Full Stack Development with MERN

Project Documentation

1. Introduction

Project Title: BookNest: Where Stories Nestle

Team Members:

- **V. Shyam Kumar** – Backend Development & Database
- **S. Vamsi Krishna** – Frontend Development & Project Implementation

2. Project Overview

Purpose: To build a full-featured online book store for modern readers to browse, explore, and purchase books online.

Features:

- User registration and login
- Browse book listings
- Filter by genre, author, ratings
- Add to cart and secure purchase
- Order tracking and history

3. Architecture

Frontend: Built using React.js for a dynamic and responsive user interface.

Backend: Node.js and Express.js provide scalable APIs and handle server-side logic.

Database: MongoDB stores book information, user profiles, purchase history, and more.

4. Setup Instructions

Prerequisites:

- Node.js
- MongoDB

- Git

Installation:

1. Clone the repository
2. Navigate to the client and server folders
3. Run `npm install` in both directories
4. Set up environment variables (.env)

5. Folder Structure

Client: Contains the React.js frontend source files. **Server:** Contains the Node.js + Express backend API code and MongoDB connections.

6. Running the Application

- **Frontend:** `npm start` in the client directory
- **Backend:** `npm start` in the server directory

7. API Documentation

- Document all endpoints with methods, parameters, and sample responses
- Example: `GET /api/books` - Returns list of books

8. Authentication

- JWT-based token authentication
- Secure session handling for login, registration

9. User Interface

- Responsive UI built with React
- Components: Header, BookList, BookDetail, Cart, Orders, Login/Register

10. Testing

- Manual and automated testing using Postman and Jest
- Unit tests for API routes and frontend components

11. Screenshots or Demo

- Include screenshots or demo GIFs showcasing different UI states

12. Known Issues

- Some advanced filters may require optimization for large datasets

13. Future Enhancements

- Recommendation engine based on past purchases
- Admin dashboard with analytics
- Mobile app version

Scenario-Based Case Study

Scenario: Sarah, an avid reader with limited time, wants a convenient way to explore and purchase books online.

Solution:

- **User Registration and Authentication:** Secure account creation and login
- **Book Listings:** Explore books with details
- **Book Selection:** Filter by genre, author, etc.
- **Purchase Process:** Add to cart, order securely
- **Order Confirmation:** Confirmation page with full details
- **Order History:** View past purchases and track orders

Technical Architecture

- **User Interface:** Responsive book browsing and purchasing UI
- **Web Server:** Hosts the UI and connects to backend services
- **API Gateway:** Routes frontend requests to respective services
- **Authentication Service:** Manages JWT tokens and login flows
- **Database:** Stores books, users, orders, inventory
- **View Books & Categories:** Filter and browse books
- **Inventory Service:** Manages stock levels and book metadata
- **Order Management Service:** Handles cart, orders, and purchase tracking

Skills Used: HTML, CSS, JavaScript, Bootstrap, React.js, Node.js, MongoDB