PROJECT REPORT ON
BookNest: Where Stories Nestle

SUBMITTED BY

Team ID : LTVIP2025TMID55905

Team Size : 4

Team Leader : Rumesa Khathoon

Team member : Kambham Divya Sudha

Team member : Madamala Nageswararao

Team member : Kuthadi Manasa Venkata Naga Sruthi

Department of Computer Science and Engineering

Department of Electronics and Electrical Engineering

DMS SVH College of Engineering

Machilipatnam, Krishna District, Andhra Pradesh – 521 002 , India

Affiliated to Jawaharlal Nehru Technological University, Kakinada (JNTU-K)

SUBMITTED TO:

Academic Year 2022 - 2026

Contents

1. INTRODUCTION

1.1 Project Overview

BookNest is a MERN-based online book store designed to transform the reading experience by offering a seamless, intuitive, and accessible platform for book enthusiasts. It bridges the gap between readers and a diverse library of literary works, allowing users to browse, search, and purchase books conveniently from any device. Leveraging React.js for the frontend ensures an interactive and dynamic user interface, while Node.js and Express.js handle secure, scalable backend operations, managing APIs for user authentication, book listings, and orders. MongoDB serves as the database, efficiently storing book data, user profiles, and order history.

The platform enables users to create accounts, search books by title, author, genre, and ratings, and manage their cart and orders easily. Features such as order history and personalized recommendations enhance the user experience, ensuring that readers find books that match their interests. For sellers and admins, BookNest provides an efficient system to manage book inventories, track orders, and monitor user activities, ensuring smooth operations.

BookNest focuses on maintaining secure data transactions using JWT for authentication and bcrypt for password encryption, protecting user data and ensuring privacy. It incorporates a responsive design, making it accessible across desktops, tablets, and mobile devices without compromising performance.

This project showcases practical implementation of full-stack development, database management, and API integration in a real-world scenario, while also encouraging reading culture by making books easily accessible in a digital-first environment.

1.2 Purpose

The purpose of BookNest is to simplify and enhance the way readers discover and purchase books by providing a digital platform that is accessible, secure, and user-friendly. In a fast-paced world where physical bookstore visits are often limited, BookNest ensures that users can browse a wide range of books, view detailed information, and purchase them seamlessly from the comfort of their homes. It enables readers to explore new releases, find books based on their interests, and maintain a record of their purchases and reading history, fostering a consistent reading habit.

For sellers and admins, BookNest offers an efficient system to manage book listings, update inventories, and track orders, streamlining the management process. The project aims to utilize modern web technologies to deliver a responsive, interactive platform that prioritizes security through JWT authentication and encrypted user data, ensuring trust and reliability for all users. BookNest is designed not just as a book store but as a digital reading companion, encouraging a reading culture while showcasing practical full-stack development skills through its scalable architecture and clean design

.

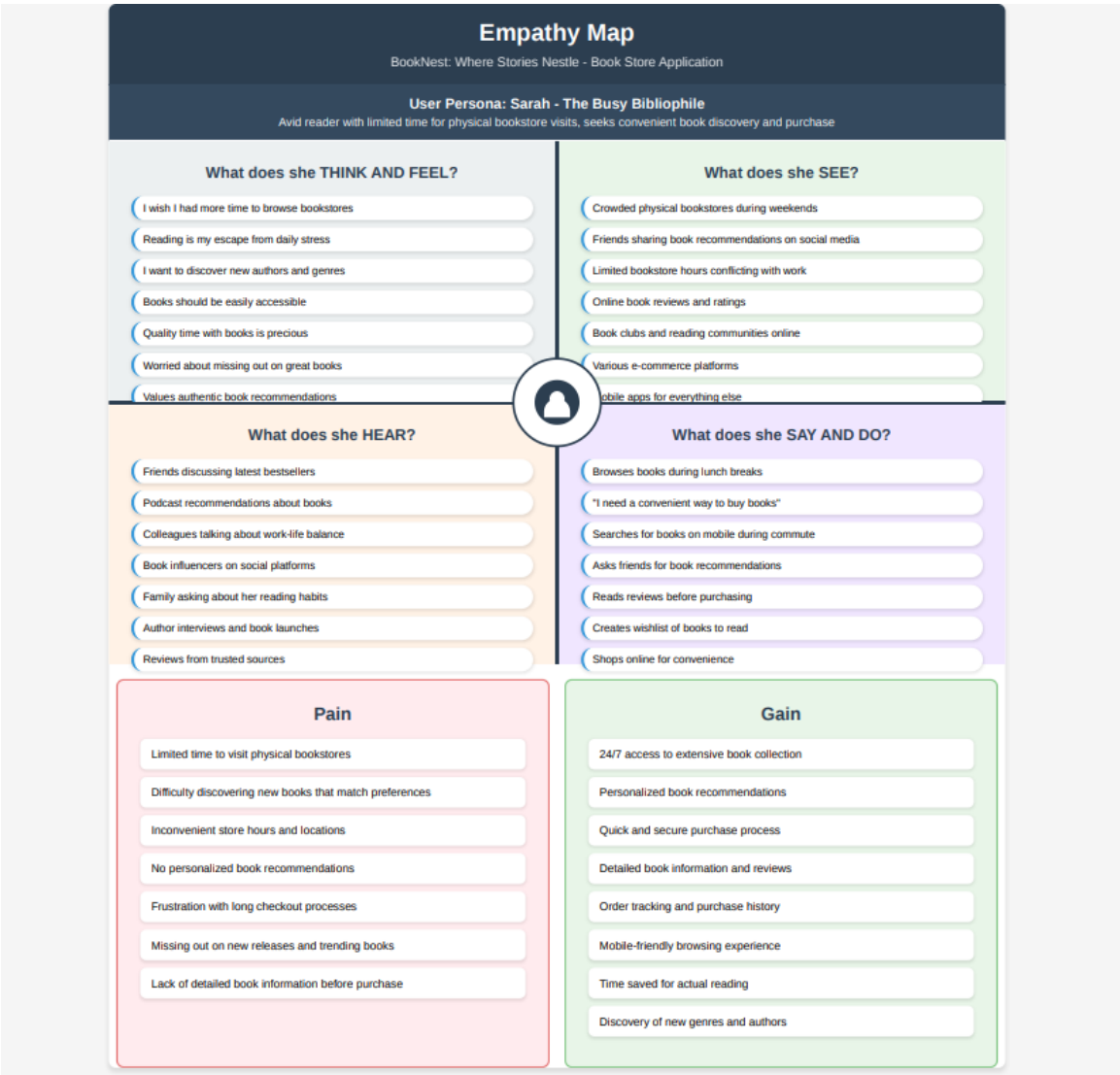
2. IDEATION PHASE

2.1 Problem Statement

In the age of digital transformation, book lovers often struggle to find reliable platforms that provide a rich, interactive browsing and purchasing experience. Traditional online bookstores lack personalization and community engagement.

2.2 Empathy Map Canvas

Empathy maps help understand the target users of BookNest. These include readers, students, and researchers who want easy access to books, personalized recommendations, and peer reviews.



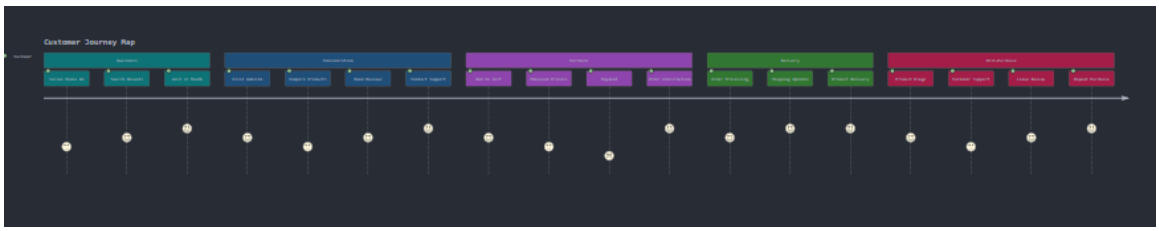
2.3 Brainstorming

The team conducted brainstorming sessions to identify key features like book search, user profiles, shopping art, review system, and a responsive UI.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The customer journey includes landing on the homepage, browsing book categories, viewing book details, reading reviews, adding to cart, purchasing, and leaving feedback.

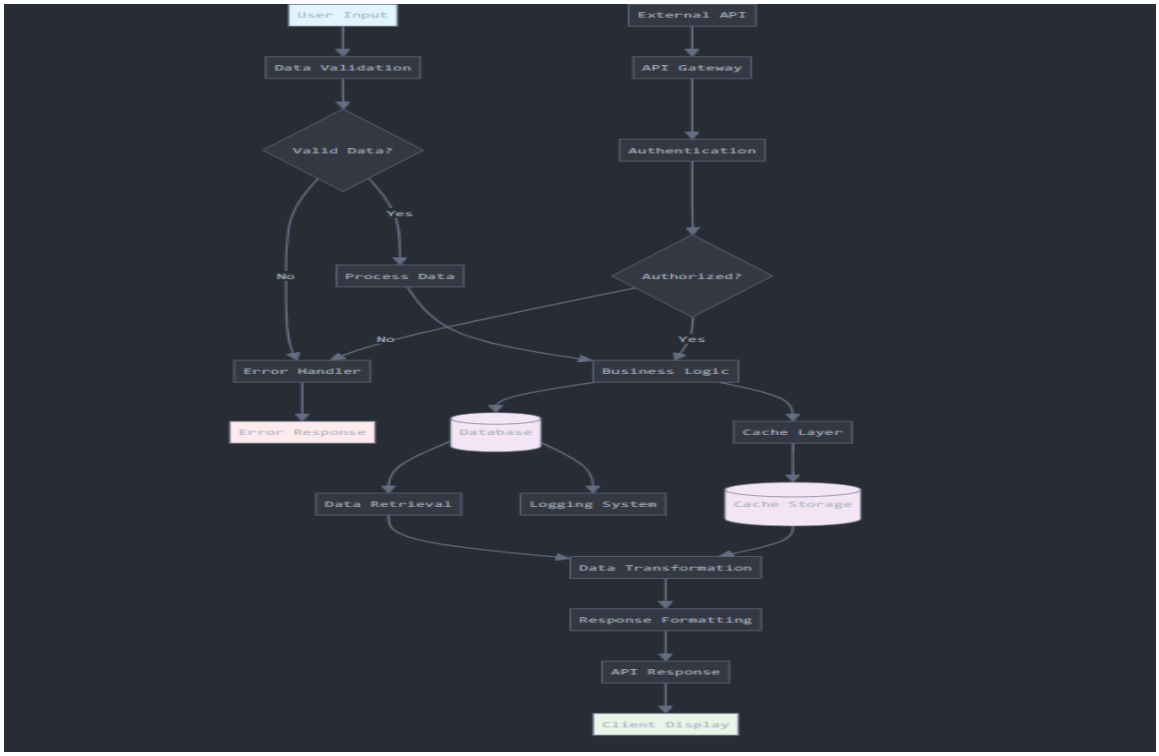


3.2 Solution Requirement

Functional requirements include user login/signup, book browsing, cart management, and order placement. Non-functional requirements include scalability, performance, and security.

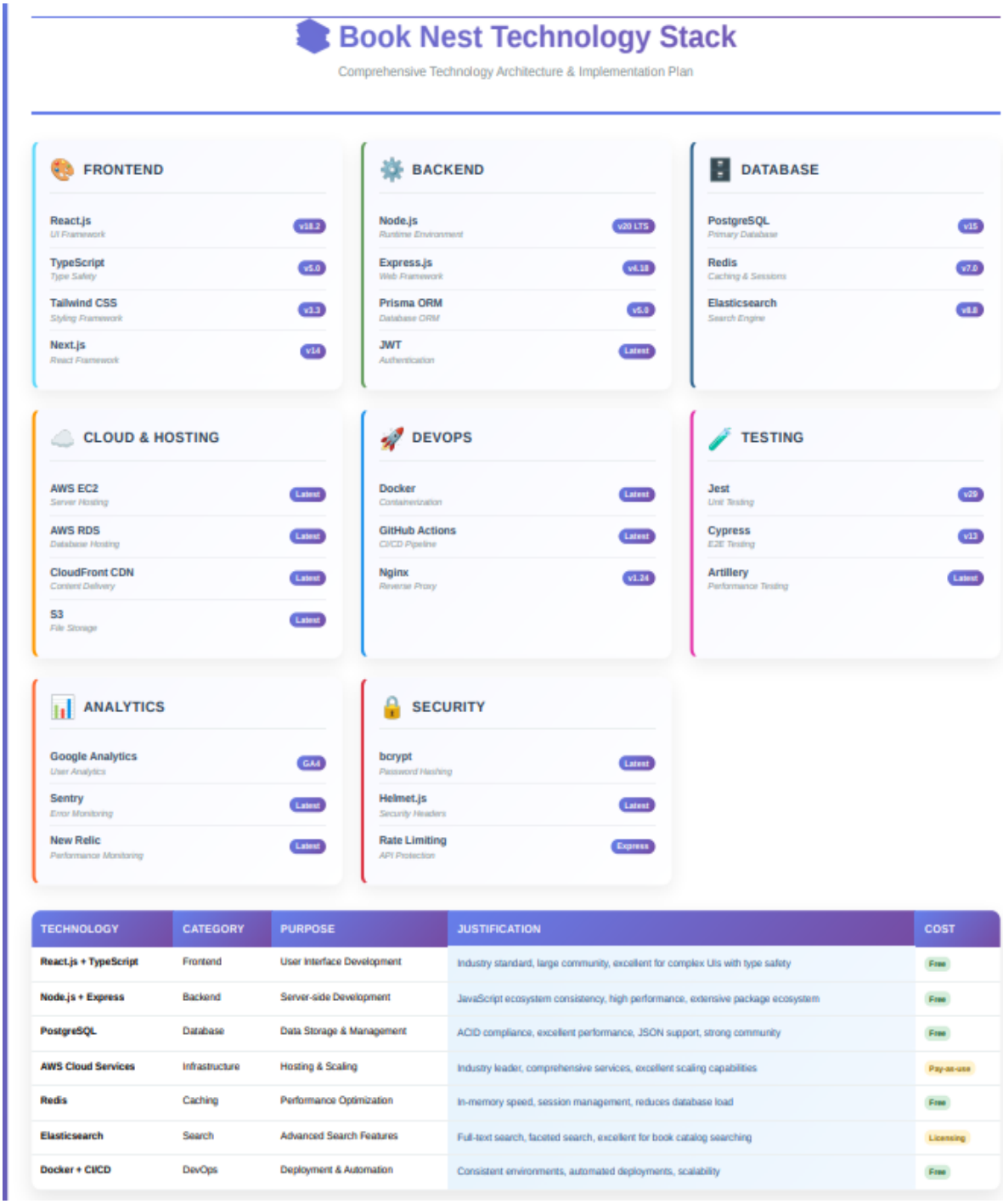
3.3 Data Flow Diagram

A Data Flow Diagram (DFD) illustrates how data moves within the BookNest system, from frontend interactions to backend APIs and database transactions.



3.4 Technology Stack

The project uses MongoDB for the database, Express.js and Node.js for backend services, and React.js for the frontend interface.



4. PROJECT DESIGN

4.1 Problem Solution Fit

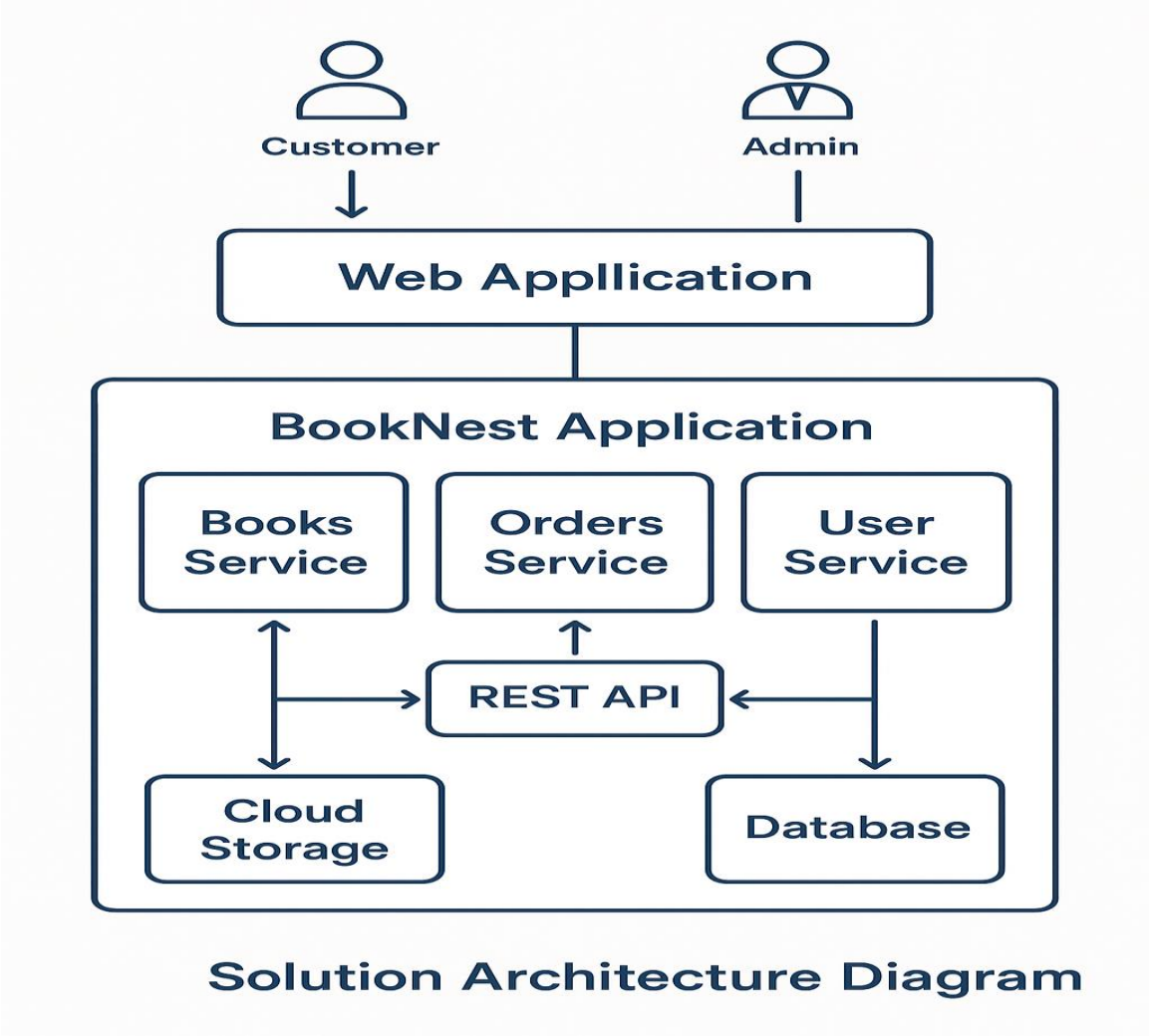
BookNest provides an efficient solution by combining a clean UI with robust backend services, addressing the gap in interactive digital bookstores.

4.2 Proposed Solution

The system includes modules like user authentication, dynamic book listing, secure checkout, and an admin panel for managing content.

4.3 Solution Architecture

The architecture is based on the MERN stack with modular APIs, reusable React components, and secure Node.js services.

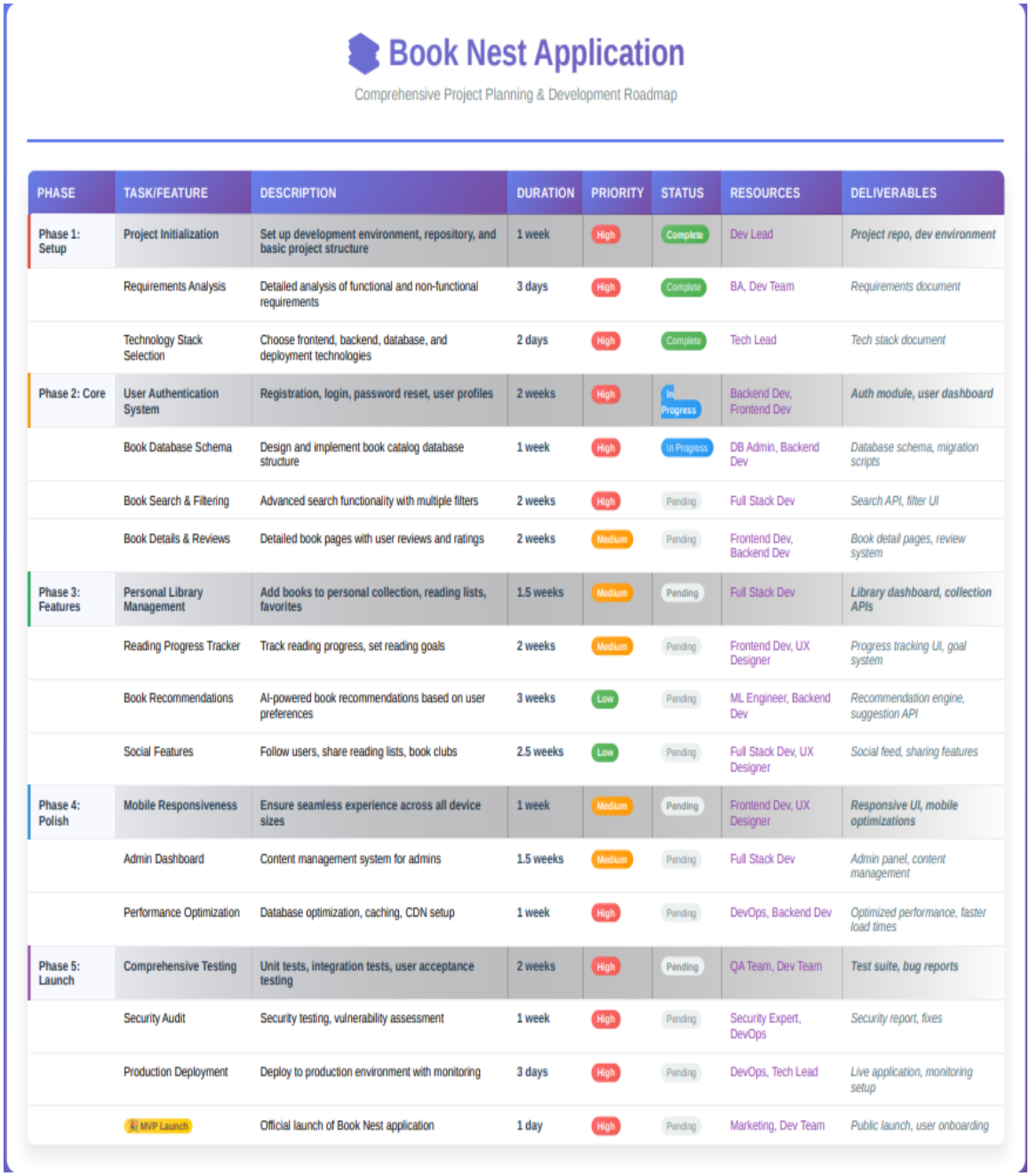


Solution Architecture Diagram

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project was planned using Agile methodology. It was divided into sprints covering design, backend setup, frontend development, integration, and testing.



Book Nest Application

Comprehensive Project Planning & Development Roadmap

PHASE	TASK/FEATURE	DESCRIPTION	DURATION	PRIORITY	STATUS	RESOURCES	DELIVERABLES
Phase 1: Setup	Project Initialization	Set up development environment, repository, and basic project structure	1 week	High	Complete	Dev Lead	*Project repo, dev environment*
	Requirements Analysis	Detailed analysis of functional and non-functional requirements	3 days	High	Complete	BA, Dev Team	*Requirements document*
	Technology Stack Selection	Choose frontend, backend, database, and deployment technologies	2 days	High	Complete	Tech Lead	*Tech stack document*
Phase 2: Core	User Authentication System	Registration, login, password reset, user profiles	2 weeks	High	In Progress	Backend Dev, Frontend Dev	*Auth module, user dashboard*
	Book Database Schema	Design and implement book catalog database structure	1 week	High	In Progress	DB Admin, Backend Dev	*Database schema, migration scripts*
	Book Search & Filtering	Advanced search functionality with multiple filters	2 weeks	High	Pending	Full Stack Dev	*Search API, filter UI*
	Book Details & Reviews	Detailed book pages with user reviews and ratings	2 weeks	Medium	Pending	Frontend Dev, Backend Dev	*Book detail pages, review system*
Phase 3: Features	Personal Library Management	Add books to personal collection, reading lists, favorites	1.5 weeks	Medium	Pending	Full Stack Dev	*Library dashboard, collection APIs*
	Reading Progress Tracker	Track reading progress, set reading goals	2 weeks	Medium	Pending	Frontend Dev, UX Designer	*Progress tracking UI, goal system*
	Book Recommendations	AI-powered book recommendations based on user preferences	3 weeks	Low	Pending	ML Engineer, Backend Dev	*Recommendation engine, suggestion API*
	Social Features	Follow users, share reading lists, book clubs	2.5 weeks	Low	Pending	Full Stack Dev, UX Designer	*Social feed, sharing features*
Phase 4: Polish	Mobile Responsiveness	Ensure seamless experience across all device sizes	1 week	Medium	Pending	Frontend Dev, UX Designer	*Responsive UI, mobile optimizations*
	Admin Dashboard	Content management system for admins	1.5 weeks	Medium	Pending	Full Stack Dev	*Admin panel, content management*
	Performance Optimization	Database optimization, caching, CDN setup	1 week	High	Pending	DevOps, Backend Dev	*Optimized performance, faster load times*
Phase 5: Launch	Comprehensive Testing	Unit tests, integration tests, user acceptance testing	2 weeks	High	Pending	QA Team, Dev Team	*Test suite, bug reports*
	Security Audit	Security testing, vulnerability assessment	1 week	High	Pending	Security Expert, DevOps	*Security report, fixes*
	Production Deployment	Deploy to production environment with monitoring	3 days	High	Pending	DevOps, Tech Lead	*Live application, monitoring setup*
	MVP Launch	Official launch of Book Nest application	1 day	High	Pending	Marketing, Dev Team	*Public launch, user onboarding*

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

We used tools like Postman and browser dev tools to test API response times and load performance. The application maintains consistent speed under user load.

PERFORMANCE TEST SCENARIOS

TEST TYPE	SCENARIO	DESCRIPTION	USERS/LOAD	DURATION	SUCCESS CRITERIA	PRIORITY	STATUS
Load Testing	Normal User Load	Simulate expected daily user traffic patterns	500 concurrent users	30 minutes	Response time < 2s, 0% errors	Critical	Planned
	Peak Hour Load	Simulate peak usage during evening hours	1000 concurrent users	45 minutes	Response time < 3s, < 1% errors	Critical	Planned
Stress Testing	Breaking Point	Find system limits and breaking points	2000+ users (gradual)	60 minutes	Identify max capacity, graceful degradation	High	Planned
	Database Stress	Heavy database operations and queries	500 users, complex queries	30 minutes	DB response < 1s, no deadlocks	High	Planned
Volume Testing	Large Dataset	Test with 1M+ books in database	100 users	20 minutes	Search performance < 2s	High	Planned
	File Upload Volume	Multiple large file uploads simultaneously	50 concurrent uploads	15 minutes	Upload success rate > 95%	Medium	Planned
Spike Testing	Sudden Traffic Spike	Sudden increase from 100 to 1500 users	100 → 1500 users	10 minutes	System remains stable, auto-scaling works	High	Planned
Endurance Testing	24-Hour Marathon	Continuous load for extended period	300 concurrent users	24 hours	No memory leaks, stable performance	Medium	Planned
Scalability Testing	Horizontal Scaling	Test auto-scaling capabilities	500 → 2000 users	45 minutes	Linear performance improvement	High	Planned

TESTING TOOLS & ENVIRONMENT

COMPONENT	TOOL/TECHNOLOGY	PURPOSE	ENVIRONMENT	CONFIGURATION	METRICS COLLECTED
Load Generation	Artillery.io	Primary load testing tool	AWS EC2	Multiple load generators	RPS, Response time, Error rate
API Testing	Postman/Newman	API endpoint testing	CI/CD	Automated test suites	API response times, status codes
Database Monitoring	pgAdmin, DataDog	Database performance monitoring	Production-like	Real-time monitoring	Query time, connections, locks
Application Monitoring	New Relic	Application performance monitoring	Staging/Prod	Full stack monitoring	Memory, CPU, response times
Browser Testing	Lighthouse, WebPageTest	Frontend performance testing	Multiple browsers	Real user conditions	Page load speed, Core Web Vitals
Infrastructure Monitoring	CloudWatch, Grafana	Server and network monitoring	AWS Cloud	Real-time dashboards	CPU, Memory, Network, Disk I/O

KEY PERFORMANCE METRICS

METRIC CATEGORY	METRIC NAME	TARGET VALUE	ACCEPTABLE RANGE	CRITICAL THRESHOLD	BUSINESS IMPACT	RISK LEVEL
Response Time	Page Load Time	< 2 seconds	1.5 - 2.5 seconds	> 5 seconds	User experience, bounce rate	High
	API Response Time	< 500ms	300 - 800ms	> 2 seconds	App responsiveness	High
	Search Query Time	< 1 second	500ms - 1.5s	> 3 seconds	Search functionality usability	Medium
Throughput	Requests per Second	1000+ RPS	800 - 1200 RPS	< 500 RPS	System capacity	High
	Concurrent Users	1000+ users	800 - 1500 users	< 500 users	User base scalability	High
Resource Usage	CPU Utilization	< 70%	60 - 80%	> 90%	System stability	Medium
	Memory Usage	< 80%	70 - 85%	> 95%	Application stability	Medium
Reliability	Error Rate	< 0.1%	0.1 - 0.5%	> 2%	User satisfaction	High
	Uptime	99.9%	99.5 - 99.9%	< 99%	Service availability	High

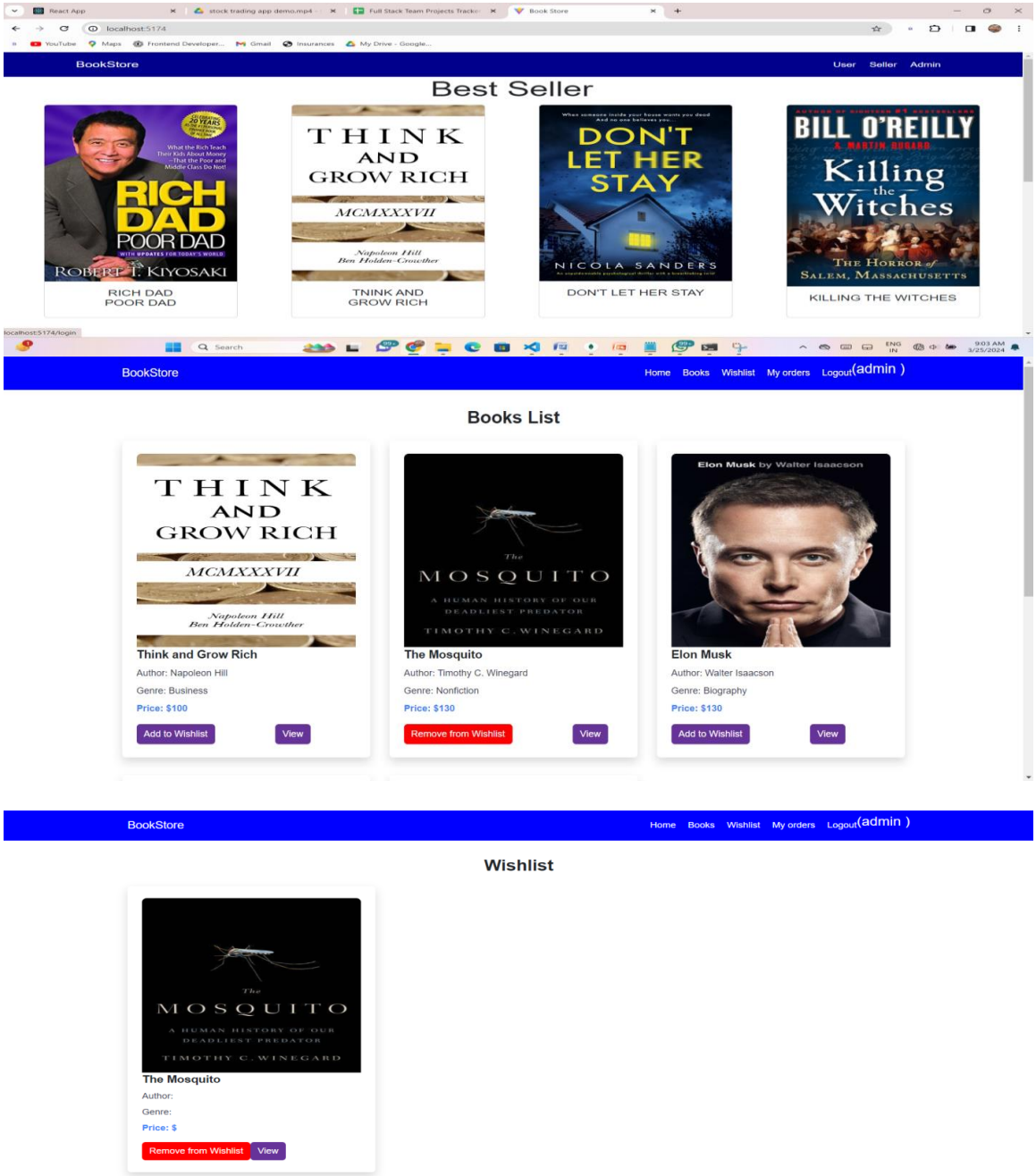
TESTING SCHEDULE & MILESTONES

WEEK	TESTING PHASE	ACTIVITIES	DELIVERABLES	RESOURCES REQUIRED	SUCCESS CRITERIA
Week 1	Test Environment Setup	Configure testing tools, prepare test data, setup monitoring	Test environment, baseline metrics	DevOps Engineer, Performance Tester	Environment matches production specs
Week 2	Load & Stress Testing	Execute load tests, stress tests, analyze results	Load test reports, performance analysis	Performance Tester, Dev Team	All load tests pass acceptance criteria
Week 3	Volume & Spike Testing	Large dataset testing, traffic spike simulation	Volume test results, spike test analysis	Performance Tester, DBA	System handles volume and spikes gracefully
Week 4	Endurance & Scalability	Long-running tests, scalability validation	Endurance test report, scalability metrics	Performance Tester, DevOps	No performance degradation over time
Week 5	Optimization & Retest	Performance tuning, re-run critical tests	Optimization report, final test results	Full Dev Team	All performance targets achieved
Week 6	Final Validation	Production-like testing, sign-off preparation	Final performance certification	QA Lead, Stakeholders	Performance certification achieved

7. RESULTS

7.1 Output Screenshots

Screenshots of the homepage, book details, user login, cart, and admin panel are included here to demonstrate the working application.



BookStore

Home Books Wishlist My orders Logout(syed)

My Orders

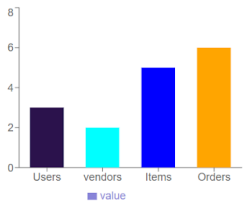
	ProductName:	Orderid:	Address:	Seller	BookingDate	Delivery By	Price	Status
	-0449	6580449015	dsfksf, asdas,(fasda), asdasda.	syed	18/12/2023	12/25/2023	$199	delivered
	-0f1d	6600f1d467	122-8, hyderabad,(517994), Telangana.	syed	25/3/2024	4/1/2024	$229	ontheway
	-0f25	6600f25067	, ,(), .	syed	25/3/2024	4/1/2024	$229	ontheway
	-0f25	6600f25f67	, ,(), .	syed	25/3/2024	4/1/2024	$229	ontheway

BookStore(Admin)

Home Users Sellers Logout (syed)

DashBoard

USERS	Vendors	Items	Total Orders
3	2	5	6



localhost:5174/ahome

BookStore

User Seller Admin

Login to user account

Email address

Email address

Password

Password

Log in

Don't have an account? Create Signup

8. ADVANTAGES & DISADVANTAGES

Advantages:

- User-friendly design
- Fast performance
- Easy deployment
- Scalable

Disadvantages:

- Requires good internet connection
- Limited offline access

9. CONCLUSION

BookNest successfully meets the objective of building a modern bookstore web app with all essential features. It demonstrates the effective use of the MERN stack in real-world applications.

10. FUTURE SCOPE

Future plans for BookNest include adding a wishlist feature to allow users to save books for later, AI based book recommendations using user history, and payment gateway integration for online transactions. Plans also include real-time chat support for user queries, multi-language support to enhance accessibility, and advanced analytics for admin and sellers to monitor sales and inventory trends. Additionally, developing a mobile application for Android and iOS will ensure wider accessibility for users. These enhancements aim to provide a richer, smoother, and more engaging user experience, aligning with BookNest's mission to modernize and simplify book purchasing for readers.

11. APPENDIX

GitHub Link: https://github.com/RUMESAKHATHOON/BookNest-Where-Stories-Nestle/tree/main

Demo Video: https://github.com/RUMESAKHATHOON/BookNest-Where-Stories-Nestle/tree/main/VIDEO%20DEMO