---

🧠 Design Philosophy

BookNest focuses on **user-centric design** with clean UI, fast navigation, and a consistent browsing experience. The component-based architecture ensures modularity, making it easy to scale or customize. Responsive layouts adapt gracefully to different screen sizes, ensuring inclusivity for all readers.

🔄 Flow of Operations

1. **User Journey**:
   - Registers and logs in securely
   - Browses books filtered by category or author
   - Adds selections to cart
   - Proceeds to checkout and confirms the order
   - Receives order history and status updates
2. **Admin Operations (Future Scope)**:
   - Login as admin
   - Add/update/remove books
   - View user orders and statuses

🛡 Security Measures

- **JWT Authentication**: Used to secure protected routes, ensuring user sessions remain valid and safe.
- **Password Hashing**: User passwords are hashed using bcrypt to prevent data breaches.
- **Environment Variables**: Critical configs like DB URLs and secrets are stored in .env files to prevent leakage.

Scalability & Performance

- Backend services are modular and can be split into microservices for scaling.
- MongoDB indexes optimize read operations, especially for book searches.
- Vite ensures blazing-fast frontend reload and builds.
- Ready for deployment on platforms like Render, Vercel, or Netlify with environment separation.

Folder Deep Dive

Backend

```
backend/
├── server.js           # Entry point
├── routes/
│   ├── authRoutes.js      # Login/Register APIs
│   ├── bookRoutes.js      # Fetch/Create books
│   └── orderRoutes.js     # Handle order processing
├── models/
│   ├── User.js
│   ├── Book.js
│   └── Order.js
└── middleware/
    ├── authMiddleware.js  # Route protection
    └── errorHandler.js    # Central error handling
```

Frontend

```
frontend/
├── src/
│   ├── components/
│   │   ├── BookCard.jsx
│   │   ├── Cart.jsx
│   │   └── Navbar.jsx
│   ├── pages/
│   │   ├── Home.jsx
│   │   ├── Login.jsx
│   │   ├── Register.jsx
│   │   └── OrderHistory.jsx
│   └── main.jsx         # Root file with routing
```

Conclusion

BookNest blends design and functionality to offer a seamless literary experience. With an extensible architecture and forward-thinking tech stack, it’s poised for growth, scalability, and future enhancements — where readers meet their next favorite story at the click of a button.
