📘 BOOKNEST – PROJECT REPORT

# 1. INTRODUCTION

## 1.1 Project Overview

BookNest is a full-stack web application built using the MERN (MongoDB, Express, React, Node.js) stack that serves as an online book-selling platform. It supports role-based access for users, sellers, and admins.

## 1.2 Purpose

The purpose is to streamline book selling and buying digitally by offering an intuitive interface for customers, vendors, and platform admins to manage their respective operations.

# 2. IDEATION PHASE

## 2.1 Problem Statement

There is no dedicated platform for small-scale book sellers and users that combines inventory management, order processing, and admin oversight in a single application.

## 2.2 Empathy Map Canvas

Think & Feel: Frustrated by confusing online book interfaces  
See: Overloaded listings, untrusted platforms  
Say & Do: Want quick navigation and simple checkout  
Hear: Friends saying existing platforms are complex  
Pain: No seller dashboard, unorganized orders  
Gain: Streamlined book operations and better access

## 2.3 Brainstorming

• Role-based dashboard (user, seller, admin)  
• Cart and order management  
• CRUD for books  
• Admin analytics  
• Book image support via URL  
• Login & register authentication

# 3. REQUIREMENT ANALYSIS

## 3.1 Customer Journey map

1. User: Register → Browse books → Add to cart → Place order → View order  
2. Seller: Login → Add books → View own products → Edit/Delete  
3. Admin: View users/orders/books → Update order status → Delete users/books

## 3.2 Solution Requirement

• Functional roles (user, seller, admin)  
• Auth token-based routing  
• MongoDB for persistent storage  
• Protected API routes

## 3.3 Data Flow Diagram

[Frontend] → [React App] → API Calls → [Express Server] → Authentication → MongoDB

## 3.4 Technology Stack

• Frontend: React.js, Axios  
• Backend: Node.js, Express.js  
• Database: MongoDB (Compass)  
• Authentication: JWT  
• Charting: Recharts  
• UI: Custom CSS

# 4. PROJECT DESIGN

## 4.1 Problem Solution Fit

The solution bridges the gap between individual sellers and book buyers with admin controls to ensure quality and order tracking.

## 4.2 Proposed Solution

• Central platform to manage books  
• Seller-specific book inventory  
• Admin dashboard with metrics  
• User-friendly cart & order management

## 4.3 Solution Architecture

• Component-based React frontend  
• RESTful APIs  
• Role-based rendering and routing  
• Token-based access control

# 5. PROJECT PLANNING & SCHEDULING

## 5.1 Project Planning

• Requirements Gathering: 2 days  
• UI/UX Design: 3 days  
• Backend Development: 4 days  
• Frontend Integration: 4 days  
• Testing & Debugging: 2 days  
• Deployment: 1 day

# 6. FUNCTIONAL AND PERFORMANCE TESTING

## 6.1 Performance Testing

• Load tested MongoDB queries  
• Verified API latency  
• Optimized token verification

# 7. RESULTS

## 7.1 Output Screenshots

[https://drive.google.com/drive/folders/1nikZEsVk\_QoKP-9KqPUXA-AbBjTKJFwp?usp=drive\_link](#_7.1_Output_Screenshots)

# 8. ADVANTAGES & DISADVANTAGES

✅ Advantages:  
• Role-based access  
• Modular architecture  
• Live feedback and metrics  
  
❌ Disadvantages:  
• No file upload for images  
• Basic validation only  
• Still in local deployment

# 9. CONCLUSION

BookNest achieves its objective of providing a full-fledged platform for managing book sales with simplicity, role-based features, and extendable structure.

# 10. FUTURE SCOPE

• File/image upload support  
• Ratings and reviews  
• Notifications  
• Responsive mobile layout  
• Payment gateway integration

# 11. APPENDIX

• Source Code: [https://github.com/Satwikindupuri/BookNest.git](#_11._APPENDIX)  
• Project Demo: [https://drive.google.com/drive/folders/1nikZEsVk\_QoKP-9KqPUXA-AbBjTKJFwp?usp=drive\_link](#_11._APPENDIX)