**BookNest: MERN Bookstore Application - Report**

**1. Introduction**

**Project Title:** BookNest: MERN Bookstore Application

**Team Members:**

* Jithin.M
* Sudhakar.R
* Jashwanth.M
* Karan Singh

**2. Project Overview**

**Purpose:**  
BookNest is a modern full-stack bookstore application designed to enhance the experience of browsing, purchasing, and managing books. Built using the MERN stack, it focuses on providing a user-friendly interface for customers and robust administrative tools for store management.

**Features:**

1. **Book Catalog**: Dynamic browsing of books displayed with search and filter options.
2. **Book Details**: Detailed information, including descriptions, pricing, and authorship.
3. **Cart Functionality**: Add books to a cart and proceed to checkout seamlessly.
4. **Admin Dashboard**: Enables CRUD operations (Create, Read, Update, Delete) for book inventory.
5. **Authentication**: Secure login for both users and admins.
6. **Firebase Integration**: Used for project environment configuration in the frontend.

**3. Architecture**

**Frontend:**

* Developed using **React.js** with state management via Redux Toolkit.
* Key components:
  + **Book Listing**: Displays books with intuitive navigation.
  + **Cart System**: Dynamically manages cart items and calculates total costs.
  + **Authentication**: Login and signup features secured with JWT.
  + **Admin Dashboard**: Tools for inventory management.

**Backend:**

* Built with **Node.js** and **Express.js**, ensuring scalable and efficient API handling.
* Key features:
  + API Endpoints:
    - /books for book data management.
    - /orders for handling user orders.
    - /users for authentication and profiles.
  + Middleware: Ensures secure authentication and error management using **JWT**.

**Database:**

* **MongoDB** is utilized to store:
  + **Books**: Information about titles, authors, prices, and descriptions.
  + **Users**: Authentication details and purchase history.
  + **Orders**: Records of customer purchases and statuses.

**4. Setup Instructions**

**Installation:**

**Clone the repository:**

https://github.com/jithin2003/web-app.git

**Frontend Firebase Configuration**

1. Navigate to the frontend directory:

cd frontend

1. Create a .env.local file at the root level of the frontend directory and add the Firebase configuration:

VITE\_API\_KEY="Your-API-Key"

VITE\_AUTH\_DOMAIN="your-app.firebaseapp.com"

VITE\_PROJECT\_ID="your-firebase-project-id"

VITE\_STORAGE\_BUCKET="your-app.appspot.com"

VITE\_MESSAGING\_SENDERID="your-sender-id"

VITE\_APPID="your-app-id"

1. Install dependencies:

npm install

1. Start the frontend development server:

npm run dev

**Backend Setup**

1. Navigate to the backend directory:

cd backend

1. Create a .env file and include:

DB\_URL="Your MongoDB connection string"

JWT\_SECRET\_KEY="Your JWT secret"

1. Install backend dependencies:

npm install

1. Start the backend server:

npm run dev

**Access the Application**

* Open your browser and navigate to http://localhost:5173.

**5. Folder Structure**

**Client:**

* src/components: Shared components such as Navbar, BookCard, and CartItem.
* src/pages: Pages like Home, Cart, and AdminPanel.
* src/redux: Manages state for books, authentication, and cart items.

**Server:**

* routes: Defines RESTful endpoints for books, users, and orders.
* controllers: Contains the logic for handling backend requests.
* models: MongoDB schemas for Books, Users, and Orders.
* middleware: Handles authentication with JWT and error management.

**6. API Documentation**

**Endpoints:**

1. **GET /books**
   * Fetches all available books.
   * Optional parameters: Filters like author, genre, or price.
   * **Response:**

json

[

{

"id": "001",

"title": "The Great Gatsby",

"author": "F. Scott Fitzgerald",

"price": 10.99

}

]

1. **POST /orders**
   * Places a new order.
   * Parameters: User ID, book details, quantities.
   * **Response:**

json

{

"message": "Order placed successfully",

"orderId": "12345"

}

**7. User Interface**

**Screens:**

* **Home Page**: Lists all books with filters and search options.
* **Book Details Page**: Detailed information about individual books.
* **Cart Page**: Displays selected items and checkout options.
* **Admin Dashboard**: Features for adding, editing, and removing books.

**8. Known Issues**

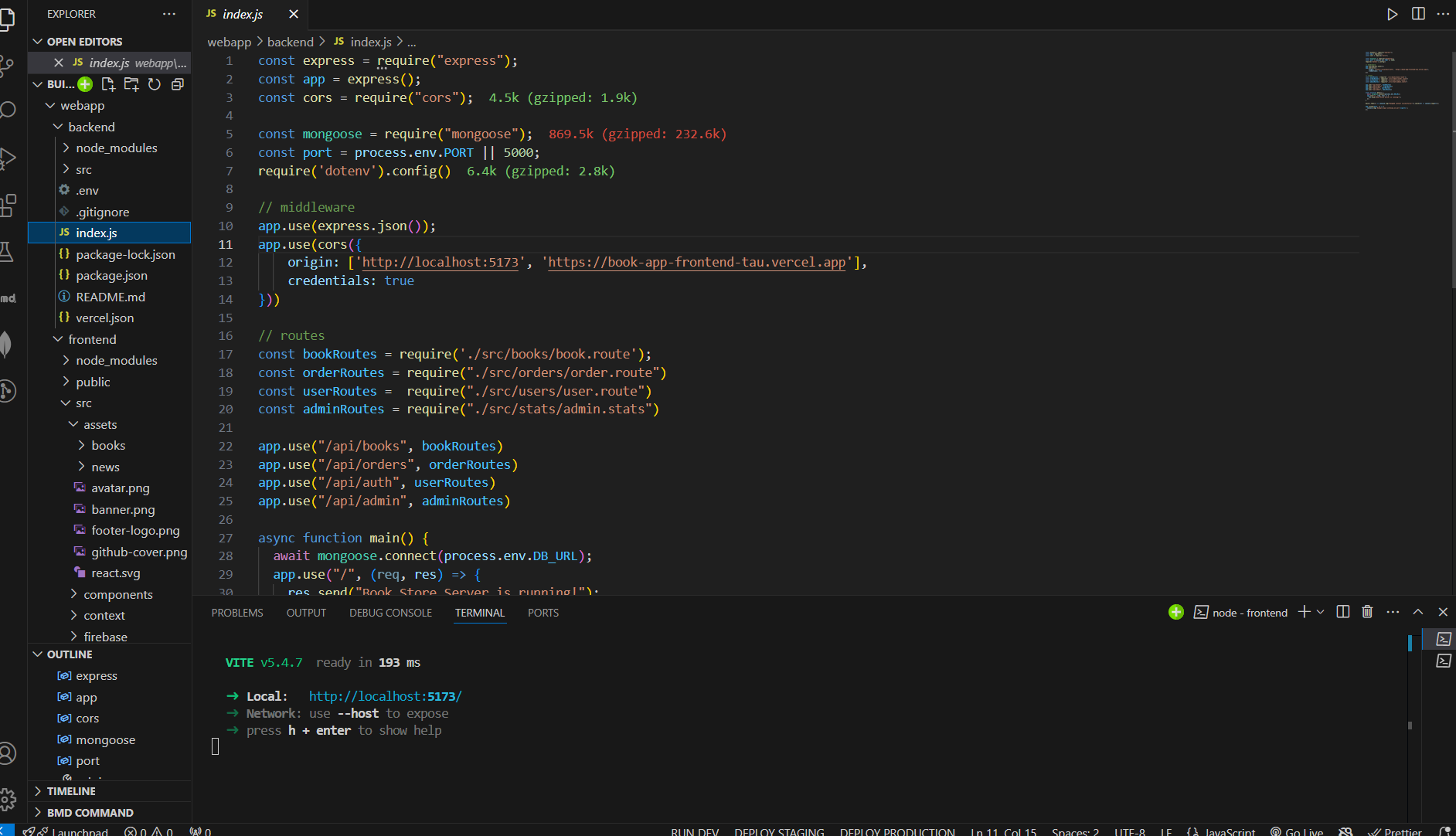
* Minor UI alignment issues on smaller screen devices.
* Backend performance optimization required for larger datasets.

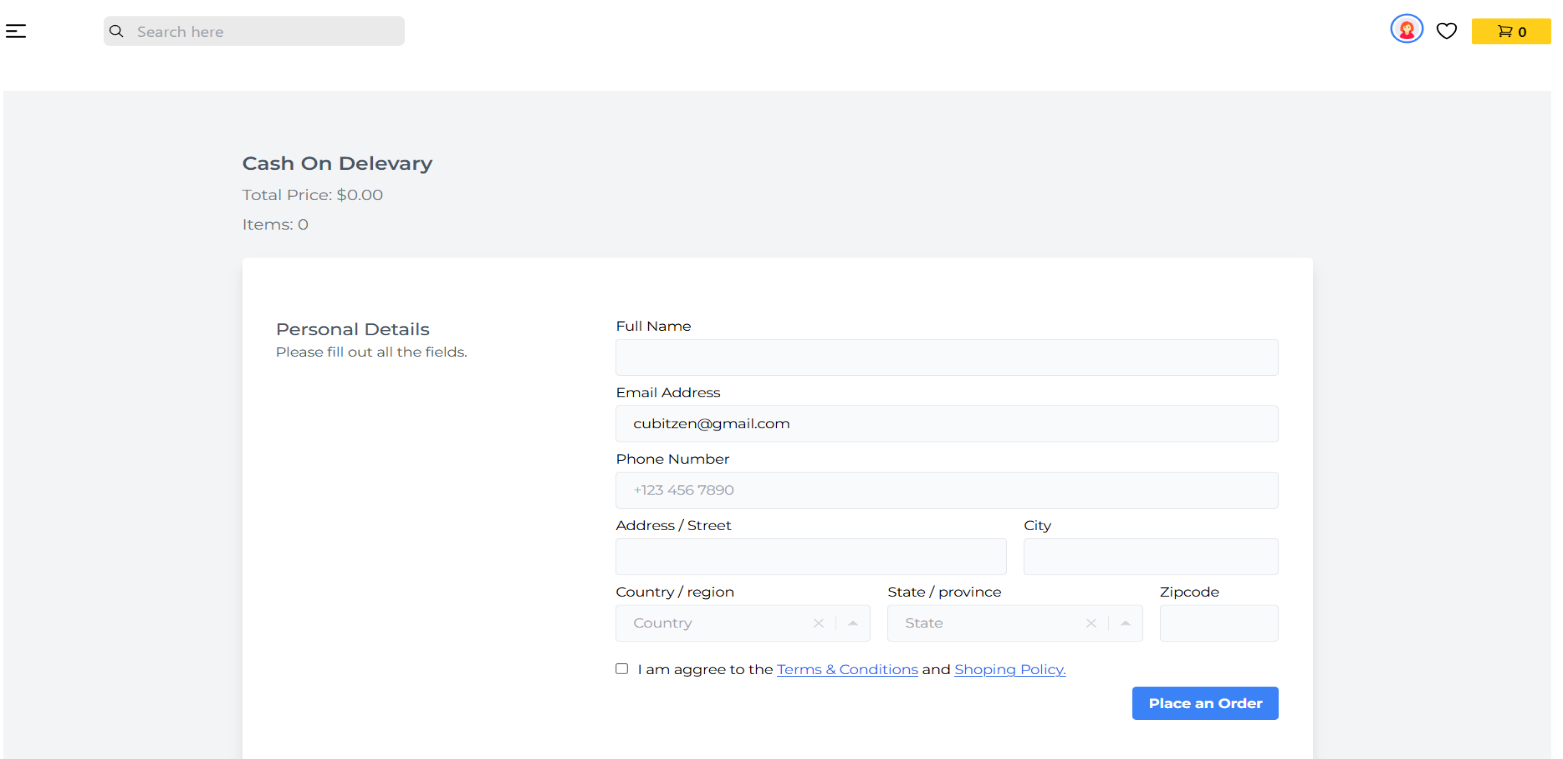
**9. Future Enhancements**

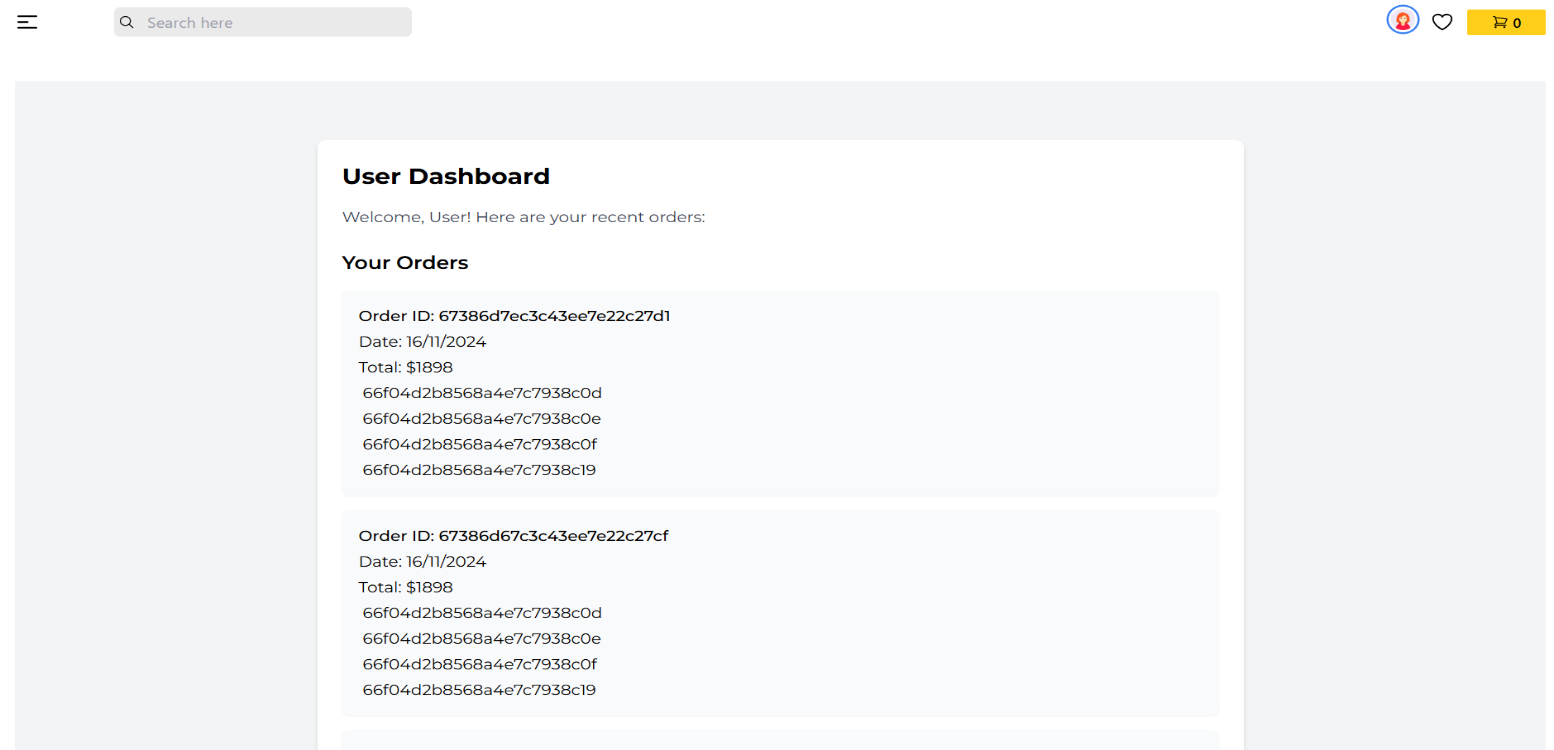
* Implement user reviews and ratings for books.
* Develop a mobile-friendly version using **React Native**.
* Enhance filtering with advanced options like genres and publication dates.

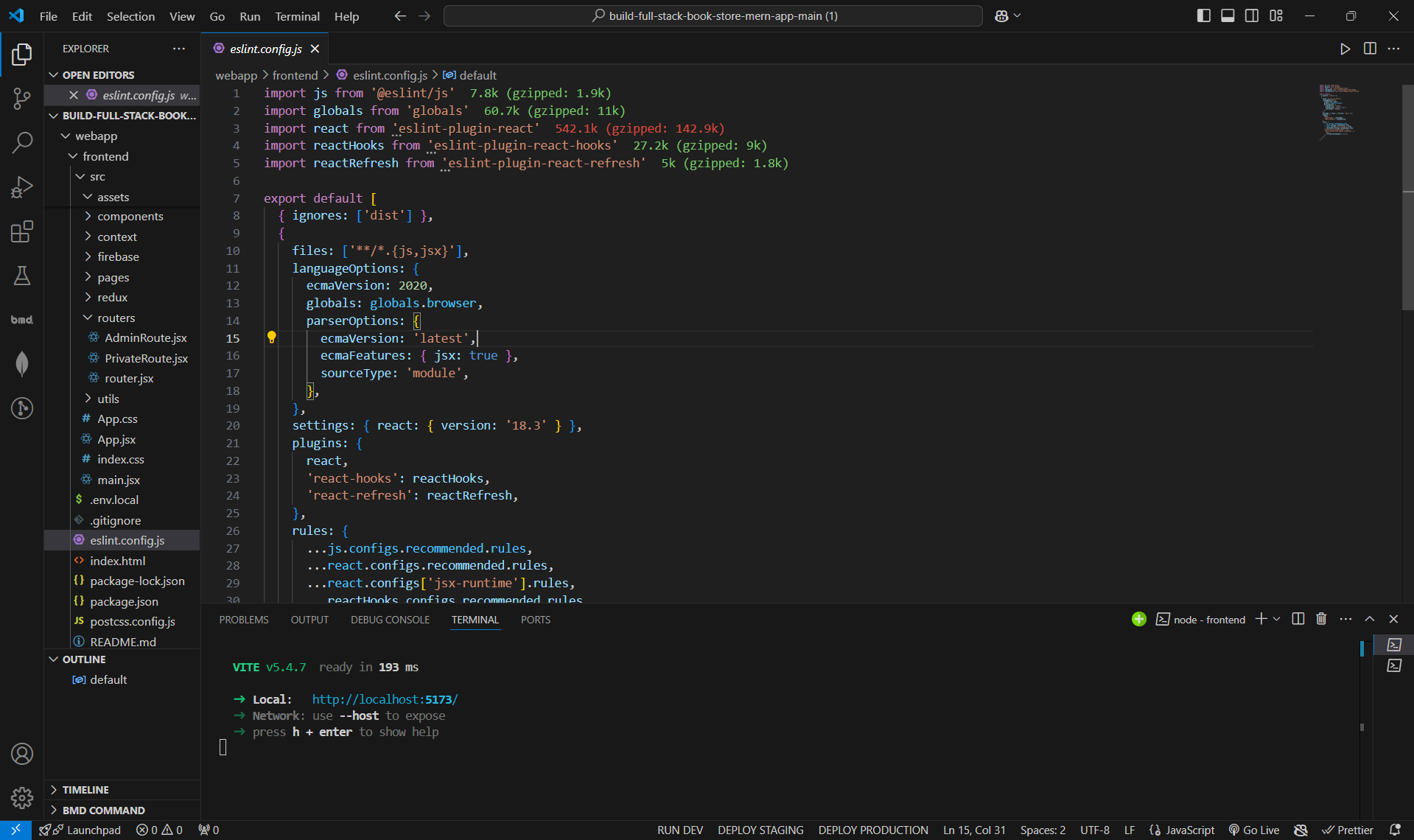
**10. Screenshots or Demo**

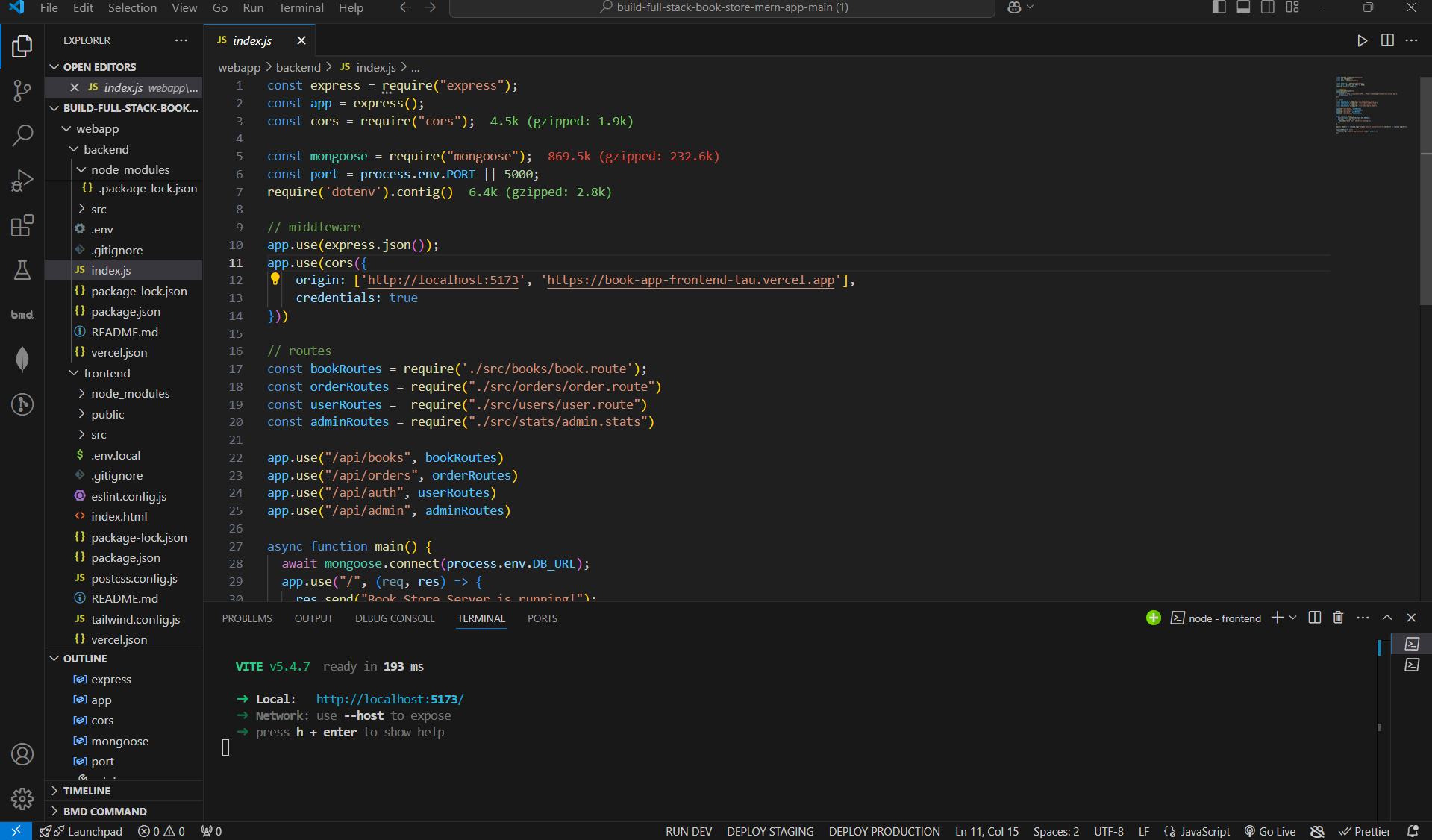
Screenshots of key UI components and admin panel

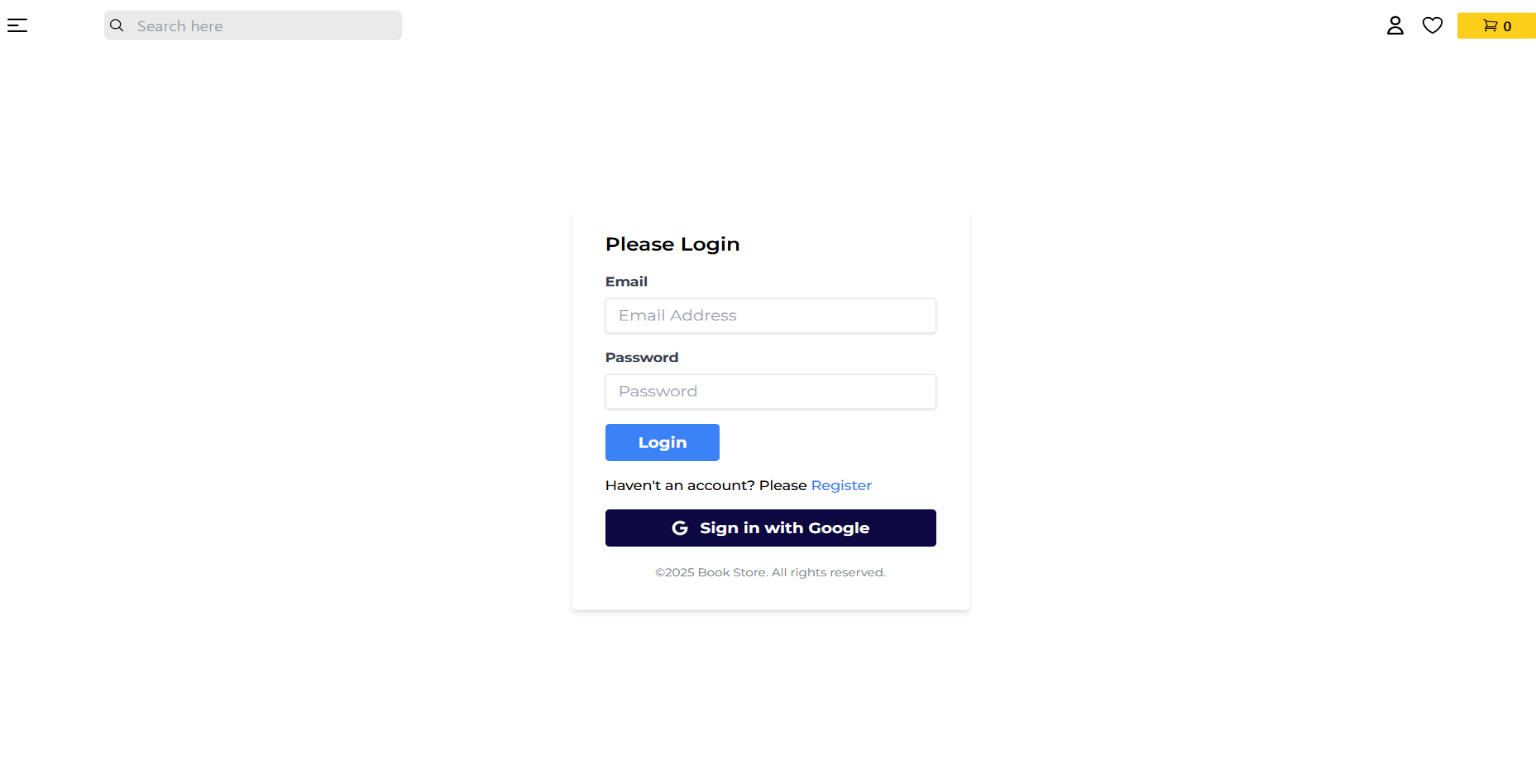


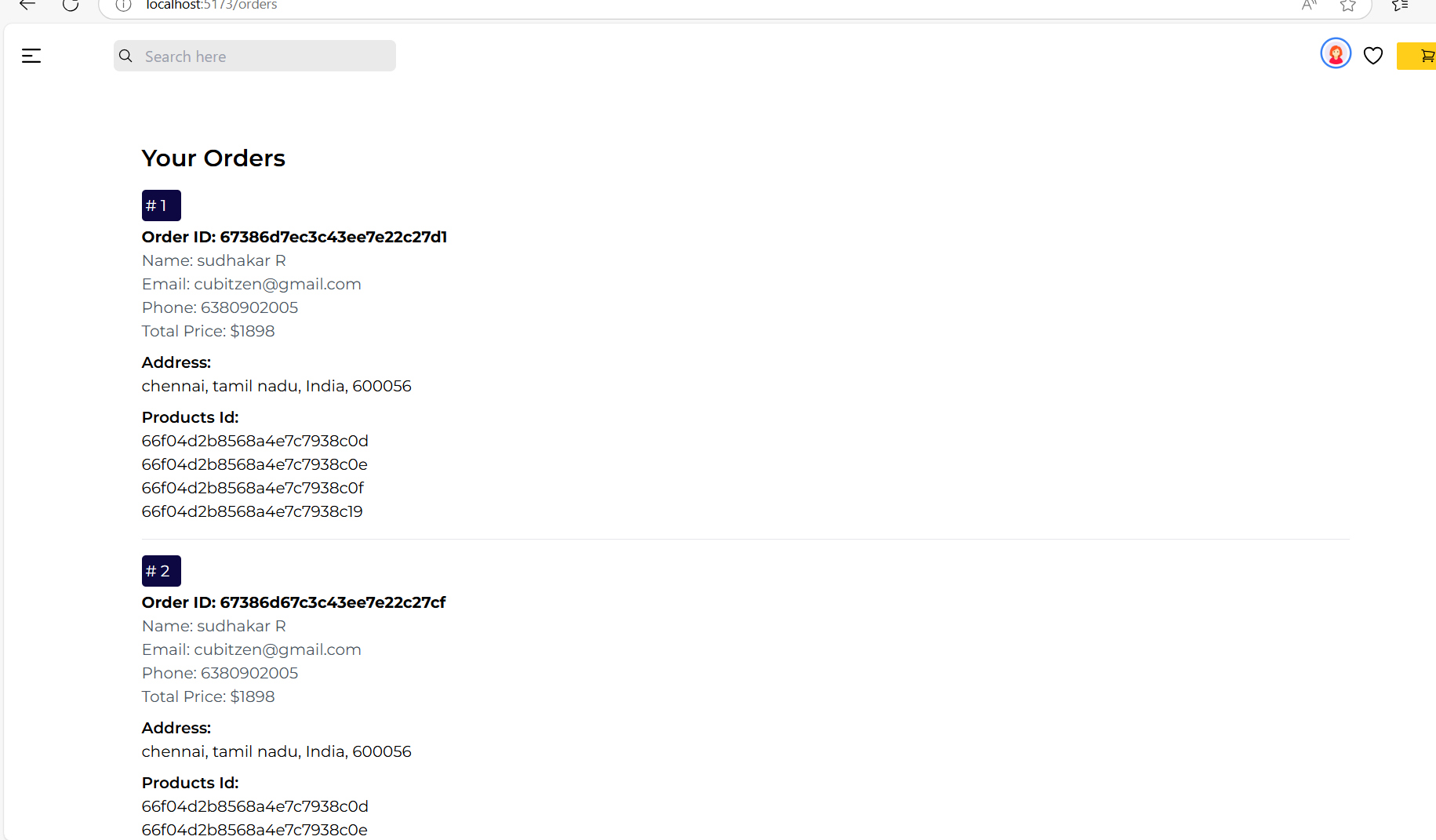


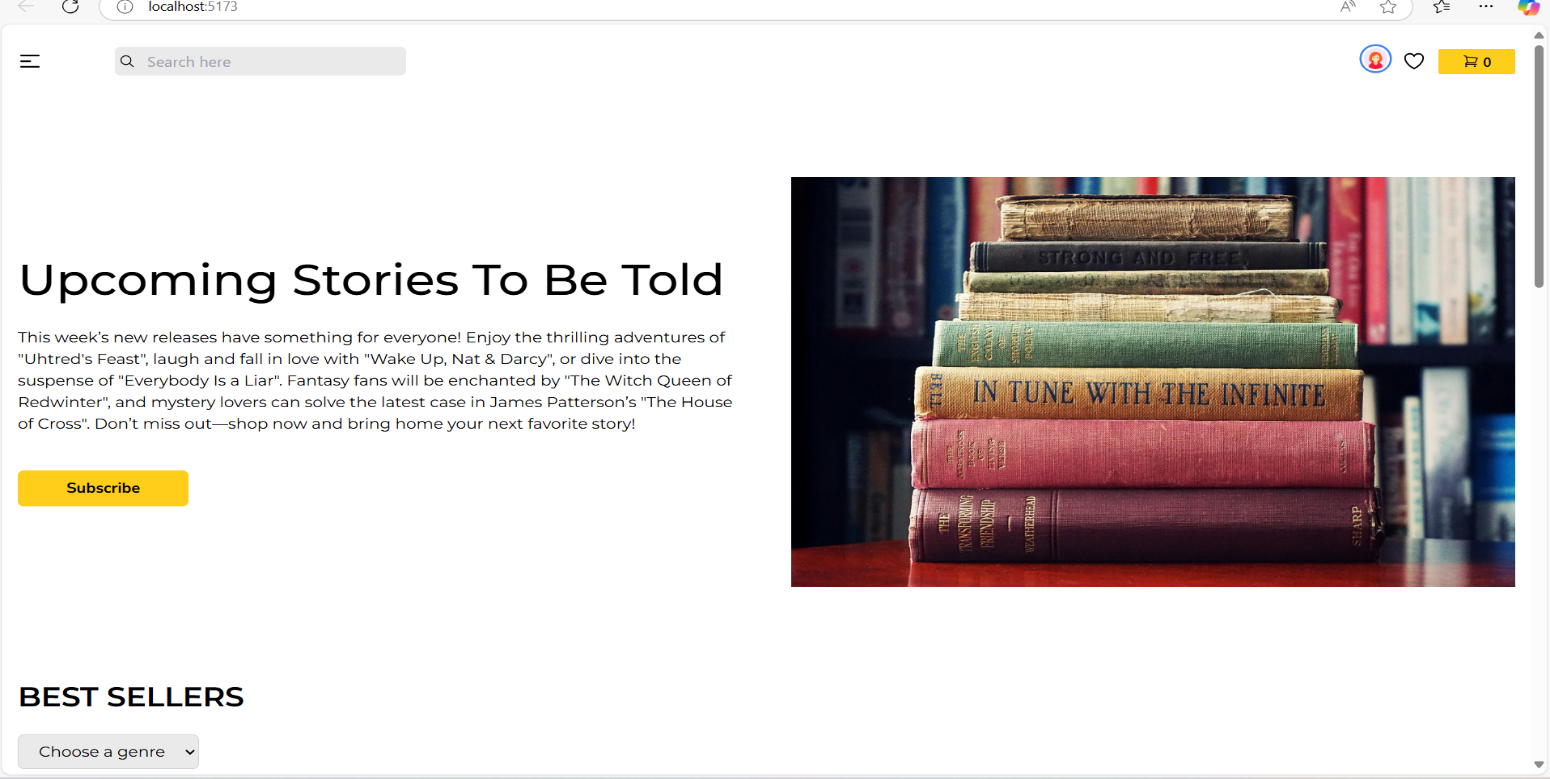












Link to a live demo.

<https://drive.google.com/file/d/1G5oS6oXoLMOBQt1Q8aVgLeuV11vuJzYI/view?usp=drivesdk>