**BOOK STORE Application**

**A PROJECT REPORT**

***Submitted by***

**ABINAYA SHREE D 210121104002**

**KESAVA MOORTHI A 210121104025**

**ANTO JERIN JOSHVA A 210121104009**

**GOKUL KANNAN K 210121104015**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING IN**

**COMPUTER SCIENCE AND ENGINEERING**



**ALPHA COLLEGE OF ENGINEERING**

**ANNA UNIVERSITY:CHENNAI-600025**

**NOVEMBER- DECEMBER 2024**



**BONAFIDE CERTIFICATE**

Certified that this project report “**Book Store Application”** is the bonafide work of **“Abinaya Shree D (210121104002) , Kesava Moorthi A(210121104025) , Anto Jerin Joshva A (210121104009) , Gokul Kannan K (210121104015)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.Jerry Alexander M.E.,(Ph.D) Mrs. S.DEEPA M.E**

**HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR**

Department of Computer Science Department of Computer And Engineering, Science And Engineering

Alpha College Of Engineering Alpha College Of Engineering Thirumazhisai-600124 Thirumazhisai-600124

This project work was submitted for viva voice held on at Alpha College Of

Engineering Subject Code/Name: NM1042 MERN Stack Powered by Mongo DB.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

|  |  |
| --- | --- |
| **S.NO** | **TABLE OF CONTENTS** |
| **1.** | **INTRODUCTION** |
| **2.** | **PROJECT OVERVIEW** |
| **3.** | **ARCHITECTURE** |
| **4.** | **SETUP INSTRUCTIONS** |
| **5.** | **FOLDER STRUCTURE** |
| **6.** | **RUNNING THE APPLICATION** |
| **7.** | **API DOCUMENTATION** |
| **8.** | **AUTHENTICATION** |
| **9.** | **USER INTERFACE** |
| **10.** | **TESTING** |
| **11.** | **SCREENSHOTS OR DEMO** |
| **12.** | **KNOWN ISSUES** |
| **13.** | **FUTURE ENHANCEMENT** |

### 1. Introduction

**Project Title:** Bookstore Application

**Team Members:**

* + Abinaya shree D
  + Kesava Moorthi A
  + Anto jerin joshva A
  + Gokul kannan K

### 2. Project Overview

## Purpose: The Book Store Project is a comprehensive e-commerce solution designed to streamline book discovery, purchase, and management. It offers a user-friendly interface with features like advanced search, filtering options, personalized recommendations, secure payment integration, and real-time order tracking. Administrators benefit from robust backend tools, including inventory management, sales analytics, and operational optimization, enabling data-driven decision-making. With scalable architecture, multi-platform compatibility, and stringent security protocols, the platform caters to diverse customer needs while driving business growth and expanding market reach.

**Features:**

1. **User Features:**

Advanced search books by Title, Author, Genre or Price.

Personalized Recommendations, AI-driven suggestion based on preferences.

1. **Admin Features:**

Inventory Management like Add, Update and Track Book stock.

Reports and Analytics Insights into sales trends and customer behaviour.

1. **Technical Features:**

Secure Payment Integration Supports cards, UPI , and Wallets.

Responsive Design Optimized for Web, Tablet and Mobile Devices.

1. **Security:**

Data Encryption protects user and payment data.

Authentication Role-Based Access with two-factor authentication.

### 3. Architecture

**Frontend:**

* Developed using **React.js** for its component-based architecture and state management capabilities.
* Features dynamic components like:
  + **Product Listings**: Displays products with sorting and filtering options.
  + **Cart Management**: Allows users to add, update, or remove items in their cart.
  + **User Authentication**: Login and registration pages with secure validation.
  + **Admin Panel**: Provides sellers with tools to manage inventory and view analytics.

**Backend:**

* Built with **Node.js** and **Express.js**, ensuring scalability and high performance.
* Features include:
  + **API Endpoints**:
    - /books for fetching product data.
    - /orders for processing customer orders.
    - /users for managing user authentication.
  + **Middleware** for error handling and authentication using **JWT**.

**Database:**

* **MongoDB** serves as the database, storing collections for:
  + **Users**: Authentication credentials, profiles, and purchase history.
  + **Products**: Information on inventory, prices, categories, and descriptions.
  + **Orders**: Details about placed orders, delivery status, and payment.

This architecture ensures modularity, scalability, and efficient data management.

**4. Setup Instructions**

**Prerequisites:**

* **Node.js**: v14 or later
* **MongoDB**: Installed locally or set up using a cloud provider like MongoDB Atlas
* **npm**: Package manager for installing dependencies

**Installation:**

**Clone the repository:**  
<https://github.com/ABINAYASHREE2204/bookstorenm>

**Navigate to the project directory:**  
cd bookstore

**Install dependencies for the backend and frontend:**

Backend:  
cd backend

npm install

Frontend:  
cd frontend

npm install

**Run the servers:**

Backend:  
node index.js

Frontend  
npm start:dev

### 5. Folder Structure

**Client:**

* src/components: Contains reusable components like **Navbar**, **ProductCard**, **CartItem**, etc.
* src/pages: Holds page components such as **Home**, **Cart**, **Checkout**, and **AdminPanel**.
* src/services: Manages API interactions for fetching and posting data.
* src/redux: Implements state management for cart items, user authentication, and order status.

**Server:**

* routes: Defines all RESTful API routes for users, products, and orders.
* controllers: Contains the logic for handling API requests and responses.
* models: Defines database schemas for **Users**, **Products**, and **Orders**.
* middleware: Handles authentication (JWT) and error management.

### 

### 6. Running the Application

**Frontend:**

Navigate to the client directory:  
cd frontend

Start the React application:  
npm start

**Backend:**

Navigate to the server directory:  
cd backend

Start the Node.js server:  
node index.js

### 

### 

### 7. API Documentation

**Endpoints:**

1. **GET /products**
   * Fetches the list of all products.
   * Parameters: Fliters like Author, Genre, Price.

Response:  
json  
{

"\_id": 1,

      "title": "How to Grow Your Online Store",

      "description": "Learn the best strategies to grow your online store in today's competitive market.",

      "category": "business",

      "trending": true,

      "coverImage": "book-1.png",

      "oldPrice": 29.99,

      "newPrice": 19.99

    }

**POST /orders**

* + Places a new order.
  + Parameters: Book ID, Book details, and Category.

Response:  
json  
{

"message": "Order placed successfully",

"orderId": "456"

}

### 8. Authentication

**JWT-based Authentication:**

* **Login**: Issues a JWT token upon successful authentication.
* **Token Validation**: Protects private routes like /orders and /admin.
* **Logout**: Invalidates the token on the client side.

### 9. User Interface

**Screens:**

* **Home:** Displays trending products and categories.
* **Book Details:** Shows detailed information about a selected item.
* **Cart:** Summarizes selected products and their quantities.
* **Admin Dashboard:** Offers order status updates and analytics for sellers.

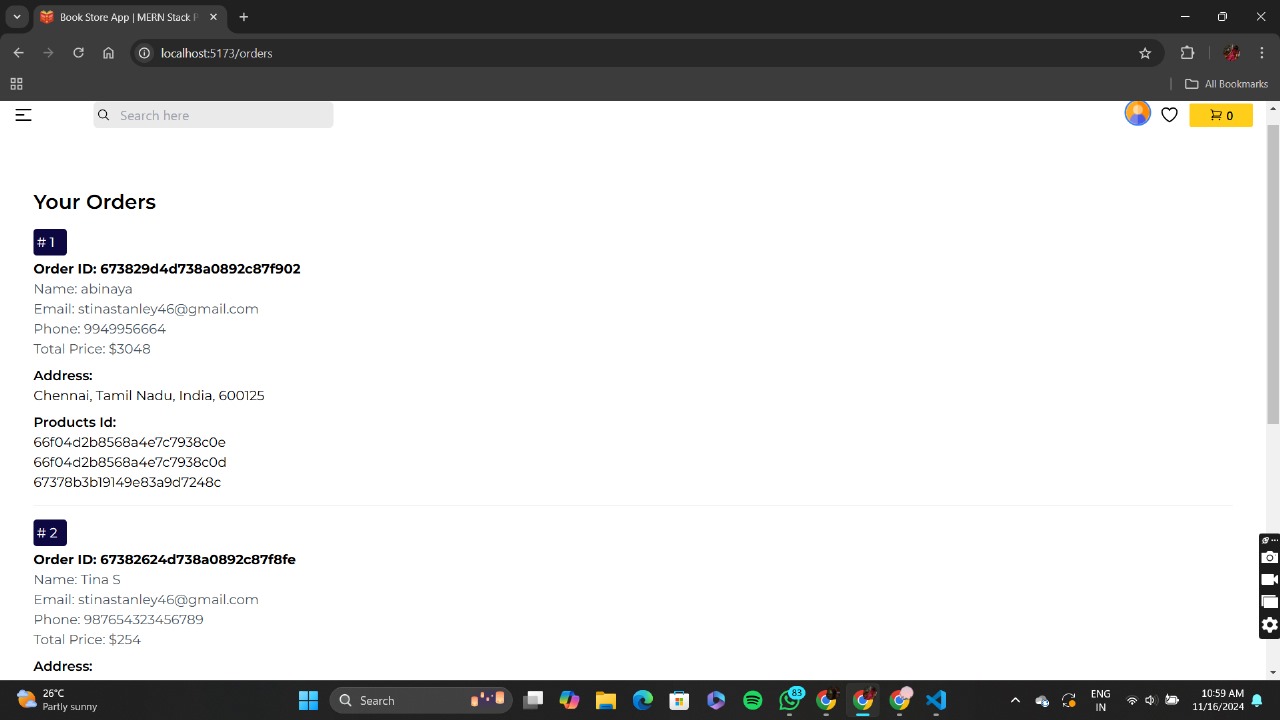
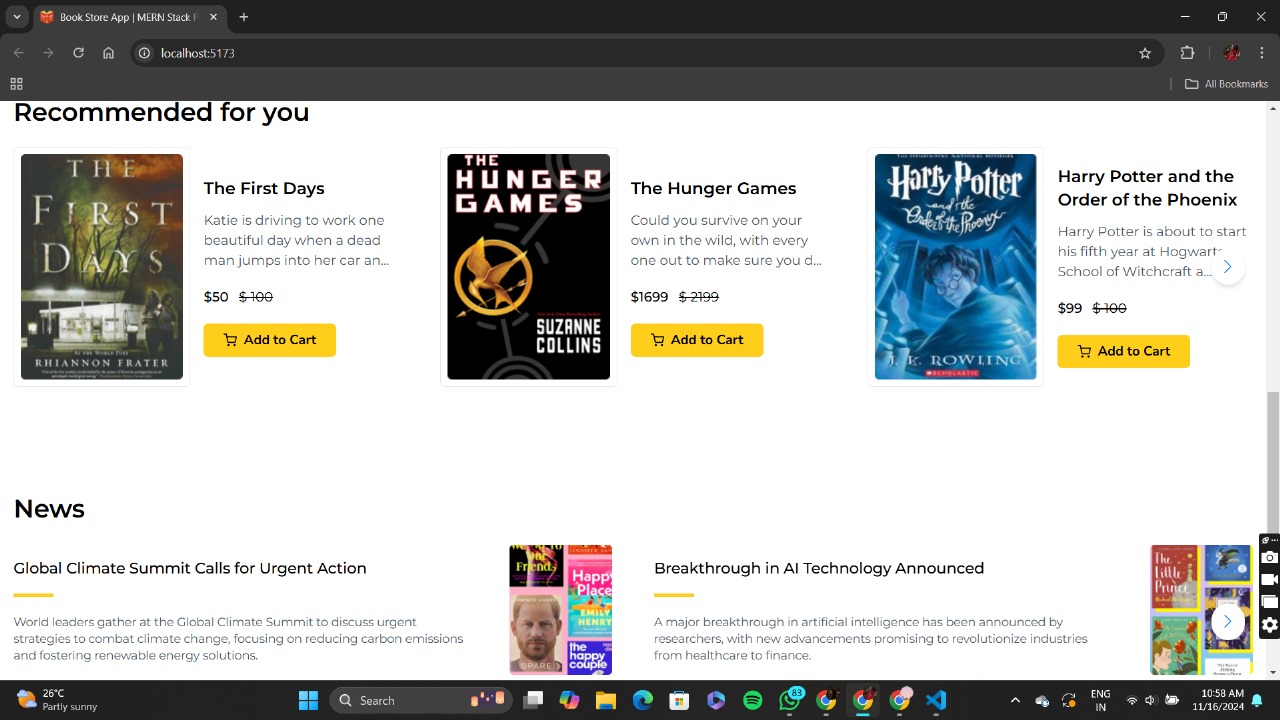
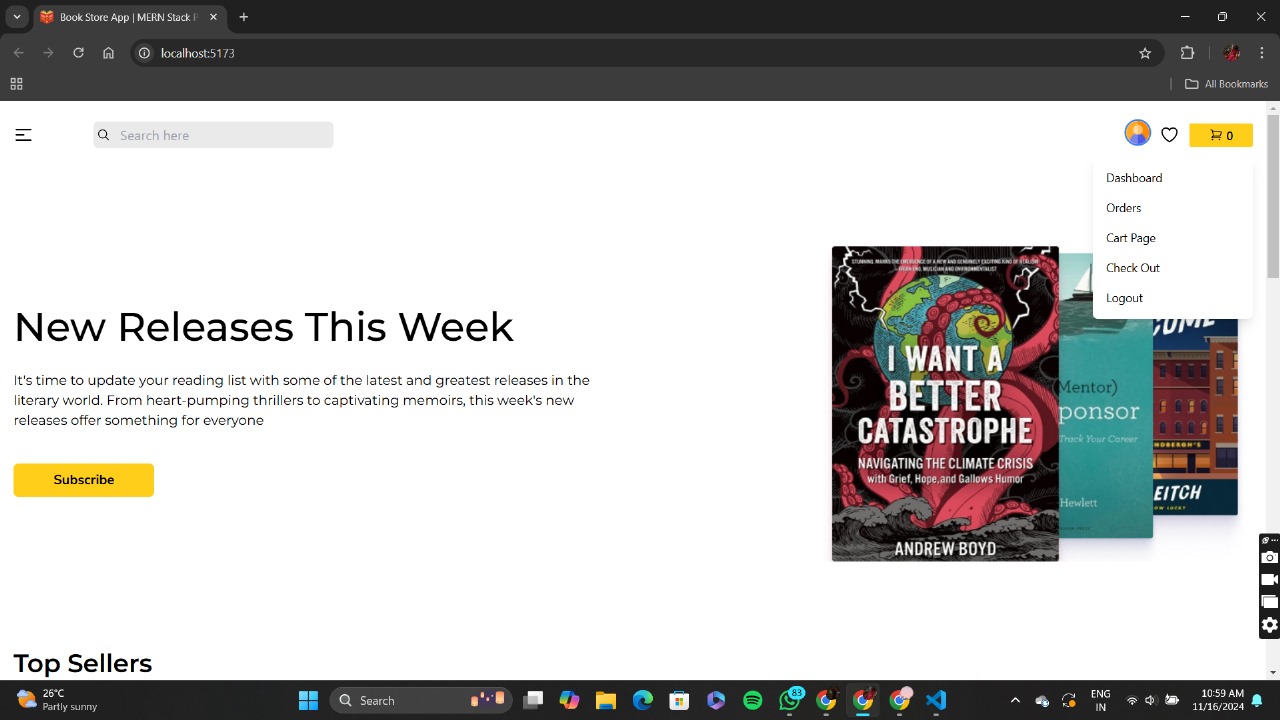
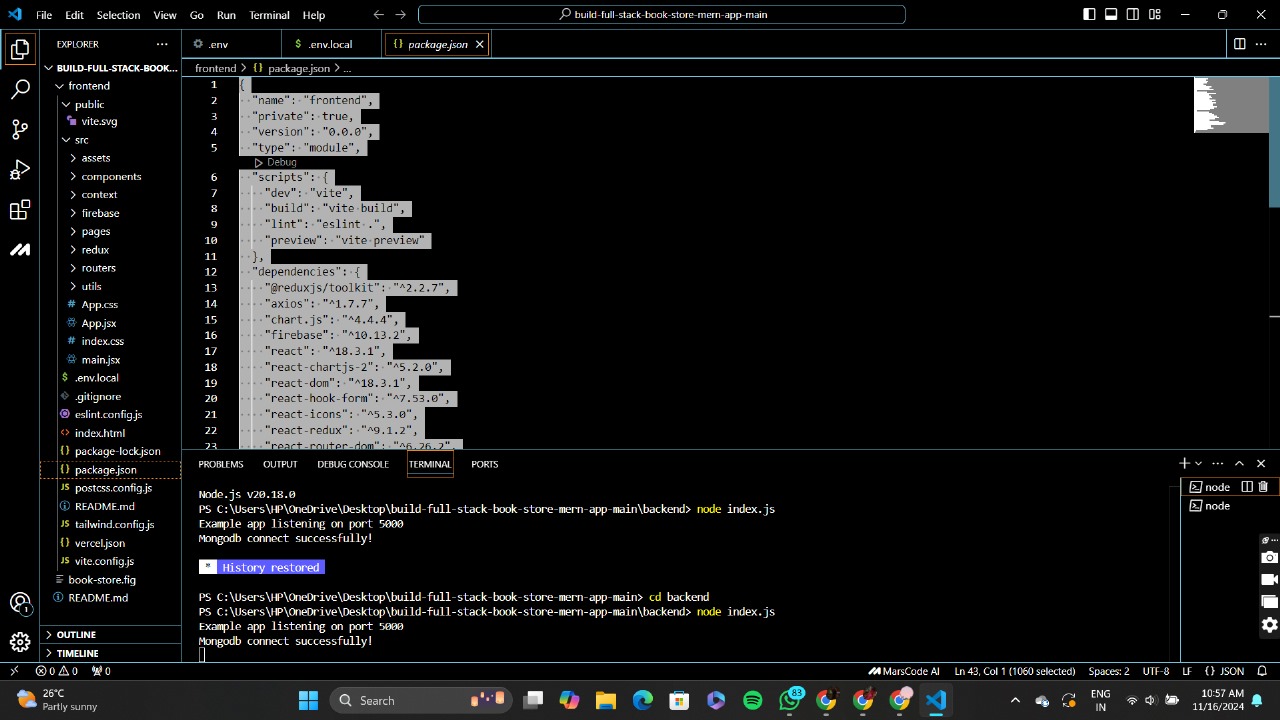
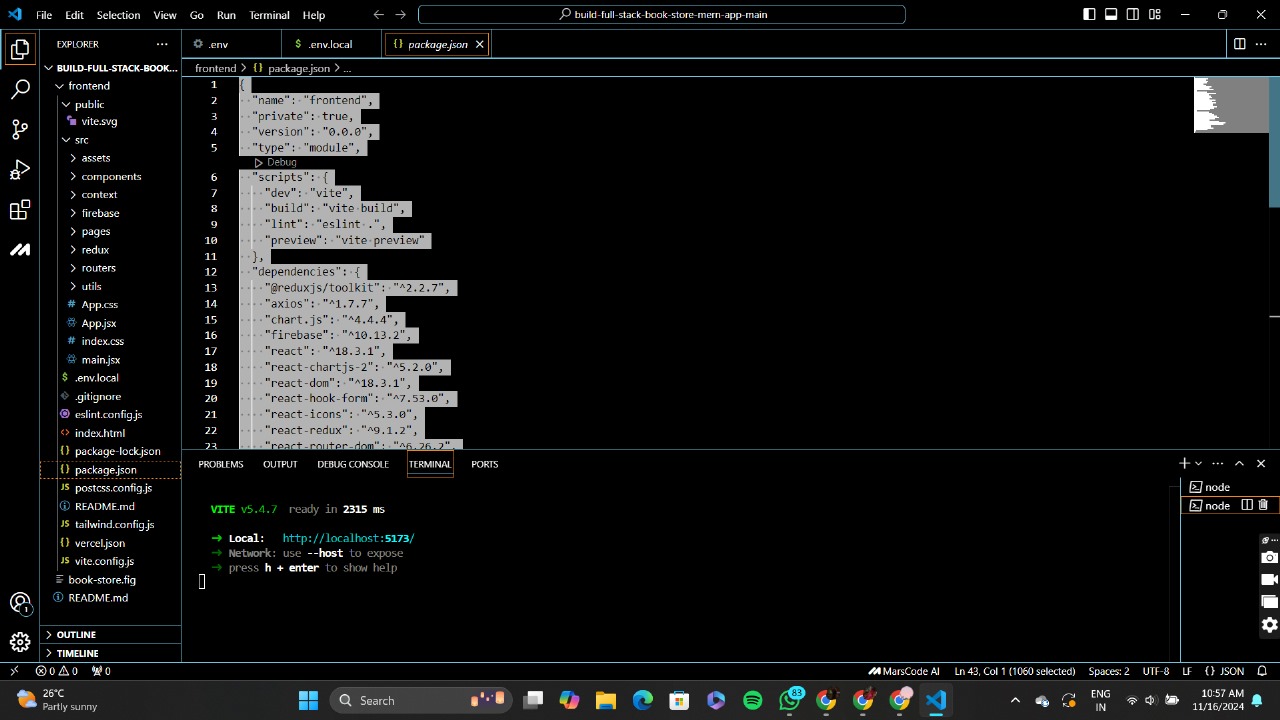
### 

### 10. Testing

* **Unit Testing:** Conducted using **Jest** for components and backend logic.
* **API Testing:** Performed using **Postman** to validate endpoints.

### 11. Screenshots or Demo

Screenshots of key UI components and admin panel.



Link to a live demo.



### <https://drive.google.com/file/d/1KwnOZqA_MH-isHI5kKAsYiUTOKjbMN7S/view?usp=sharing>

### 12. Known Issues

* **Slow Search Performance:** Optimization is required for large product datasets.
* **UI Bugs:** Minor alignment issues on smaller screens.

### 13. Future Enhancements

* **Voice Search:** Enable users to search using voice commands.
* **Mobile App:** Create a cross-platform app using React Native.
* **Multi-language Support:** Expand accessibility for global audiences.