JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BOOK STORE - USING MERN STACK

Team Leader : SANJAIKUMARAN .P (310821104083)
Team Member : ARUL MC MILLAN C.X (310821104013)
Team Member : RAHUL . R (310821104073)
Team Member : PAAUL ANTONY . P (310821104066)

SUPERVISOR NAME : MRS.D.JEEVITHA .M.E
ASSISTANT PROFESSOR /CSE

PROJECT OVERVIEW:

This project is a full-stack Book Store application that enables users to browse, purchase, and manage books, while also allowing sellers to list their items for sale. The application leverages React for the frontend, Node.js and Express for the backend, and MongoDB for database management. It includes user authentication and authorization to protect user data and ensure secure access. Key features include user and seller registration, item management, order handling, and wishlist, functionality.

Architecture

- **User**
  - Account creation (sign up/login).
  - View items and make purchases.
  - View and manage orders.

- **Seller**
  - Add and manage product listings.
  - View orders related to their products.
  - Update item details.

- **Admin**
  - Manage users and sellers.
  - View all orders and items.
  - Delete or update listings.

SETUP INSTRUCTION:

- To set up this project, clone the repository and navigate to the backend folder, then run npm install to install backend dependencies. Next, do the same for the frontend by navigating to the frontend folder and running npm install.

- For the backend, run npm start to launch the server, and for the frontend, run npm run dev to start the development server.

- Ensure MongoDB is running locally or connected to a cloud service for the database functionality.

FOLDER STRUCTURE:

- The project has a frontend folder containing React components, styling, and configuration files for the user interface.
- The backend folder includes API routes, models, controllers, and middleware for handling authentication, authorization, and database interaction.
- The uploads folder stores images and files uploaded by users or sellers.
- Configuration files like package.json and .env are placed at the root for managing dependencies and environment variables

Frontend Development

- **Features**: Responsive design, dynamic routing with React Router, and a user-friendly interface.
- **Purpose**: Provides a seamless user experience for browsing books, managing accounts, and performing transactions.
- **Technology Used**: React.js,

Backend Development

- **Technology Used**: Node.js with Express.js.
- **API Management**: RESTful APIs handle user authentication, book data, orders, and seller operations.
- **File Handling**: Integrated with Multer for managing file uploads such as book images.

Database

- **Technology Used**: MongoDB.
- **Structure**: Collections for Users, Books, Orders, Sellers, and Wishlist.
- **Purpose**: Efficiently stores and retrieves data for the application's core functionalities

Key Functionalities

- User registration, login, and profile management.
- Adding, browsing, and purchasing books.
- Seller management tools for book listing and order tracking.
- Wishlist and order history for users.

Book Store Application:

The frontend and backend components must be executed seamlessly.

Start by navigating to the backend directory and running the command npm install to install dependencies, followed by npm start to launch the server.

For the frontend, navigate to the respective directory, run npm install to set up the required modules, and use npm run dev to start the development server.

Once both the frontend and backend are active, the application becomes fully functional and ready to serve user requests.

API DOCUMENTATION:

The backend API exposes several endpoints for managing users, sellers, items, and orders. Key endpoints include:

POST /signup: Registers a new user.

GET /items: Fetches all available items.

POST /items: Adds a new item for sale by the seller.

GET /orders: Retrieves orders for a specific user or seller.

PUT /items: Enter the books into collections

DELETE /Items: Remove from wishlist / my orders

Other endpoints are available for handling item deletion, order creation, and managing the wishlist , with role-based authorization in place for security.

Authentication:

- User login where credentials (email and password) are verified against the database.
- Session or token is issued to identify the user.
- Authorization ensures resources access
- Secure Authentication: JWT is used to authenticate users by storing user information in a token, which is sent with each request to validate the user's identity.
- Stateless and Scalable: JWT is stateless, meaning the server doesn't need to store session data, making the application more scalable and efficient

MongoDB Database Overview

MongoDB is a powerful, open-source, NoSQL database designed for modern applications. Unlike traditional relational databases, it stores data in flexible, JSON-like documents, allowing for dynamic and schema-less structures. This makes MongoDB ideal for handling unstructured or semi-structured data.

Key Features of MongoDB:

1.Scalability
2.Flexibility
3.High Performance
4.Rich Query Language

Testing

Puppeteer: Automates browser interactions to test user workflows.

Cucumber: Ensures features match requirements with BDD approach.

Steps in Automation

1. Write feature files for each role's scenarios (e.g., Seller adds a book).
2. 2. Use Puppeteer in step definitions to interact with UI elements.
3. 3. Execute tests and ensure expected results match outputs.

localhost:5173/uproducts

BookStore

Home Books Wishlist My orders Logout (Sanjai Customer)

Books List

Mister Magic

Author: Kiersten White

Genre: Thriller Audiobook Adult Horror Mystery Thriller Paranormal Adult Fiction

Price: $400

Remove from Wishlist View

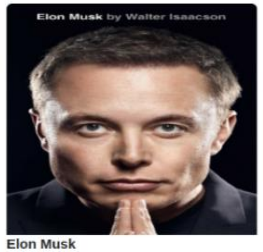
River Sing Me Home

Author: Eleanor Shearer

Genre: Historical Fiction Fiction Historical Audiobook Literary Fiction Race Adult

Price: $330

Add to Wishlist View



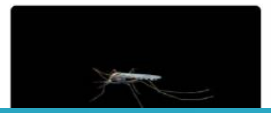
Elon Musk

Author: Walter Isaacson

Genre: Biography

Price: $440

Add to Wishlist View

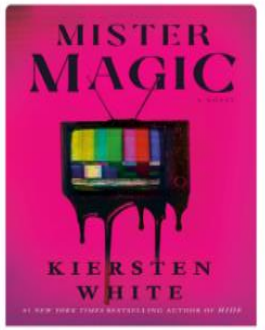


localhost:5173/wishlist

BookStore

Home Books Wishlist My orders Logout (Sanjai Customer)

Wishlist



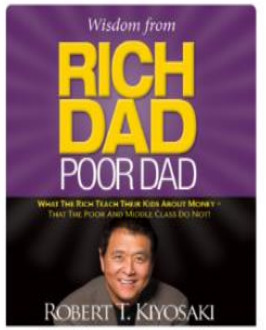
Mister Magic

Author:

Genre:

Price: $

Remove from Wishlist View



Rich Dad Poor Dad

Author:

Genre:

Price: $

Remove from Wishlist View

localhost:5173/myorders

BookStore

Home Books Wishlist My orders Logout (Sanjai Customer)

My Orders

ProductName:	Orderid:	Address:	Seller	BookingDate	Delivery By	Price	Status
-8687	6738687e6b	26, ewrer,(63565), ewrew.	Sanjai Seller	16/11/2024	11/23/2024	$429	ontheway

ProductName:	Orderid:	Address:	Seller	BookingDate	Delivery By	Price	Status
-8689	673868916b	213, Neyvell,(600119), Tssmil.	Sanjai Seller	16/11/2024	11/23/2024	$654	ontheway

Contact us

"Embark on a literary journey with our book haven – where every page turns into an adventure!"

Call At: 127-865-586-67

localhost:5173/shome

BookStore(Seller)

Home Myproducts Add Books Orders Logout (Sanjai Seller)

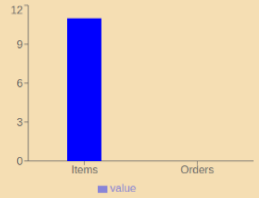
DashBoard

Items

11

Total Orders

0



Contact us

"Embark on a literary journey with our book haven – where every page turns into an adventure!"

Call At: 127-865-586-67

KNOWN ISSUES :

Cross-Origin Resource Sharing (CORS) Errors:

If backend APIs are not correctly configured to handle requests from the frontend, CORS errors may arise during data fetching.

Image Upload Challenges:

Issues with file paths or storage configurations may lead to image upload failures.

Session Handling:

There might be limitations in maintaining user sessions, causing users to log in repeatedly.

Deployment Bugs:

While the project may work locally, deploying to a live server may reveal unexpected platform-specific issues.

Future Improvements & Conclusion

- **Potential Enhancements:**
  - **Payment Gateway Integration: Adding a payment system for seamless transactions.**
  - **Search and Filters: Improve item search functionality (by category, price, etc.).**
  - **User Reviews: Allow users to leave ratings and reviews on items.**
  - **Mobile Optimization: Enhance mobile responsiveness and UI.**

Thank You