

1. Introduction

- Project Title: *Book Store*
- Team Members:
 - Asgar George S A– Role :Frontend Developer
 - Pavan S– Role :Backend Developer
 - Mohammed Irfan P– Role :Database Manager
 - Sanjeev S– Role :Full Stack Developer

2. Project Overview

- Purpose: The *Book Store* project aims to create a digital bookstore that allows users to explore, search, and purchase books online. This platform provides an accessible space for book lovers and simplifies the book-purchasing process.
- Features:
 - User registration and login with authentication.
 - Book search and filter by genre, author, or title.
 - Detailed book view with description, price, and rating.
 - Shopping cart for managing book purchases.
 - Secure checkout process and payment integration.
 - Admin panel for adding, updating, and removing books.

3. Architecture

- Frontend: The frontend is built using React, ensuring a dynamic and responsive user experience. Component-based architecture allows for the modular organization of features such as navigation, book listings, search filters, and the shopping cart.
- Backend: Node.js and Express.js handle the backend, managing server-side logic and API requests. RESTful APIs support CRUD operations for books, users, and orders.
- Database: MongoDB is used as the database to store user profiles, book details, orders, and cart items. Schemas are designed to optimize performance and allow efficient querying.

4. Setup Instructions

- Prerequisites:
 1. *Node.js* (version 22.11.0)
 2. *MongoDB* (version 8.0.1)
- Installation:
 1. Clone the repository:

<https://github.com/Irfan13042004/Naan-Mudhalvan-Mongo-DB-Project>
 2. Navigate to the project directory.
 3. Install dependencies:
 - Frontend: `cd client && npm install`
 - Backend: `cd server && npm install`
 4. Set up environment variables:
 - MongoDB URI
 - Any necessary API keys for authentication and payments

5. Folder Structure

- Client:
 - *src*: Contains main React components, services, and assets.
 - *public*: Static files for the React application.
 - *package.json*: Lists dependencies and scripts for the frontend.
- Server:
 - *routes*: Defines API routes for books, users, and orders.
 - *models*: Schemas for MongoDB collections.
 - *controllers*: Functions for handling API requests.
 - *config*: Configuration files for environment variables.
 - *package.json*: Lists dependencies and scripts for the backend.

6. Running the Application

- To start the application locally:
 - Frontend: Run `npm start` in the *client* directory.

- Backend: Run `npm start` in the `server` directory.

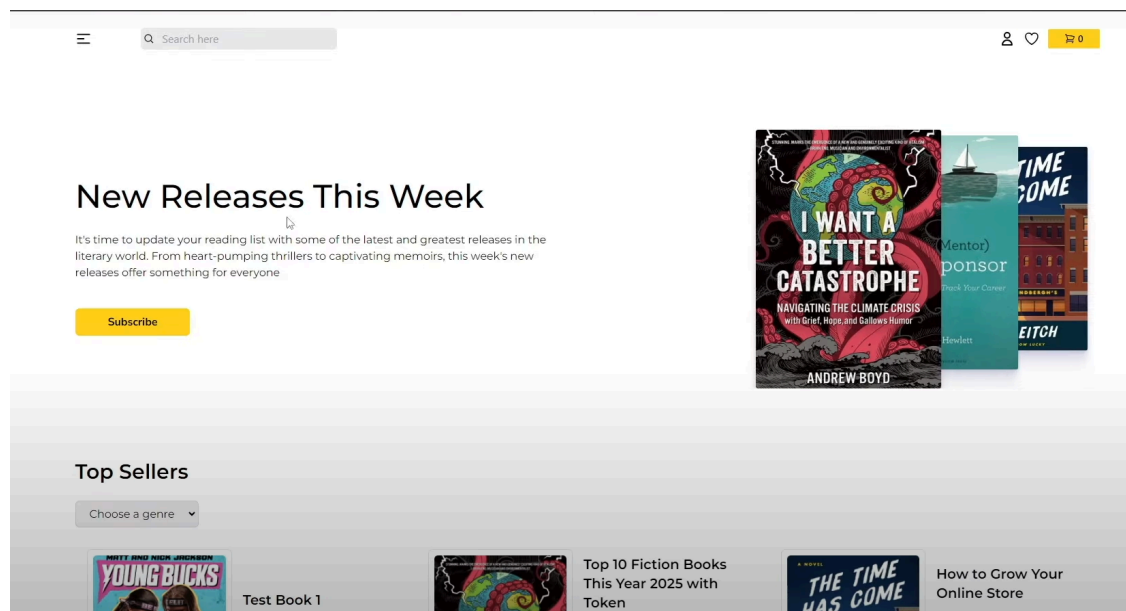
7. API Documentation

- Endpoints:
 - `/api/books` (GET): Fetch all books.
 - `/api/books/` (GET): Fetch book details by ID.
 - `/api/users` (POST): Register a new user.
 - `/api/cart` (POST): Add items to cart.
 - Each endpoint includes methods, required parameters, and response examples.

8. Authentication

- Authentication is managed using JWT tokens. Upon login, a token is issued and stored on the client side to maintain session state. Middleware is implemented to verify tokens for protected routes, ensuring secure access to user-specific and sensitive data.

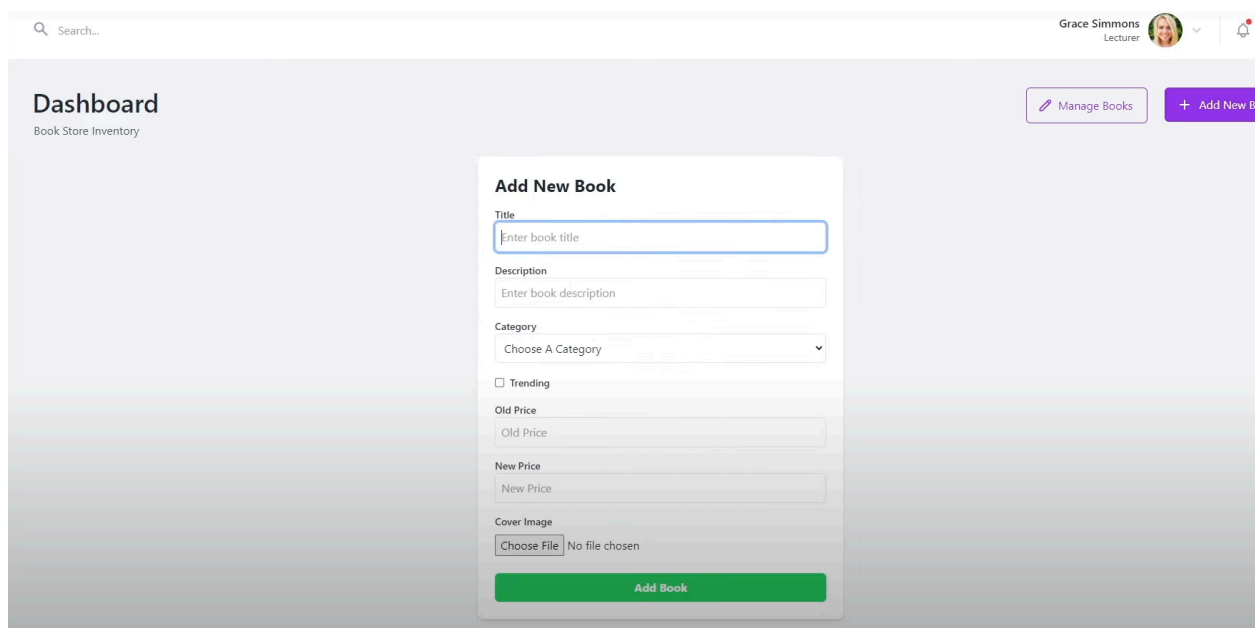
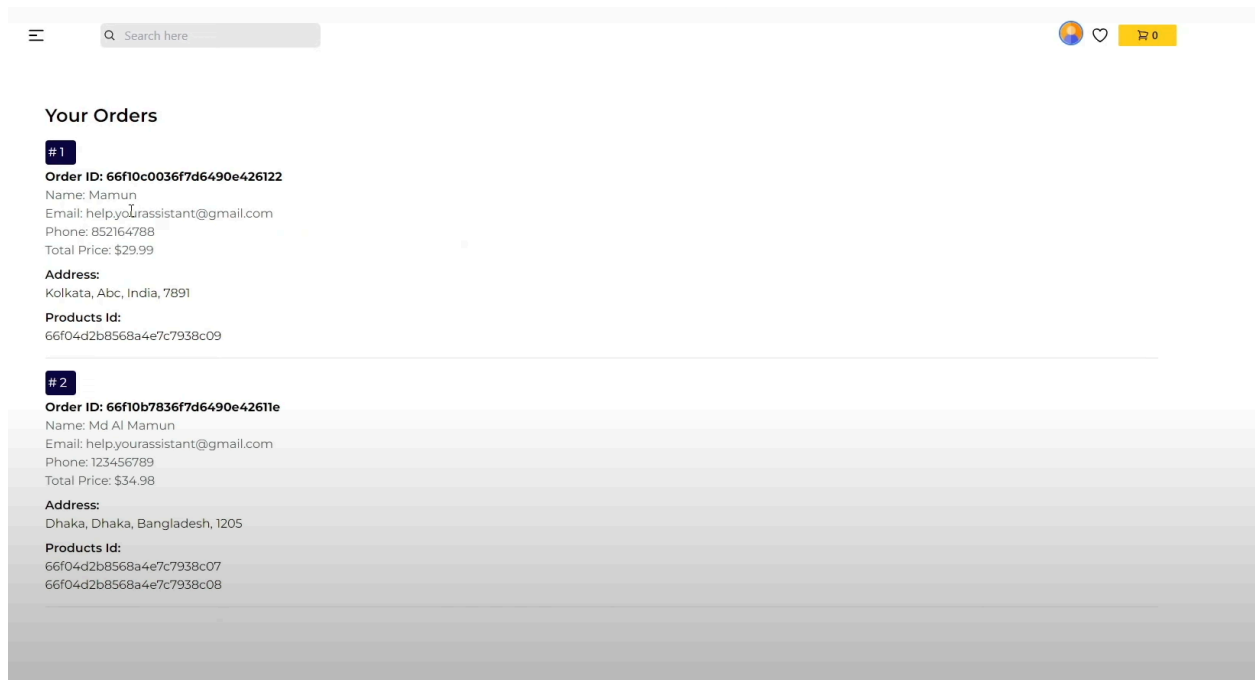
9. User Interface



10. Testing

- Testing strategies include unit testing with *Jest* for React components and API testing with *Mocha* and *Chai*. These tests ensure components and endpoints work as expected under various scenarios.

11. Screenshots or Demo



The screenshot shows a checkout interface for a 'Cash On Delevary' order. At the top, there's a search bar and a shopping cart icon with '1' item. The order summary indicates a 'Total Price: \$20.00' and 'Items: 1'. The main section is a 'Personal Details' form with the instruction 'Please fill out all the fields.' The form includes the following fields: 'Full Name' (filled with 'Moon'), 'Email Address' (filled with 'helpyourassistant@gmail.com'), 'Phone Number' (filled with '123456'), 'Address / Street' (filled with 'A'), 'City' (filled with 'B'), 'Country / region' (filled with 'C'), 'State / province' (filled with 'D'), and 'Zipcode' (filled with '123456'). Each field has a small 'x' icon for clearing the input. At the bottom of the form, there is a checkbox for 'I agree to the Terms & Conditions and Shopping Policy.' and a blue 'Place an Order' button.

12. Known Issues

- Document any bugs, limitations, or areas needing improvement, such as performance issues with large data sets or specific edge cases in the shopping cart functionality.

13. Future Enhancements

- Potential features to add include:
 - User reviews and ratings for books.
 - Personalized book recommendations.
 - Additional payment options.
 - Multi-language support for wider accessibility.