

# **Project Report**

## **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 – Where Stories Nestle** is a full-stack web application developed using the MERN (MongoDB, Express.js, React, Node.js) stack. The application provides an online platform for users to browse, search, and purchase books across various genres and authors.

Traditional bookstores require physical presence and offer limited browsing convenience. BookNest overcomes these limitations by providing a centralized digital bookstore where users can explore books, manage carts, place orders, and track purchase history. Sellers can manage inventory and orders, while administrators monitor users, sellers, and overall system activity.

The application follows a client-server architecture with React.js handling the frontend, Node.js and Express.js managing backend services, and MongoDB ensuring efficient data storage and retrieval.

## **1.2 Purpose:**

The purpose of the BookNest application is to simplify and enhance the book purchasing experience by providing a user-friendly and secure online bookstore.

### **Key objectives of the system include:**

- Enable users to browse and purchase books online
- Provide sellers with tools to manage inventory and orders
- Allow administrators to control users, sellers, and content
- Ensure secure authentication and smooth transactions
- Offer a scalable digital solution for book commerce.

Overall, the project aims to bridge the gap between readers and online book services by offering a reliable and structured digital bookstore platform that improves user experience, simplifies book purchasing, and enhances operational efficiency.

# **2. IDEATION PHASE**

## **2.1 Problem Statement:**

In the modern digital era, book lovers often face difficulties visiting physical bookstores due to time constraints, limited availability, and lack of variety. Existing online platforms may lack intuitive design or seller transparency.

## Problem Statement -1:

### Define Problem Statements

	I am:	I'm trying to:	But:	Which makes me feel:
Customer	A customer	Purchase books online easily	I cannot quickly find books with complete details.	Confused and frustrated
Seller	A seller	Manage my book inventory digitally	There is no centralized platform	Overwhelmed
Administrator	An administrator	Monitor users and transactions	There is no unified system	Concerned about system control

## Problem Statement -2:

### Problem Statement

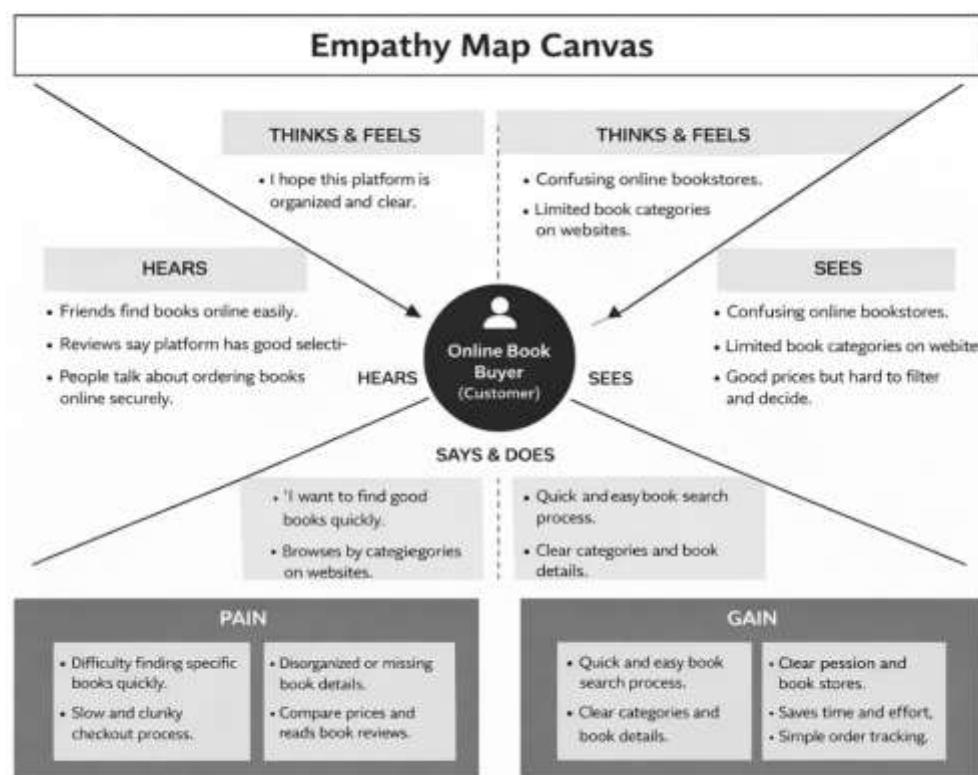
Who is having the problem?	What's the problem?	Why is this a problem?	How are they accomplishing this today?	What would my solution do?
Customer	Cannot browse and purchase books online easily	Limited access to organized and detailed browsing	Visiting physical bookstores with limited hours, manual browsing	Enable purchase of books online anytime
Seller	Managing inventory manually is time-consuming	Causes inventory errors and inefficiencies	Tracking stock manually or using un-integrated tools	Centralized inventory management platform
Administrator	Lacks unified control over platform users and activities	Difficulty monitoring and managing the platform holistically	Managing users, sellers and books manually without integration	Provide centralized platform management tools

Problem Statement (PS)	I am	I'm trying to	But	Because	Which makes me feel
PS-1	A customer	To browse and purchase books online easily	I cannot quickly find books with complete details	There is limited access to organized and detailed	Confused and frustrated

				online browsing	
PS-2	A seller	To manage my book inventory digitally	There is no centralized platform	Inventory is handled manually or using unintegrated tools	Overwhelmed

## 2.2 Empathy Map:

User: - Online Book Buyer (Customer purchasing books digitally)



## 2.2 Brainstorm & Idea Prioritization:-

Step-1: Team Gathering, Collaboration and Select the Problem Statement



## Step-2: Brainstorm, Idea Listing and Grouping

**2**

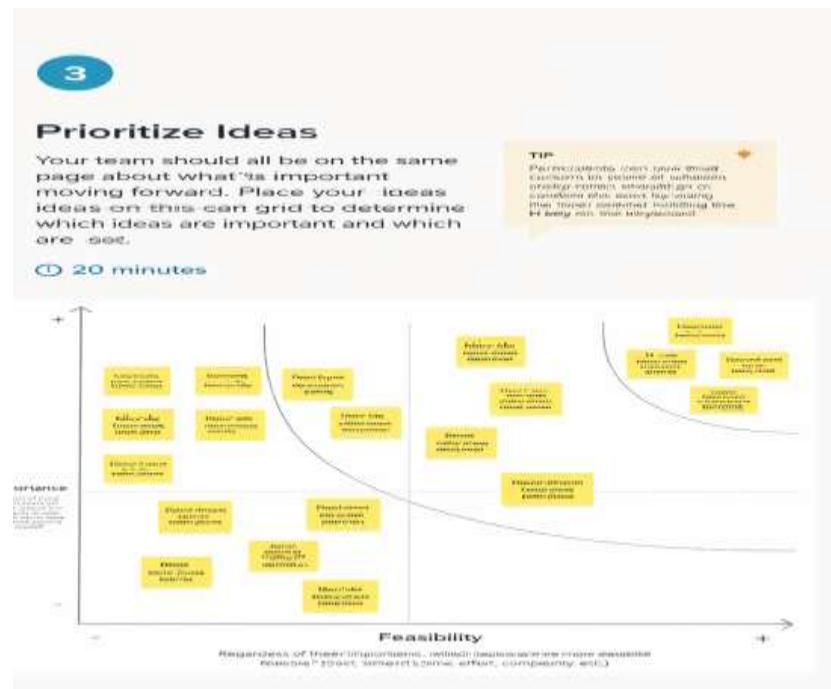
**Brainstorm & Group Ideas**

Write down any ideas that come to mind that address your problem statement.

⌚ 10 minutes

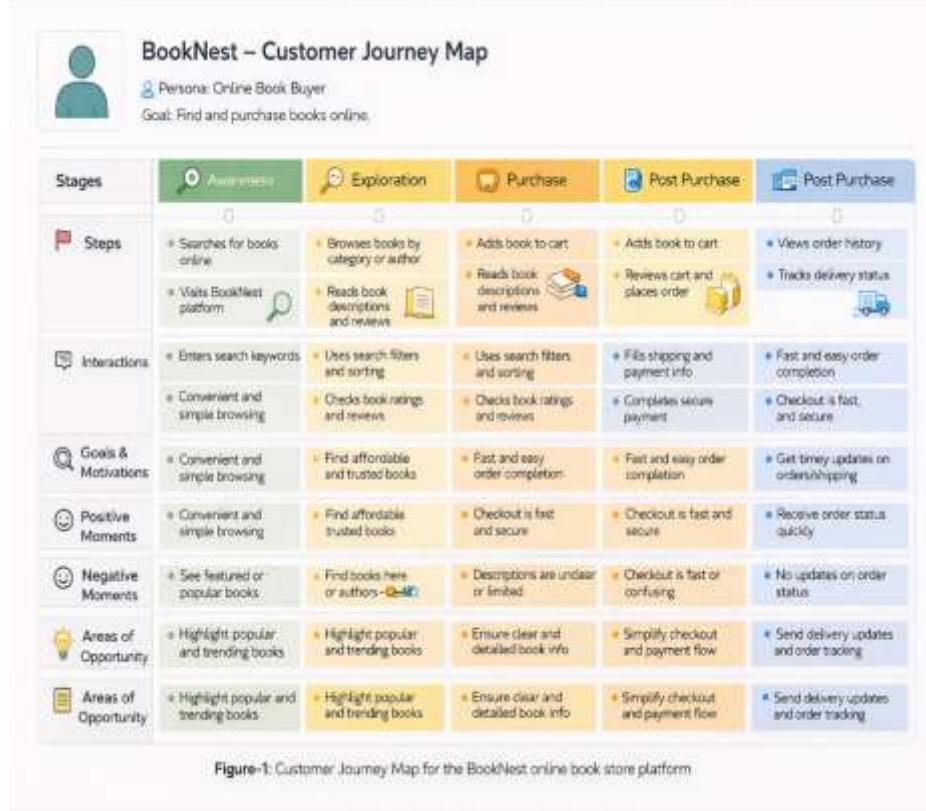
Narendra Mangam				
Customer Experience Ideas	Seller Experience Ideas		Admin & Governance	
User authentication (UAT)	Seller registration & approval	Seller reputation approval	Administic book category	Agent user onboarding
User registration & approval	Merchant onboarding & approval	Merchant category	Administic book category	Technical feature ideas
Technical feature ideas	Review fetch ideas	Merchant book id & creation		

## Step-3: Idea Prioritization



## REQUIREMENT ANALYSIS

### 3.1 Customer Journey map:



## 3.2 Solution Requirement

### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Role selection (User / Doctor) Secure password creation
FR-2	User Authentication	Login using Email & Password JWT-based authentication
FR-3	Book Browsing & Search	View book list on dashboard Filter by genre / author / price
FR-4	Book Purchase	Add books to cart Select quantity
FR-5	Order Management	View order history Track order status
FR-6	Seller Dashboard	Add new books Update inventory
FR-7	Admin Governance	Manage users and sellers Approve sellers Remove inactive books
FR-8	Review & Wishlist System	Add books to wishlist Submit ratings & reviews

### Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description

NFR-1	<b>Usability</b>	Simple and user-friendly interface allowing patients to book appointments easily without technical knowledge.
NFR-2	<b>Security</b>	JWT authentication, encrypted passwords, and secure transaction handling to protect user accounts, order data, and privacy.
NFR-3	<b>Reliability</b>	System ensures accurate book listings, inventory updates, and order processing with consistent status tracking and no data loss.
NFR-4	<b>Performance</b>	Fast loading of dashboards and smooth browsing and purchasing actions through efficient API communication.
NFR-5	<b>Availability</b>	Platform accessible online anytime for customers, sellers, and administrators to manage books, orders, and user activities
NFR-6	<b>Scalability</b>	MongoDB and REST architecture allow future expansion to support more users, sellers, books, and online services.

### 3.3 Data Flow Diagram:

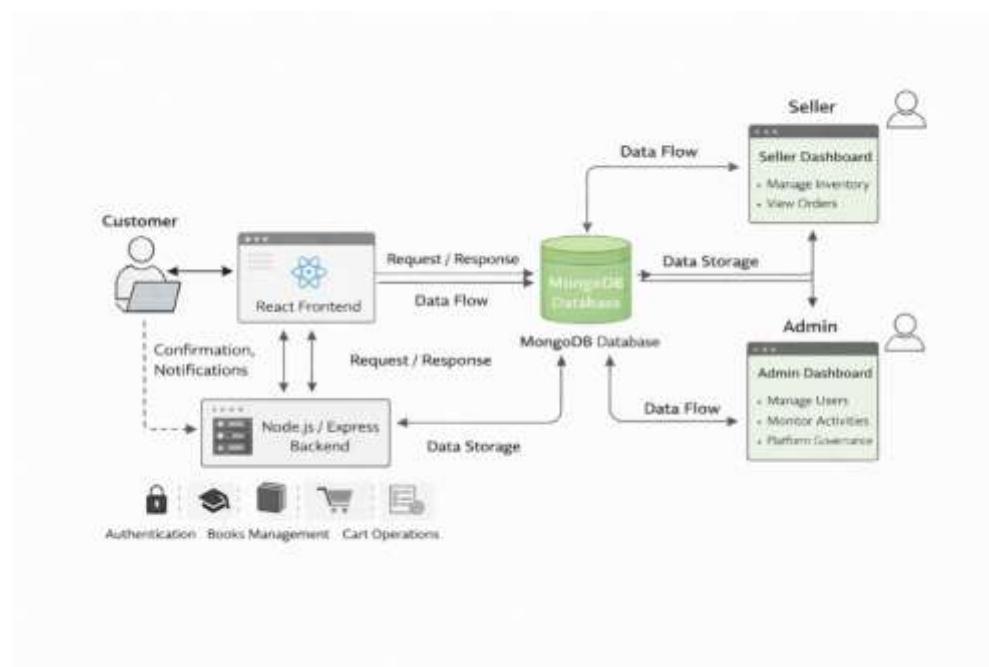


Fig:Data flow diagram of BookNest

## User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web User)	Registration	USN-1	As a customer, I can register using email and password.	Account created and dashboard accessible.	High	Sprint-1
Customer (Web User)	Authentication	USN-2	As a user, I can log in securely	Successful login redirects to dashboard	High	Sprint-1
Customer (Web User)	Book Listing	USN-3	As a customer, I can view available books.	Book list loads from database.	High	Sprint-1
Customer (Web User)	Cart & Order	USN-4	As a customer, I can add books to cart and place orders.	Order created successfully.	High	Sprint-2
Seller	Inventory Management	USN-5	As a seller, I can add and update book listings.	Books updated and visible to customers.	Medium	Sprint-2
Seller	Order Management	USN-6	As a seller, I can view customer orders.	Orders displayed in seller dashboard.	High	Sprint-3
Administrator	Platform Governance	USN-7	As an admin, I can manage users, sellers, and books.	Admin actions reflected in system.	High	Sprint-3
Customer Care Executive	Notifications	USN-8	As a customer, I receive order confirmation updates.	Notification displayed after purchase.	Medium	Sprint-4
Customer (Web User)	Order History	USN-9	As a customer, I can view my previous orders.	Past orders visible in dashboard.	Medium	Sprint-4

## 3.4 Technology Stack: -

### Technical Architecture: -

The BookNest application follows a 3-Tier Client–Server Architecture:

- **Presentation Layer (Frontend):** React.js web interface where users, sellers, and administrators interact with the system for browsing books, managing carts, and handling orders.
- **Application Layer (Backend):** Node.js and Express.js handle API requests, authentication, book management, order processing, and role-based access control.

- **Data Layer (Database):** MongoDB stores user profiles, seller information, book details, orders, and reviews.

The frontend communicates with backend REST APIs using Axios, while JWT authentication secures protected routes and user sessions.

**Table-1: Components & Technologies:**

S . N o	Component	Description	Technology
	User Interface	Web interface for customers, sellers, and admin dashboards	React.js, HTML, CSS, Bootstrap, Material UI
	Application Logic-1	Authentication & Role Management	Node.js, Express.js, JWT
	Application Logic-2	Book browsing, cart management, and order processing	Express.js REST APIs
	Application Logic-3	Inventory management and order status handling	Node.js Controllers
	Database	Stores users, sellers, books, and orders data	MongoDB, Mongoose
	Cloud Database	Cloud-hosted NoSQL database for storing application data with scalability	MongoDB Atlas
	File Storage	Book images and related media storage	IBM Block Storage or Other Storage Service or Local Filesystem
	External API-1	HTTP communication between frontend & backend	Axios
	External API-2	Not Applicable	—
	Machine Learning Model	Not used in this project	—

	Infrastructure (Server / Cloud)	Local development deployment	Node.js Local Server
--	------------------------------------	------------------------------	----------------------

**Table-2: Application Characteristics:**

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks used to build UI & Backend APIs	React.js, Express.js, Node.js
2.	Security Implementations	Role-based authentication, encrypted passwords, protected routes	JWT, bcryptjs, Middleware
3.	Scalable Architecture	3-Tier architecture separating UI, backend logic, and database	REST Architecture, MongoDB
4.	Availability	Web application accessible anytime through browser	Node.js Server
5.	Performance	Fast API responses and asynchronous communication	Axios, Express.js

## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit:



#### **4.2 Proposed Solution:**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Customers face difficulty finding and purchasing books efficiently through traditional bookstores or unorganized online platforms. Sellers struggle with manual inventory management, while administrators lack centralized control over users and transactions. These challenges lead to poor user experience, wasted time, and inefficient business operations.
2.	Idea / Solution description	BookNest is a web-based online book store application that enables customers to browse, search, and purchase books digitally through a centralized system. The platform provides role-based dashboards for Admin, Seller, and Customer. Customers can add books to cart, place orders, and track purchases. Sellers manage book listings and inventory, while administrators oversee users and platform activities using REST APIs connected to MongoDB.
3.	Novelty / Uniqueness	Role-based bookstore workflow, centralized inventory management, user-friendly browsing interface, secure authentication, real-time order tracking, and integrated admin governance in a single digital platform.
4.	Social Impact / Customer Satisfaction	Improves accessibility to books, reduces physical store dependency, saves time for customers, supports small sellers digitally, and enhances overall user satisfaction through convenient online purchasing.
5.	Business Model (Revenue Model)	Commission-based sales model, premium seller listings, featured book promotions, and future subscription plans for sellers..
6.	Scalability of the Solution	Built using 3-tier architecture (Frontend, Backend, MongoDB database), allowing scalable performance and easy expansion to

		support more users, sellers, and book categories..
--	--	--

### 4.3 Solution Architecture:

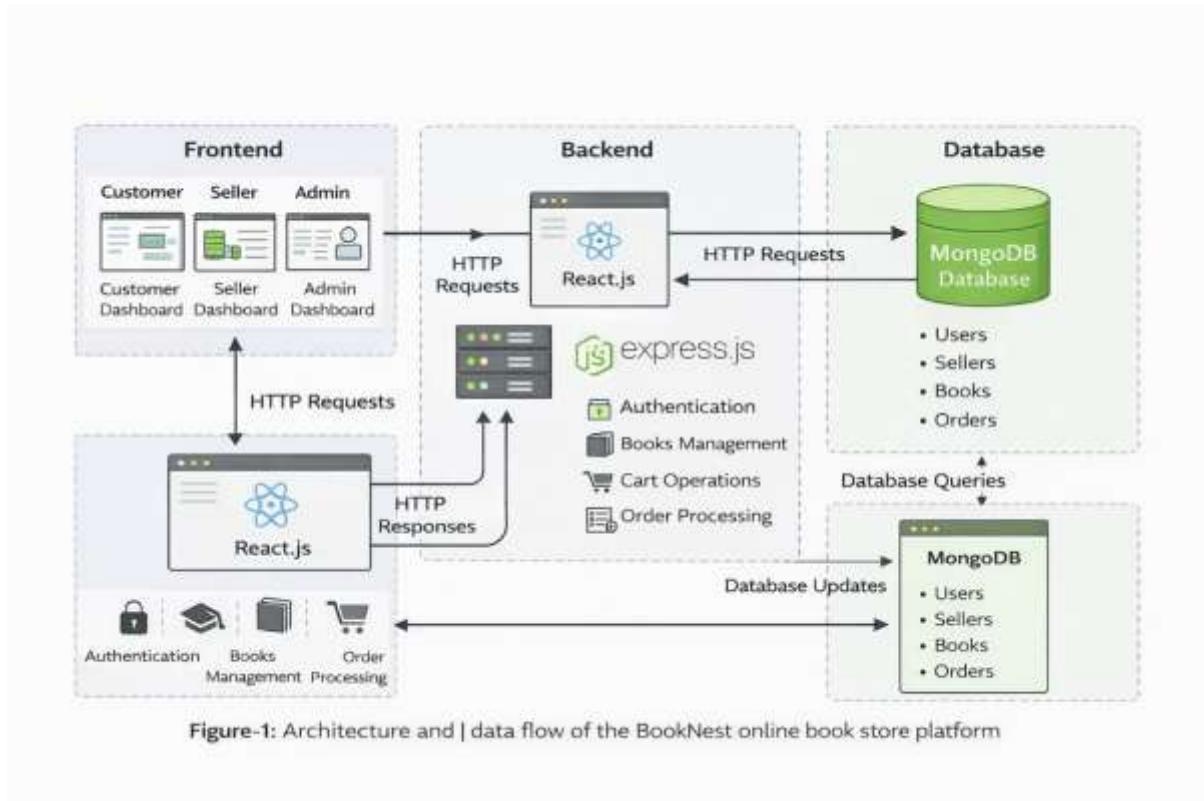


Figure-1: Architecture and data flow of the BookNest online book store platform

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning:

#### Product Backlog, Sprint Schedule, and Estimation :

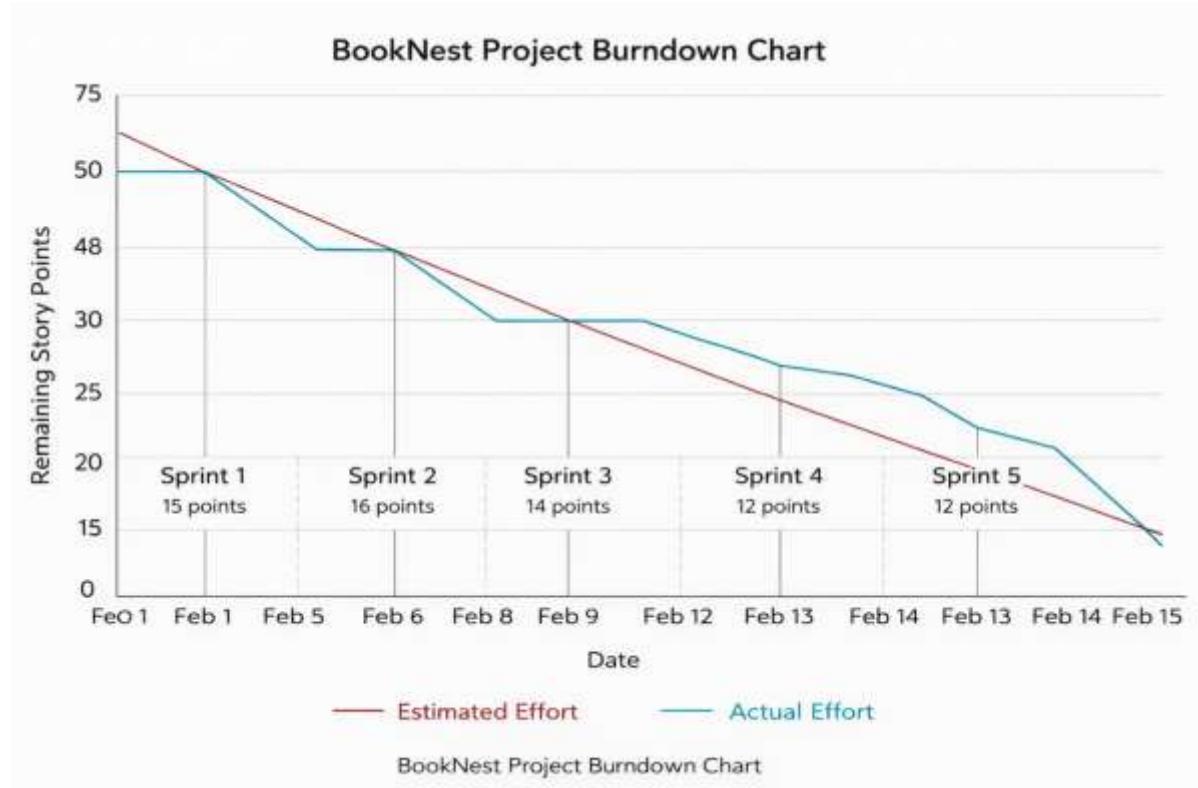
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a customer or seller, I can register using	2	High	Team

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			email and password.			
Sprint-1	Authentication	USN-2	As a user, I can log in securely using JWT authentication	2	High	Team
Sprint-1	Dashboard	USN-3	As a customer, I can view available books on the dashboard.	3	High	Team
Sprint-2	Cart & Order Placement	USN-4	As a customer, I can add books to cart and place an order.	5	High	Team
Sprint-2	Book Search & Filter	USN-5	As a customer, I can search books by title or category.	3	Medium	Team
Sprint-3	Seller Panel	USN-6	As a seller, I can add, update, and manage book inventory.	5	High	Team
Sprint-3	Admin Panel	USN-7	As an admin, I can manage users, sellers, and platform activities.	4	High	Team
Sprint-4	Notifications	USN-8	As a customer, I receive order confirmation and status updates.	3	Medium	Team
Sprint-4	Order History	USN-9	As a customer, I can view and manage order history.	3	Medium	Team

#### Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	4 Days	01 Feb 2025	04 Feb 2025	15	04 Feb 2025

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
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



## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 Performance Testing:

#### Project Overview:

Project Name: BookNest: Where Stories Nestle

BookNest is a MERN-stack web application that enables customers to browse and purchase books online. The platform provides role-based dashboards for Admin, Seller, and Customer, secure JWT authentication, inventory management, cart and order processing, and order status tracking using MongoDB Atlas.

Project Version: v1.0

Testing Period: 15 Feb 2025 to 19 Feb 2025

### Testing Scope:

#### Features Tested:

- User Registration & Login
- Role-based Dashboard
- Book Browsing
- Cart & Order Placement
- Seller Inventory Management

Order Status Notifications

#### User Stories Tested:

- USN-1 Registration
- USN-2 Login
- USN-3 Book Listing
- USN-4 Cart & Order
- USN-6 Seller Inventory
- USN-8 Notifications

#### Testing Environment:

URL/Location: <http://localhost:5173>

Credentials (if required): User → testuser@gmail.com / \*\*\*\*\*

Doctor → doctest@gmail.com / \*\*\*\*\*

Admin → admin@gmail.com / \*\*\*\*\*

#### Test Cases:

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	User Registration	Step1: Open Register Page Step2: Enter details Step3: Submit	User account created successfully	Account created	Pass

TC-002	User Login	Step1: Enter email & password Step2: Click Login	Redirect to dashboard	Dashboard loaded	Pass
TC-003	View Books	Step1: Login Step2: Open Dashboard	Book list displayed	Books visible	Pass
TC-004	Add to Cart & Order	Step1: Select book Step2: Add to cart Step3: Place order	Order created	Order placed	Pass
TC-005	Seller Adds Book	Step1: Seller login Step2: Add new book	Book added successfully	Book added	Pass
TC-006	Admin Approval	Step1: Admin manages seller	Seller status updated	Approved successfully	Pass
TC-007	Order Status Update	Step1: Seller processes order	Status changes to Completed	Status updated	Pass

### Bug Tracking:

Bug ID	Bug Description	Steps to reproduce	Severity	Status	Additional feedback
BG-001	Minor UI alignment issue in dashboard	Step-1: Login Step-2: Open Dashboard	Medium	Closed	Fixed using CSS update
BG-002	Slow image loading	Load book images	Medium	Closed	Optimization suggested

## 7. RESULTS

### 7.1 Output Screenshots:

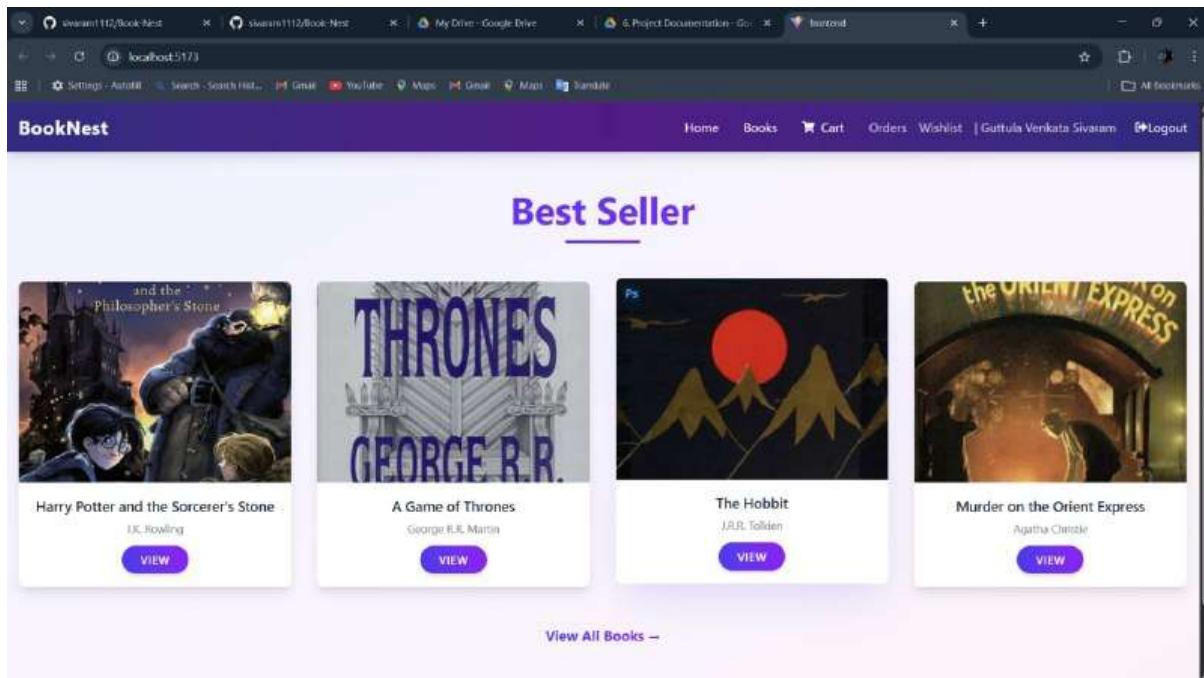


Figure-2: User Dashboard

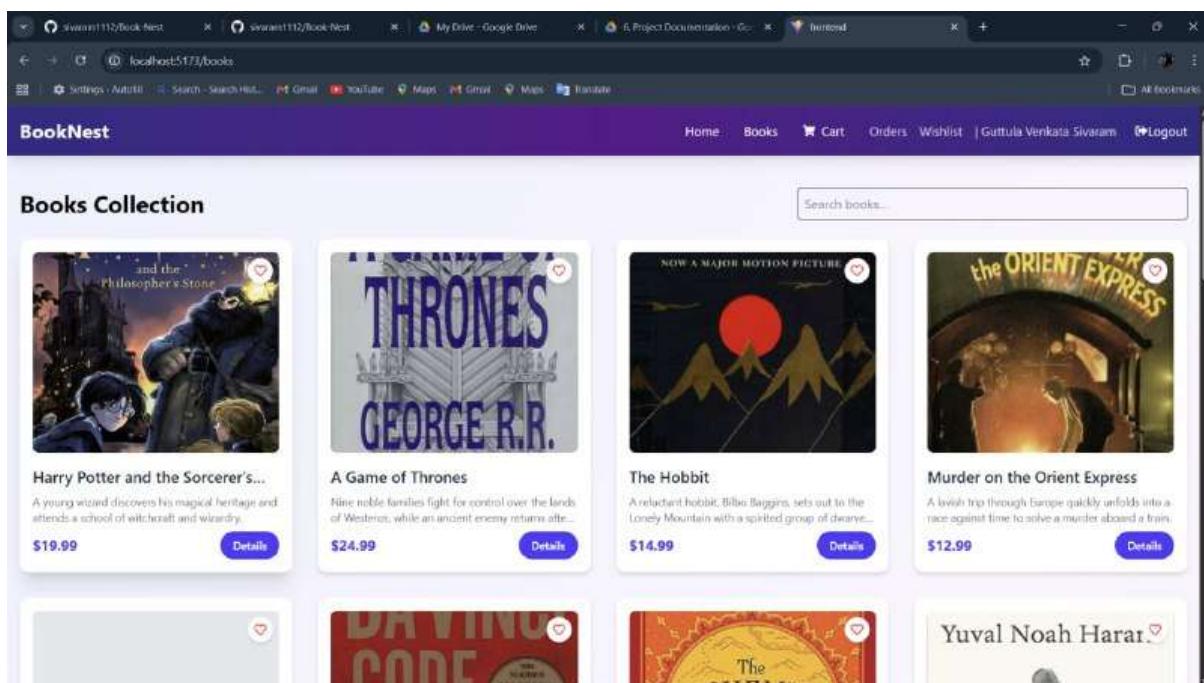


Figure-3: Collection of Books Display

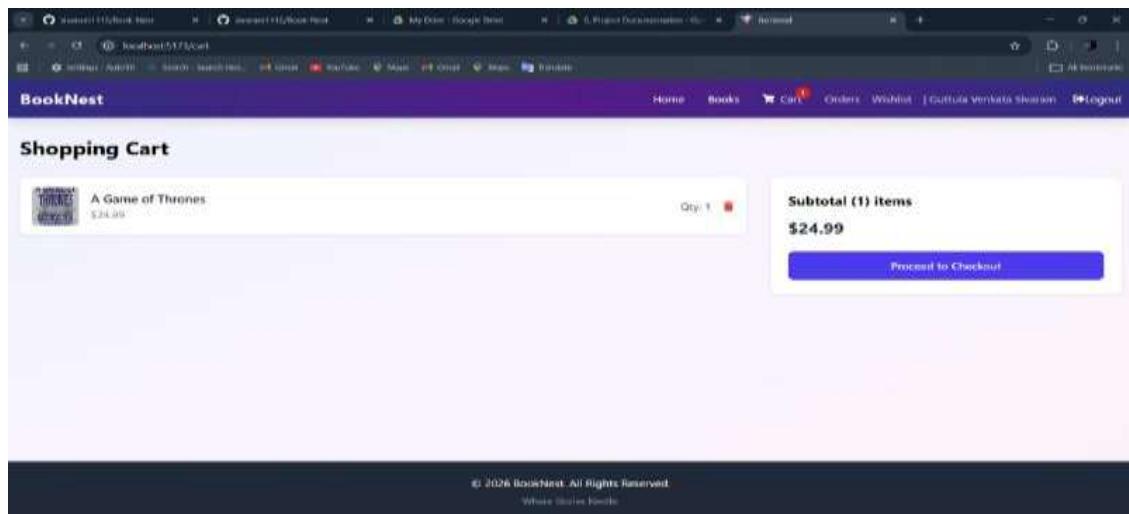


Figure-4: User Profile

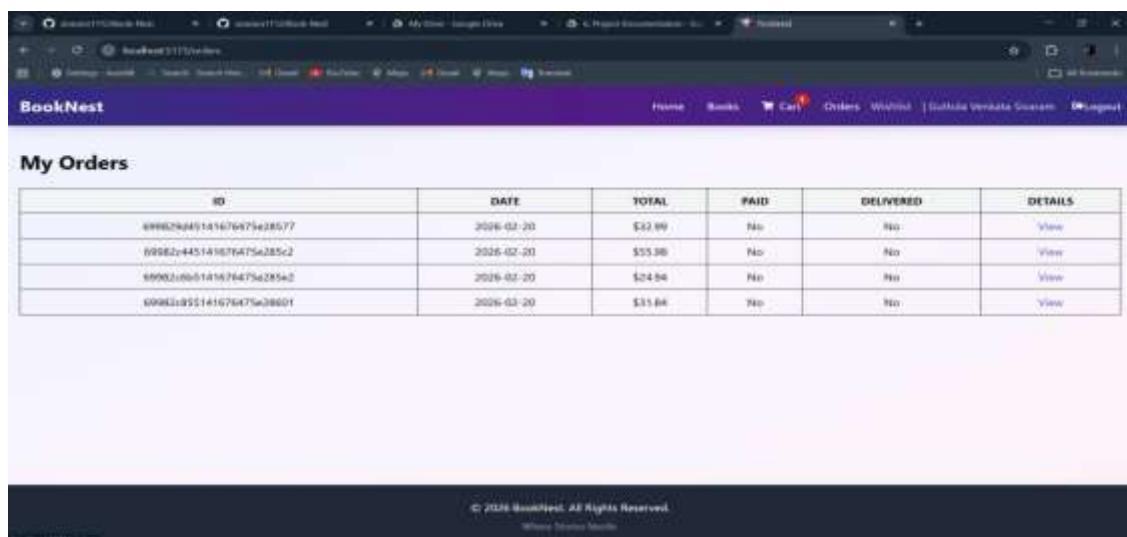
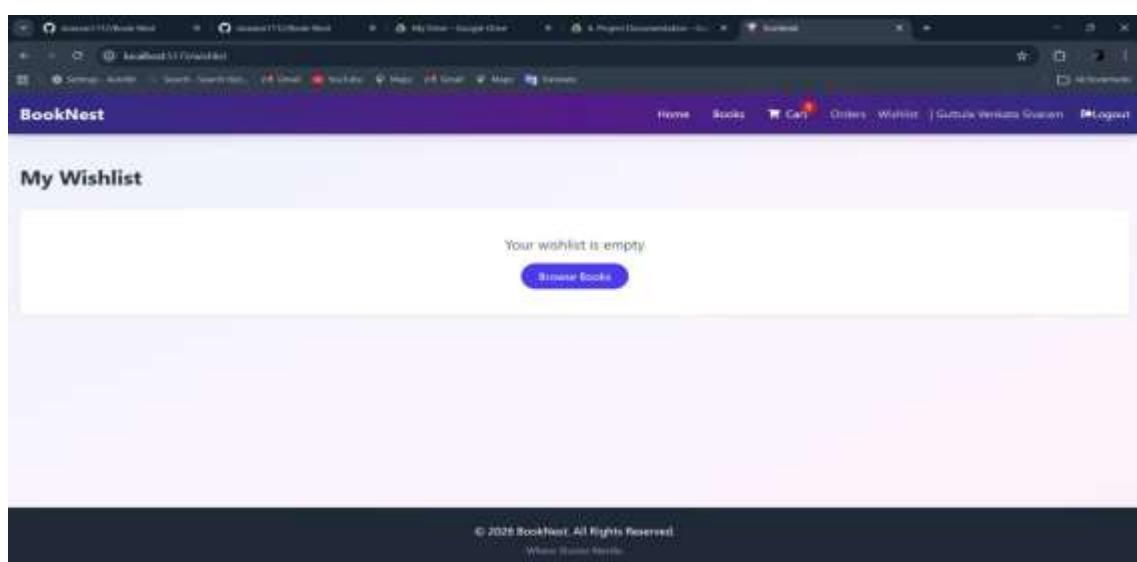
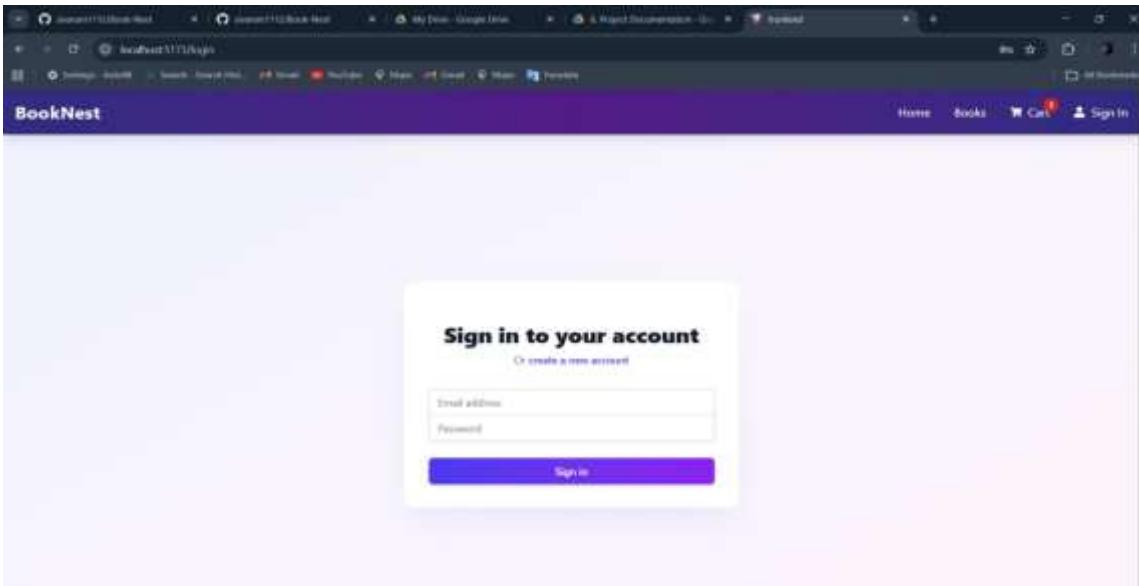


Figure-6: Order History



*Figure-5: User Wishlist Dashboard*



*Figure-6: User login page*

## 8. ADVANTAGES & DISADVANTAGES

### Advantages

- **Online Book Purchasing:** Customers can browse available books and place orders instantly, reducing manual effort and saving time.
- **Role-Based Access Control:** Separate dashboards for Admin, Seller, and Customer improve platform management and workflow organization.
- **Cloud Database Integration:** Using MongoDB Atlas ensures scalability, data security, and remote access to book, user, and order data.
- **Secure Authentication:** JWT-based login protects user accounts and restricts unauthorized access.
- **Digital Inventory Management:** Sellers can manage book listings and stock digitally, improving sales efficiency.

### Disadvantages

- Requires stable internet connectivity to access the platform.
- Image and data loading performance may depend on network speed.
- Advanced features like payment gateway and real-time notifications may require further optimization.

## 9. CONCLUSION

The BookNest Online Book Store application successfully addresses the limitations of traditional book purchasing by providing a centralized digital platform for browsing, ordering,

and managing books. Developed using modern MERN stack technologies such as React.js, Node.js, Express.js, and MongoDB Atlas, the system enables smooth interaction between customers, sellers, and administrators.

The platform improves accessibility to books by offering organized listings, secure authentication, inventory management, and order tracking. It also demonstrates practical implementation of full-stack development concepts, RESTful APIs, role-based access control, and cloud database integration.

Overall, the BookNest project delivers an efficient, scalable, and user-friendly online bookstore solution while enhancing customer convenience and supporting digital commerce growth

## 10. FUTURE SCOPE

- Development of a mobile application for Android and iOS platforms.
- Integration of an online payment gateway for secure digital transactions.
- Implementation of AI-based book recommendation system based on user interests.
- Support for e-book downloads and digital reading.
- Real-time notifications for order updates using WebSockets or Firebase.
- Advanced analytics dashboard for sellers and administrators..

## 12. APPENDIX

### GitHub Repository:

<https://github.com/sivaram1112/Book-Nest>

### Live Demo / Project Video:

<https://drive.google.com/file/d/1zdVhn28XzizmyGoCftTwDqmyjpDuQjOF/view?usp=sharing>