

PROJECT REPORT

BookNest – Online Bookstore Web Application

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. IDEATION PHASE

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

4. PROJECT DESIGN

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

7. RESULTS

7.1 Output Screenshots

8. ADVANTAGES & DISADVANTAGES

9. CONCLUSION

10. FUTURE SCOPE

11. APPENDIX

GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

BookNest is a full-stack web-based Online Bookstore developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). The application is designed to digitize bookstore operations and provide customers with a seamless and efficient book browsing and purchasing experience.

Traditional bookstore operations often involve manual inventory management, paper-based billing, and lack of real-time availability tracking. These practices lead to inventory mismanagement, billing errors, delays, and reduced customer satisfaction. BookNest addresses these challenges by offering a centralized digital platform where customers can browse available books, add them to a cart, place orders online, and track order status in real time.

The system provides secure user authentication using JWT (JSON Web Token) and encrypted passwords to ensure data protection. Customers can create accounts, log in securely, and manage their order history. The application includes an Admin Dashboard enabling bookstore administrators to manage the catalog, update inventory, monitor orders, and control user activities efficiently.

1.2 Purpose

The primary purpose of BookNest is to provide a centralized and efficient digital platform for managing bookstore inventory and customer orders. The system aims to replace traditional manual methods with an automated, secure, and user-friendly online solution.

Key objectives of the system include:

- Providing an intuitive web platform for customers to browse books and place orders online.
- Allowing users to securely register and log in using JWT-based authentication.
- Enabling customers to add books to a cart and manage their orders efficiently.
- Providing administrators with tools to manage the book catalog and track customer orders.
- Reducing manual workload and minimizing billing or order-related errors.
- Building a scalable and cloud-based bookstore system using modern MERN stack technologies.

Overall, BookNest bridges the gap between customers and bookstore management by offering a reliable, secure, and structured online book purchasing solution that enhances convenience, efficiency, and business growth.

2. IDEATION PHASE

2.1 Problem Statement

Many bookstores still manage inventory and orders manually, which causes mistakes, delays, and confusion. Customers cannot easily check real-time availability, compare books by genre or author, or place orders online. Bookstore staff find it difficult to manage stock and process orders properly, leading to poor customer experience and inefficient operations.

Problem Statement (PS)	I am	I'm trying to	But	Because	Which makes me feel
PS-1	A Customer	Browse and purchase books quickly and conveniently	I cannot view available titles or track my order	The bookstore follows manual ordering methods	Frustrated and uncertain
PS-2	A Bookstore Admin	Manage daily orders and catalog updates efficiently	Orders are recorded manually and scattered	There is no centralized digital system	Overwhelmed and inefficient

2.2 Empathy Map

The following Empathy Map was created for Arjun – a book enthusiast and regular bookstore customer – to understand user pain points, motivations, and feelings that drive the need for BookNest.

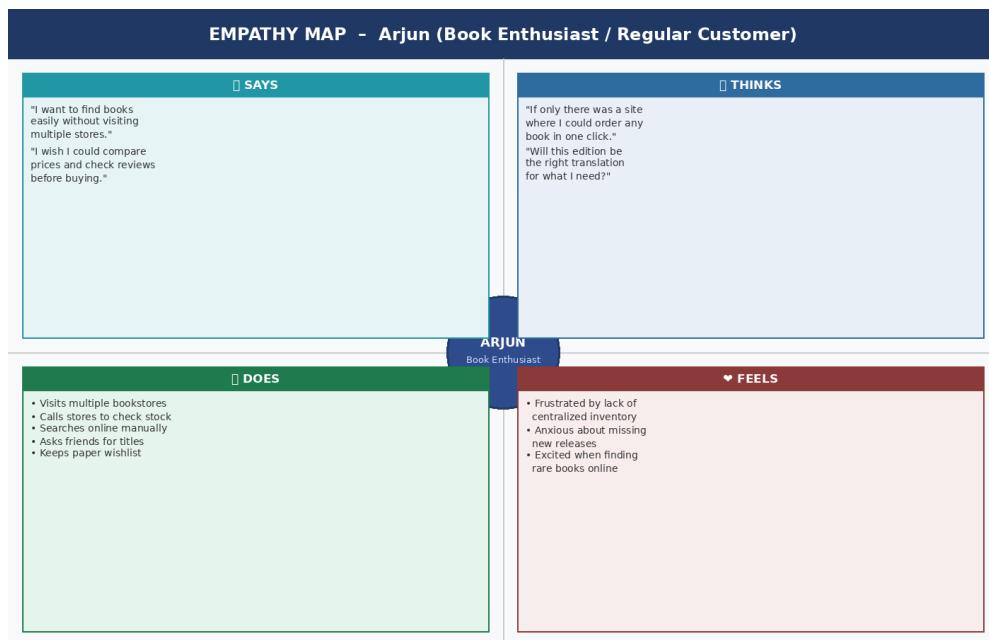


Figure-1: Empathy Map – Arjun (Book Enthusiast / Regular Customer)

2.3 Brainstorming & Idea Prioritization

The team conducted a structured brainstorming session to identify and prioritize features for BookNest. Ideas were grouped by functional area and then prioritized by impact, feasibility, and sprint assignment.

BRAINSTORM & IDEA GROUPING - BookNest Feature Planning				
User & Auth	Book Catalog	Cart & Orders	Admin Panel	Analytics & UX
User Registration	Book Listing	Add to Cart	Add / Edit Books	Order Tracking
JWT Login	Genre / Author Filter	Update Quantity	Delete Books	Order History
Role Selection (Customer / Admin)	Book Detail Page	Remove Item	Manage Orders	Notification System
Secure Password	Search Functionality	Place Order	Update Order Status	UI Responsiveness
Profile Management	Cover Image Upload	Order Confirmation	View Customers	Performance Optim.

IDEA PRIORITIZATION MATRIX								
USN-1 Registration High Sprint 1	USN-2 JWT Login High Sprint 1	USN-3 Book Catalog High Sprint 1	USN-4 Cart Mgmt High Sprint 2	USN-5 Order Placement Medium Sprint 2	USN-6 Order Tracking High Sprint 3	USN-7 Admin Panel High Sprint 3	USN-8 Notifications Medium Sprint 4	USN-9 UI Polish Medium Sprint 4

Figure-2: Idea Grouping & Prioritization Matrix

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The customer journey for BookNest begins when a user visits the platform and browses the book catalog. The user filters books by genre, author, or price, adds desired titles to the cart, and proceeds to place an order. Post-purchase, the customer receives an order confirmation and can track order status through the dashboard. The admin monitors all orders, updates inventory, and manages the overall platform.

3.2 Solution Requirements

Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Users can register using email and password Secure password creation Role selection (Customer / Admin)
FR-2	User Authentication	Login using Email & Password JWT-based authentication
FR-3	Book Listing	View available books Display book title, author, image, price, description Filter by genre / price range
FR-4	Cart Management	Add books to cart Update quantity / Remove books
FR-5	Order Placement	Place book order securely Store order details in database Generate order confirmation
FR-6	Order Management	View order history Track order status (Pending / Processing / Delivered)
FR-7	Admin Dashboard	Add / Edit / Delete books Manage orders View customer details
FR-8	Notification System	Display order confirmation messages Show order status updates

Non-Functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Simple and user-friendly interface allowing customers to browse and purchase books without technical knowledge.
NFR-2	Security	JWT authentication, encrypted passwords, and secure data handling to protect user data and privacy.
NFR-3	Reliability	System ensures accurate order processing and consistent status updates without data loss.
NFR-4	Performance	Fast loading of book catalog and real-time order actions through efficient API communication.
NFR-5	Availability	Platform accessible online anytime for customers and admins to manage book orders.
NFR-6	Scalability	MongoDB and REST architecture allow future expansion to more users, books, and services.

3.3 Data Flow Diagram

- Level 0 (Context Diagram): The customer interacts with the BookNest system by browsing books, placing orders, and viewing order history. The admin interacts with the system to manage inventory and orders.

- Level 1 (System Overview): Data flows from the user interface through the Express.js backend API, which processes authentication, catalog queries, cart operations, and order management. All data is stored and retrieved from MongoDB Atlas.
- Level 2 (Detailed Flow): Individual modules — Registration, Login, Book Listing, Cart Management, Order Placement, and Admin Management — each have defined input/output data flows connected through RESTful API endpoints.

3.4 Technology Stack

Technical Architecture:

BookNest follows a 3-Tier Client–Server Architecture: Presentation Layer (React.js), Application Layer (Node.js + Express.js), and Data Layer (MongoDB Atlas). The frontend communicates with backend REST APIs using Axios, while JWT authentication secures all protected routes.

S. No	Component	Description	Technology
1.	User Interface	Web interface for customers and admin dashboards	React.js, HTML, CSS, Bootstrap
2.	Application Logic-1	Authentication & Role Management	Node.js, Express.js, JWT
3.	Application Logic-2	Book Catalog & Order Processing	Express.js REST APIs
4.	Application Logic-3	Notification & Status Management	Node.js Controllers
5.	Database	Stores users, books, and order data	MongoDB, Mongoose
6.	Cloud Database	Cloud-hosted NoSQL database with remote access and scalability	MongoDB Atlas
7.	File Storage	Book cover image upload & storage	Local Filesystem / Cloud Storage
8.	External API-1	HTTP communication between frontend & backend	Axios
9.	Infrastructure	Local development deployment	Node.js Local Server

4. PROJECT DESIGN

4.1 Problem Solution Fit

Problem	Current State	BookNest Solution
No online catalog	Customers must visit the store to check availability	Real-time digital book catalog with search and filter
Manual order processing	Orders recorded on paper; prone to errors	Automated digital order management with database storage

Problem	Current State	BookNest Solution
No order tracking	Customers have no visibility into order status	Real-time order tracking (Pending / Processing / Delivered)
Poor admin control	Inventory managed in spreadsheets or notebooks	Admin dashboard for full catalog and order management

4.2 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement	Customers face difficulty browsing and purchasing books due to manual catalog systems, lack of real-time availability, and no online ordering. Bookstore admins need a centralized platform to manage inventory efficiently.
2.	Idea / Solution Description	BookNest is a MERN-stack web application that enables customers to browse books online through a centralized catalog. It provides role-based dashboards for Admin and Customer, real-time ordering, secure JWT authentication, and order tracking using REST APIs connected to MongoDB Atlas.
3.	Novelty / Uniqueness	Role-based bookstore workflow, admin-controlled inventory management, real-time order tracking, genre/author-based filtering, and a simplified user-friendly purchasing interface.
4.	Social Impact	Reduces customer effort, improves book accessibility, ensures updated inventory visibility, and enhances convenience through digital order management.
5.	Business Model	Commission-based order model, subscription plans for premium sellers, featured book listing, and future digital content services.
6.	Scalability	Built using 3-tier architecture (React frontend, Express backend, MongoDB Atlas), enabling scalable performance and easy expansion.

4.3 Solution Architecture

The following diagram illustrates the complete 3-Tier Architecture of BookNest, showing the data flow between the Presentation Layer (React.js), Application Layer (Node.js + Express.js REST APIs), and Data Layer (MongoDB Atlas).

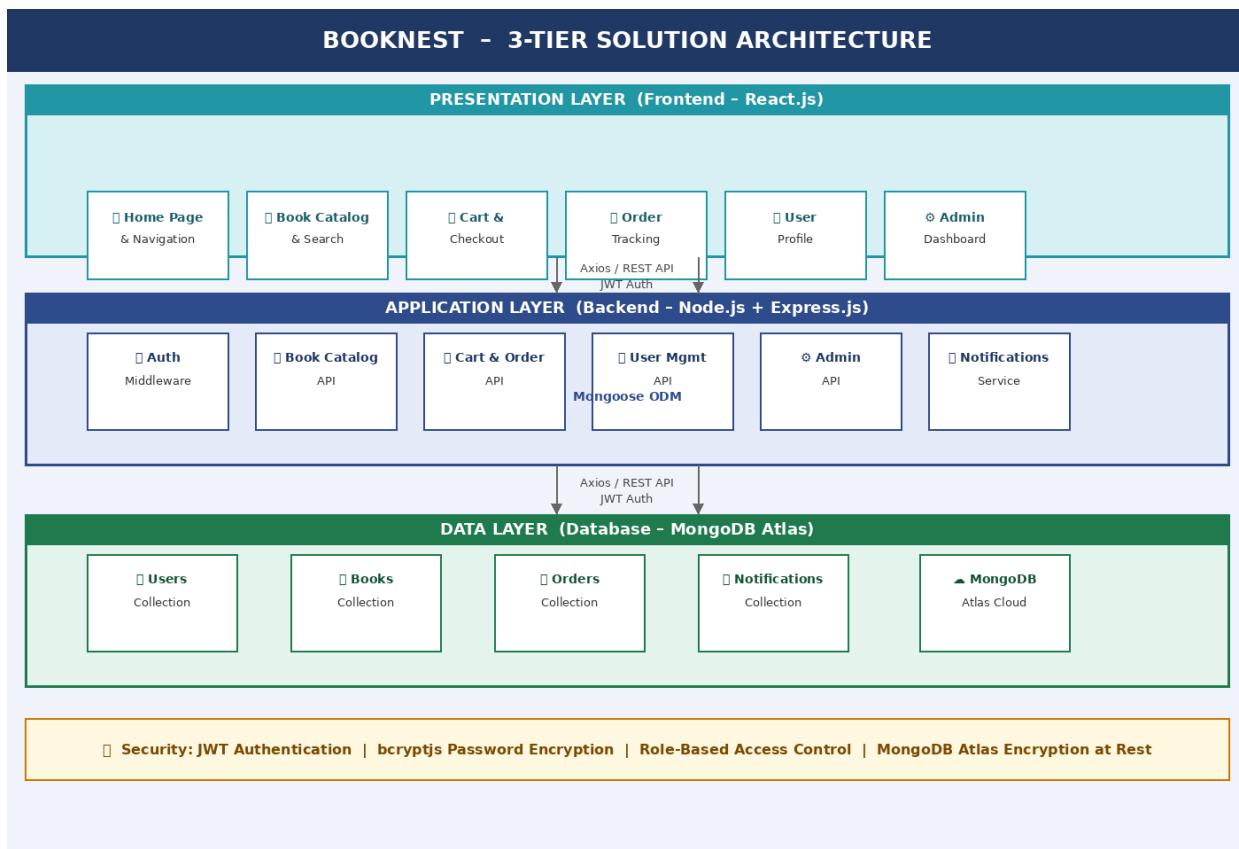


Figure-3: BookNest 3-Tier Solution Architecture Diagram

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story No.	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	User Registration	2	High	Team
Sprint-1	Authentication	USN-2	User login with JWT authentication	2	High	Team
Sprint-1	Book Listing	USN-3	Display available books from catalog	3	High	Team
Sprint-2	Cart Management	USN-4	Add books and update/remove items from cart	5	High	Team
Sprint-2	Order Placement	USN-5	Place order and store it in database	3	Medium	Team

Sprint	Functional Requirement (Epic)	User Story No.	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Order Tracking & History	USN-6	View order status and order history	5	High	Team
Sprint-3	Product/Order Management	USN-7	Admin can add, edit, delete books and update order status	4	High	Team
Sprint-4	Notifications	USN-8	Show order confirmation and status messages	3	Medium	Team
Sprint-4	UI Enhancement	USN-9	Improve UI responsiveness	3	Medium	Team

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	15	4 Days	01 Feb 2025	04 Feb 2025	15	04 Feb 2025
Sprint-2	16	4 Days	05 Feb 2025	08 Feb 2025	16	08 Feb 2025
Sprint-3	14	4 Days	09 Feb 2025	12 Feb 2025	14	12 Feb 2025
Sprint-4	12	3 Days	13 Feb 2025	15 Feb 2025	12	15 Feb 2025

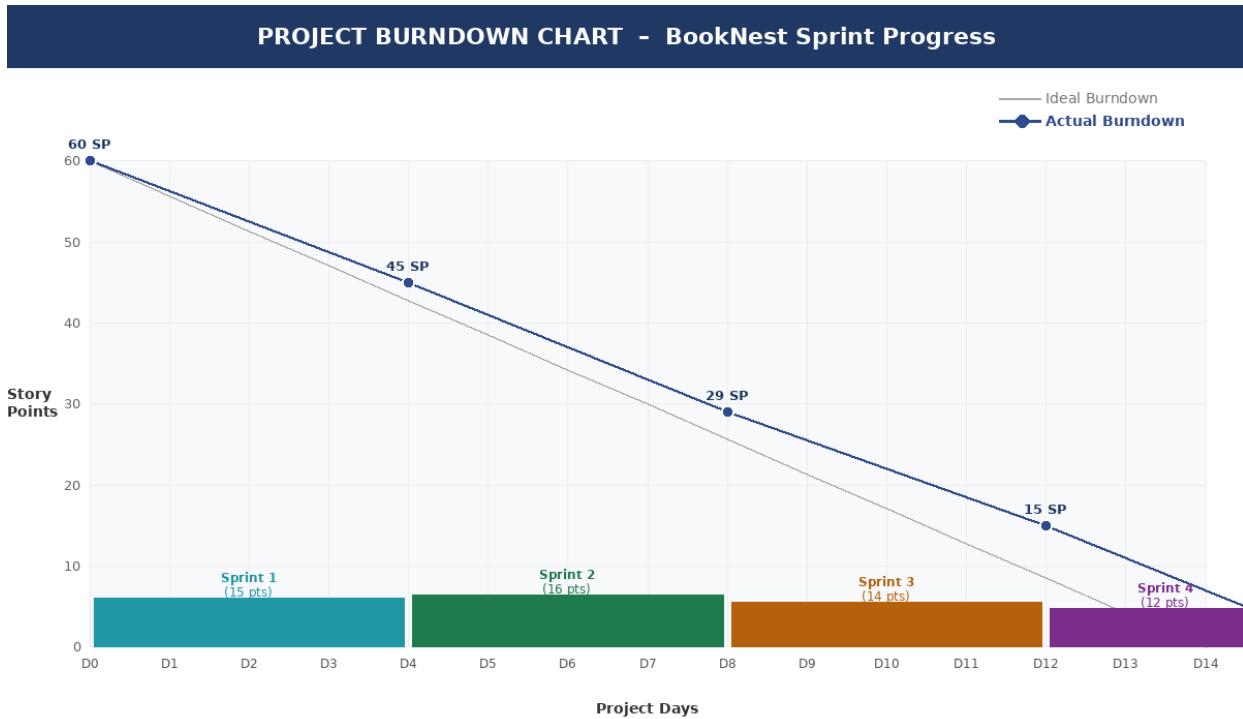


Figure-4: Sprint Burndown Chart – Story Points Remaining per Day

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Project Name: BookNest – Online Bookstore Web Application

BookNest is a MERN-stack web application that enables customers to browse books, add them to a cart, and place orders online with real-time order status tracking. The platform provides role-based dashboards for Customer and Admin, secure JWT authentication, book catalog management, and order tracking using MongoDB Atlas.

Project Version: v1.0 | Testing Period: 12 Feb 2025 to 15 Feb 2025

Features Tested:

- User Registration & Login
- Role-based Dashboard (Customer & Admin)
- Book Listing (Catalog Display with genre/author filter)
- Cart Management (Add / Update / Remove Books)
- Order Placement & Confirmation
- Order Status Tracking (Pending / Processing / Delivered)
- Admin Book Management (Add / Edit / Delete Books)
- Order Status Update by Admin
- Order Confirmation Notifications

Testing Environment:

URL/Location: <http://localhost:3000>

Credentials: User → reader1@gmail.com / ***** | Admin → admin@booknest.com / *****

Test Cases:

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-01	User Registration	1. Open website 2. Click Register 3. Enter valid details 4. Submit	Account is created successfully	Account created and redirected to login	Pass
TC-02	User Login	1. Enter valid email & password 2. Click Login	User is redirected to dashboard	Login successful and dashboard displayed	Pass
TC-03	Invalid Login	1. Enter wrong password 2. Click Login	Error message displayed	Error message shown correctly	Pass
TC-04	View Book Listing	1. Login 2. Navigate to catalog page	Books displayed with title, image, price	Books loaded from database correctly	Pass
TC-05	Add to Cart	1. Select book 2. Click Add to Cart	Item added to cart successfully	Book appears in cart	Pass
TC-06	Update Cart Quantity	1. Increase/decrease quantity in cart	Cart updates with correct total price	Quantity updated correctly	Pass
TC-07	Remove Item from Cart	1. Click Remove button in cart	Item removed from cart	Item removed successfully	Pass
TC-08	Place Order	1. Add books to cart 2. Click Place Order	Order saved in database and confirmation shown	Order stored and confirmation displayed	Pass
TC-09	Order Status Tracking	1. Navigate to Order History 2. Check order status	Order status displayed correctly	Status shown as Pending/Processing/Delivered	Pass
TC-10	Admin Add Book	1. Login as Admin 2. Add new book 3. Save	Book appears in catalog list	Book added successfully	Pass

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-11	Admin Edit Book	1. Modify book details 2. Save changes	Book updated in system	Book updated correctly	Pass
TC-12	Admin Delete Book	1. Click Delete on a book	Book removed from list	Book deleted successfully	Pass
TC-13	Admin Update Order Status	1. Change order status 2. Save	Updated status reflected for user	Status updated successfully	Pass
TC-14	Notification Display	1. Place order 2. Check confirmation message	Order confirmation displayed	Notification shown correctly	Pass

Bug Tracking:

Bug ID	Bug Description	Steps to Reproduce	Severity	Status	Additional Feedback
BUG-01	Cart total price not updating after changing book quantity	Login → Add book to cart → Increase quantity → Observe total price	High	Resolved	Fixed by recalculating total price on quantity update in frontend state management
BUG-02	Admin unable to update order status occasionally	Login as Admin → Navigate to Orders → Change status → Click Save	Medium	In Progress	Backend API needs validation check for order ID before updating status

Sign-off: Tester – Arjun Sharma | Date: 15 February 2025 | Signature: A. Sharma

7. RESULTS

7.1 Output Screenshots

The following composite screenshot panel shows the key pages of the BookNest application, including the Book Catalog, Cart & Checkout, Order Tracking, Admin Dashboard, Login Page, and Book Detail Page.

BOOKNEST - APPLICATION OUTPUT SCREENSHOTS

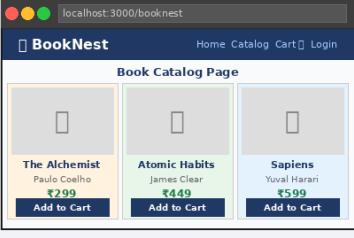


Fig-4: Admin Dashboard



Fig-5: Login Page

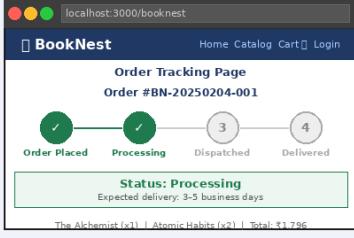


Fig-6: Order Tracking

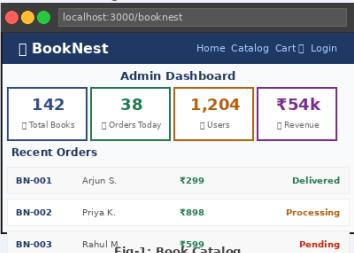


Fig-1: Book Catalog

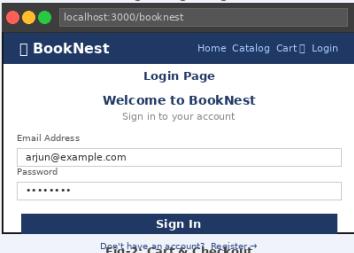


Fig-2: Cart & Checkout



Fig-3: Book Detail

TESTING SUMMARY

14 Total Test Cases	14 Passed	0 Failed	2 Bugs Found	1 Bugs Resolved	1 In Progress	100% Test Coverage	15 Feb 2025 Sign-off Date
------------------------	--------------	-------------	-----------------	--------------------	------------------	-----------------------	------------------------------

Figure-5: BookNest Application Output – Key UI Screens & Testing Summary

8. ADVANTAGES & DISADVANTAGES

Advantages

- Easy Online Book Purchasing: Customers can browse the catalog and place orders anytime without visiting the store.
- Time-Saving: Reduces waiting time compared to manual ordering systems.
- Secure Authentication: JWT-based login system ensures data security and protected user access.
- Centralized Management: Admin dashboard allows efficient management of books and orders.
- Reduced Manual Errors: Digital order storage minimizes billing mistakes and order confusion.
- Cloud Database Integration: MongoDB Atlas provides scalable and secure data storage.
- Order Tracking: Customers can view real-time order status updates.
- User-Friendly Interface: Responsive UI built with React.js improves customer experience.

Disadvantages

- Requires a stable internet connection to access the platform.
- Currently supports only web-based access (no dedicated mobile application).
- Payment gateway integration is not implemented in the current version.
- Real-time notifications may require further optimization for high traffic.
- System performance may depend on server load and hosting environment.

9. CONCLUSION

BookNest – Online Bookstore Web Application successfully addresses the challenges of traditional manual bookstore operations by providing a centralized digital platform. Using the MERN stack (MongoDB, Express.js, React.js, and Node.js), the system enables customers to browse the book catalog, manage carts, and place orders securely while allowing administrators to efficiently manage inventory and monitor orders.

The project demonstrates practical implementation of full-stack development concepts, including RESTful APIs, secure authentication using JWT, database integration with MongoDB Atlas, and responsive frontend design. By reducing manual errors and improving operational efficiency, BookNest enhances both customer convenience and bookstore management processes.

Overall, the project provides a scalable and reliable foundation for future enhancements such as payment gateway integration, mobile application development, and AI-based book recommendations.

10. FUTURE SCOPE

- Payment Gateway Integration: Integration with UPI, credit/debit cards, and net banking for secure online transactions.
- AI-Based Book Recommendation System: Personalized book suggestions based on user preferences and purchase history.
- Multiple Vendor/Seller Support: Expand the system to support multiple bookstores under one platform.
- Admin Analytics Dashboard: Advanced analytics to track sales, popular titles, customer trends, and revenue reports.
- Offers & Discount Management: Implement coupon codes, promotional offers, and loyalty reward systems.
- Mobile Application: Develop a companion mobile app for iOS and Android using React Native.
- Digital & Audiobook Support: Extend the platform to support digital downloads and audiobook subscriptions.

11. APPENDIX

GitHub Repository:

<https://github.com/EswararaoMalle/booknest.git>

Live Demo / Project Video:

https://drive.google.com/file/d/1dLsHwAoB6lN__D6xQmDTb6cM2wRFABb1/view?usp=sharing