**PROJECT REPORT**

**1. INTRODUCTION**

**1.1 Project Overview**

Book Nest is a MERN stack-based web application that transforms the traditional bookstore experience into a digital one. It allows users to seamlessly browse, search, filter, and purchase books across various genres, languages, and authors. By integrating robust backend functionality with an intuitive frontend interface, Book Nest offers a complete user journey—from account creation to order tracking—all in one platform. The application caters to both casual readers and serious collectors by ensuring secure payments, personalized recommendations, and efficient order management.

**1.2 Purpose**

The purpose of the Book Nest project is to design and develop an online bookstore platform that simplifies the book purchasing process, enhances accessibility, and provides a personalized reading experience. The system aims to:

* Eliminate the dependency on physical bookstores.
* Support readers in discovering new books through smart recommendations.
* Provide a secure and convenient online shopping experience.
* Enable users to track their orders and manage wish lists.
* Build a scalable and responsive solution using modern web technologies**.**

**2. IDEATION PHASE**

**2.1 Problem Statement**

**Customer Problem Statement:**

Link: <https://miro.com/app/board/uXjVIkhiyxk=/?share_link_id=744776086381>



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Statement (PS)** | **I am (Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| **PS-1** | a busy college student with limited time | find and order academic books quickly and affordably | many sites are cluttered or have poor search filters | they aren’t optimized for student needs | overwhelmed and frustrated when I can’t find the right book in time |
| **PS-2** | a parent looking for books for my child | discover safe, age-appropriate books with ease | recommendations are not personalized or detailed | platforms don’t filter content by age or educational value | concerned and hesitant to let my child browse alone |
| **PS-3** | an avid reader on a tight budget | track discounts and wishlist books | there’s no reminder or price-drop alert system | most stores lack notification tools | disappointed when I miss out on good deals |
| **PS-4** | a book collector who loves series and authors | find books by my favorite author or complete a series | listings don’t show related series/books clearly | metadata and tags are not well maintained | annoyed and reluctant to shop again |

**2.2 Empathy Map Canvas**

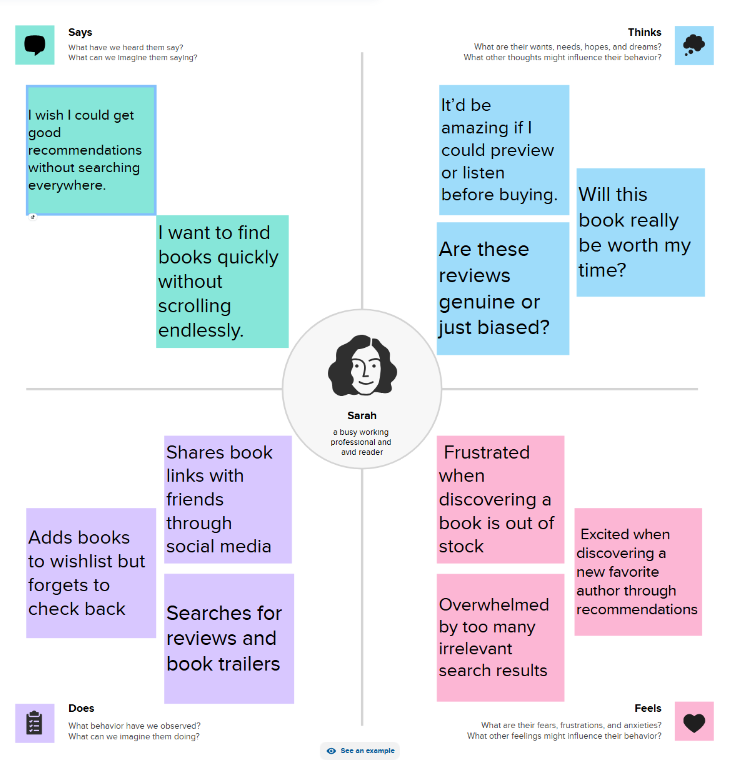
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

Link: <https://app.mural.co/t/bookorder5979/m/bookorder5979/1750780404946/8e59a8103d8a3f458ae2ffbbd263a8086050806f>

**Book Nest**



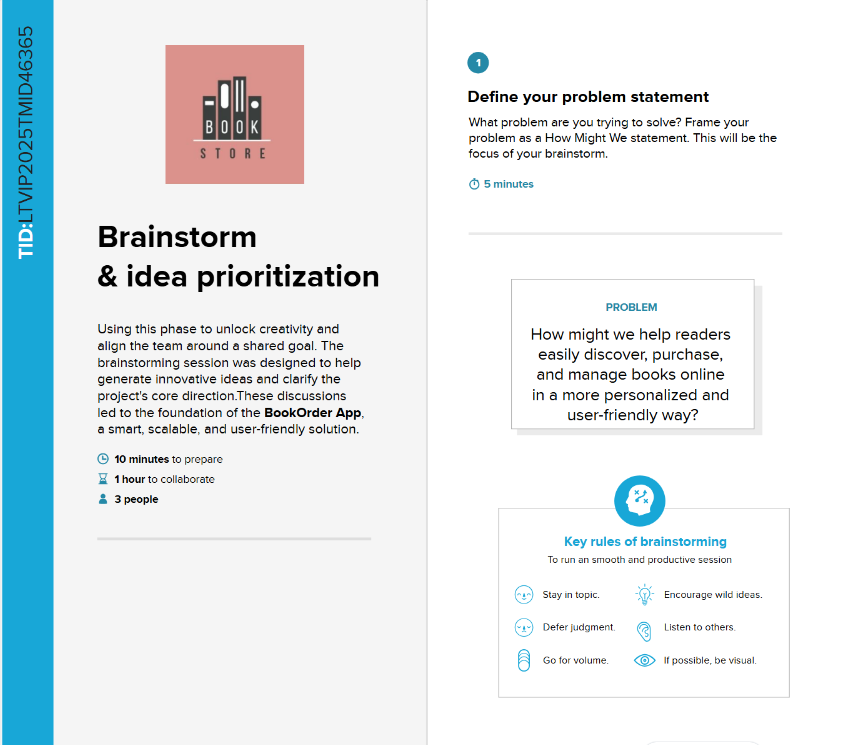
**2.3 Brainstorming**

**Brainstorm & Idea Prioritization:**

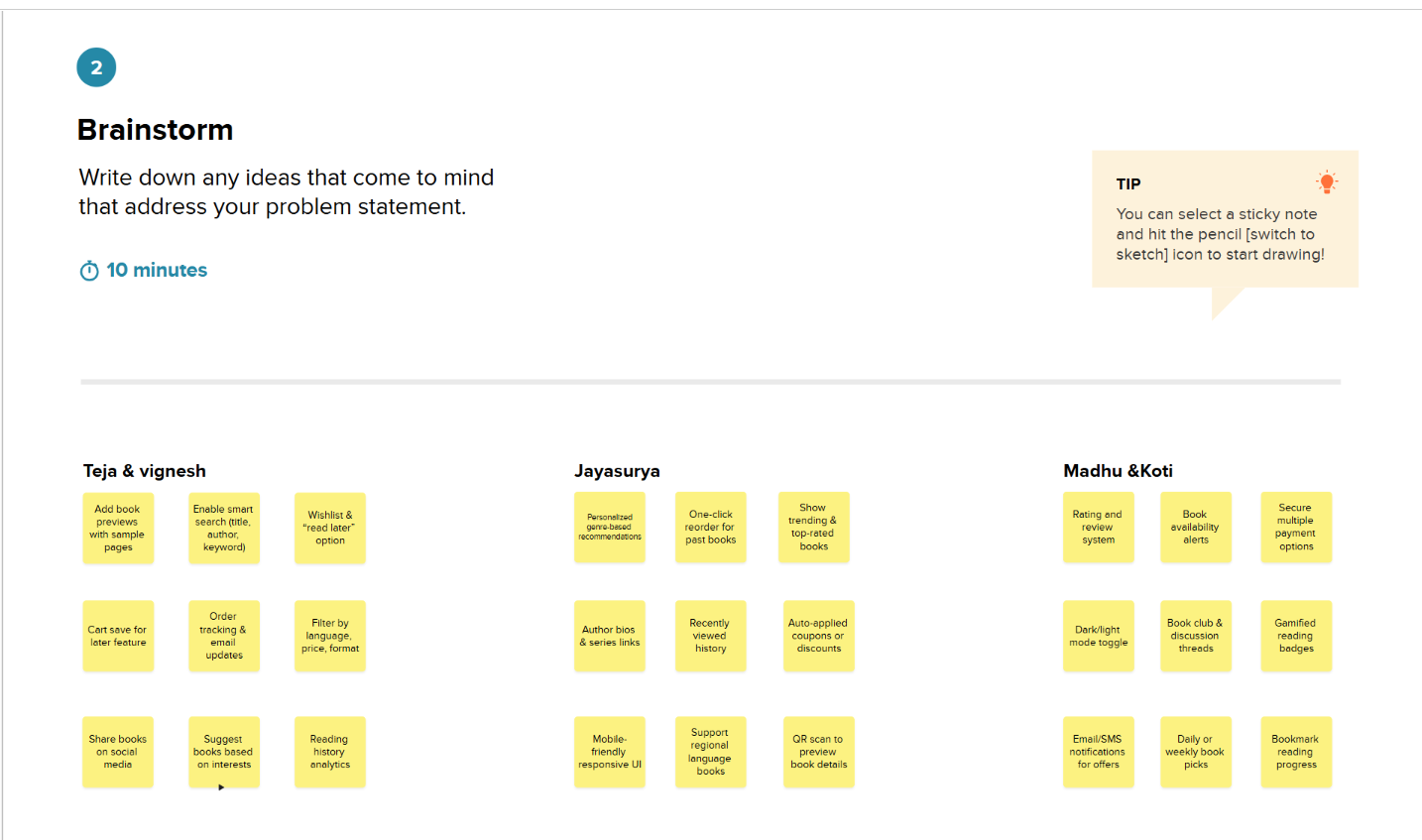
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

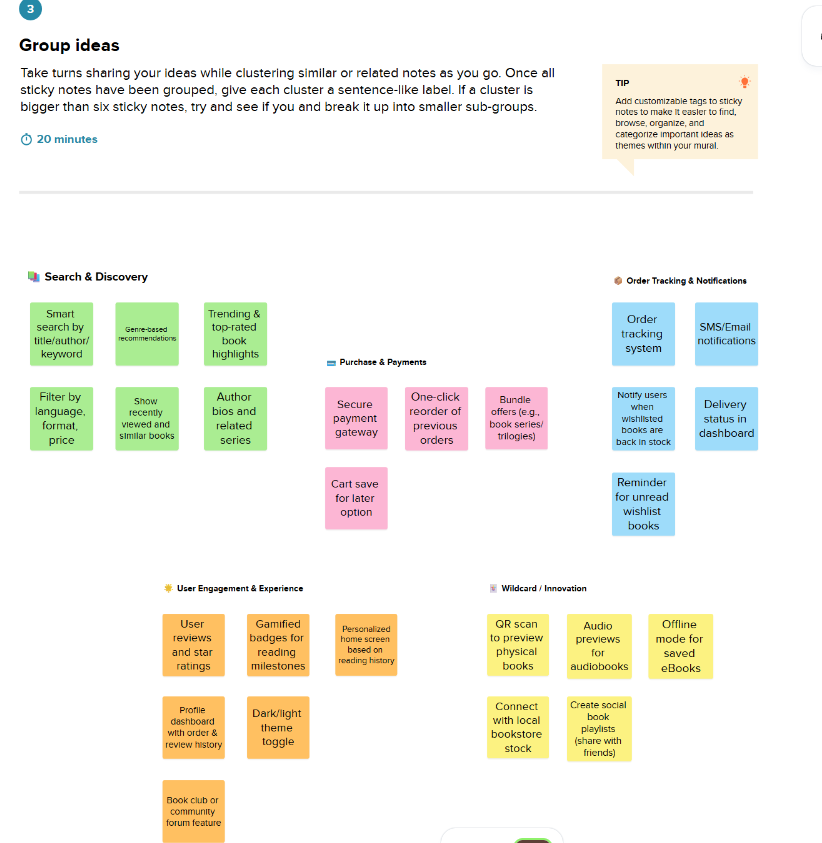
Link: <https://app.mural.co/t/bookorder0417/m/bookorder0417/1750770438520/1047a88635c51a632439c81f8caa682712057c94?sender=u427be78f24cc5d79ea230073>

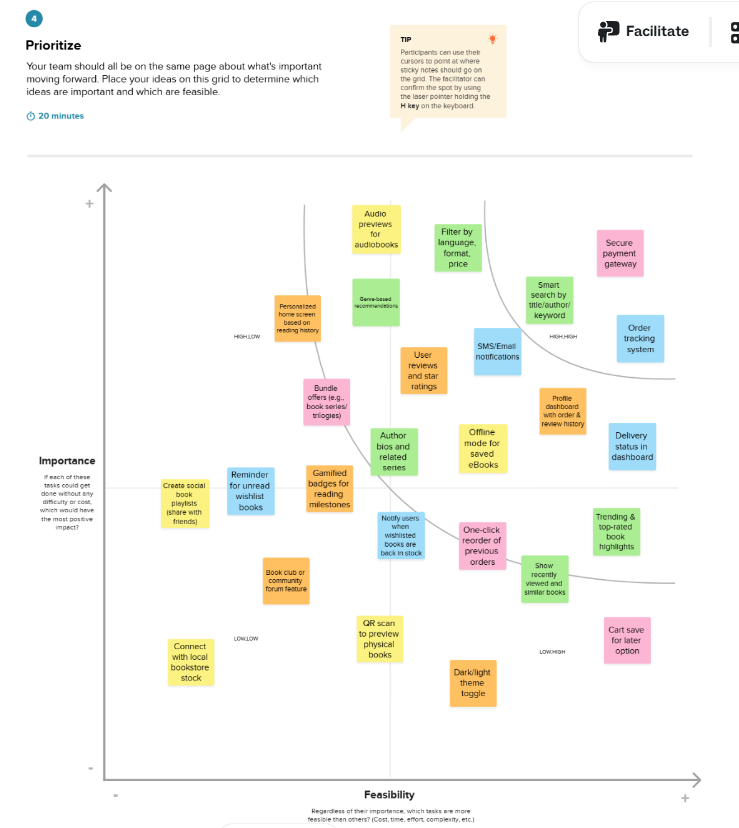
**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

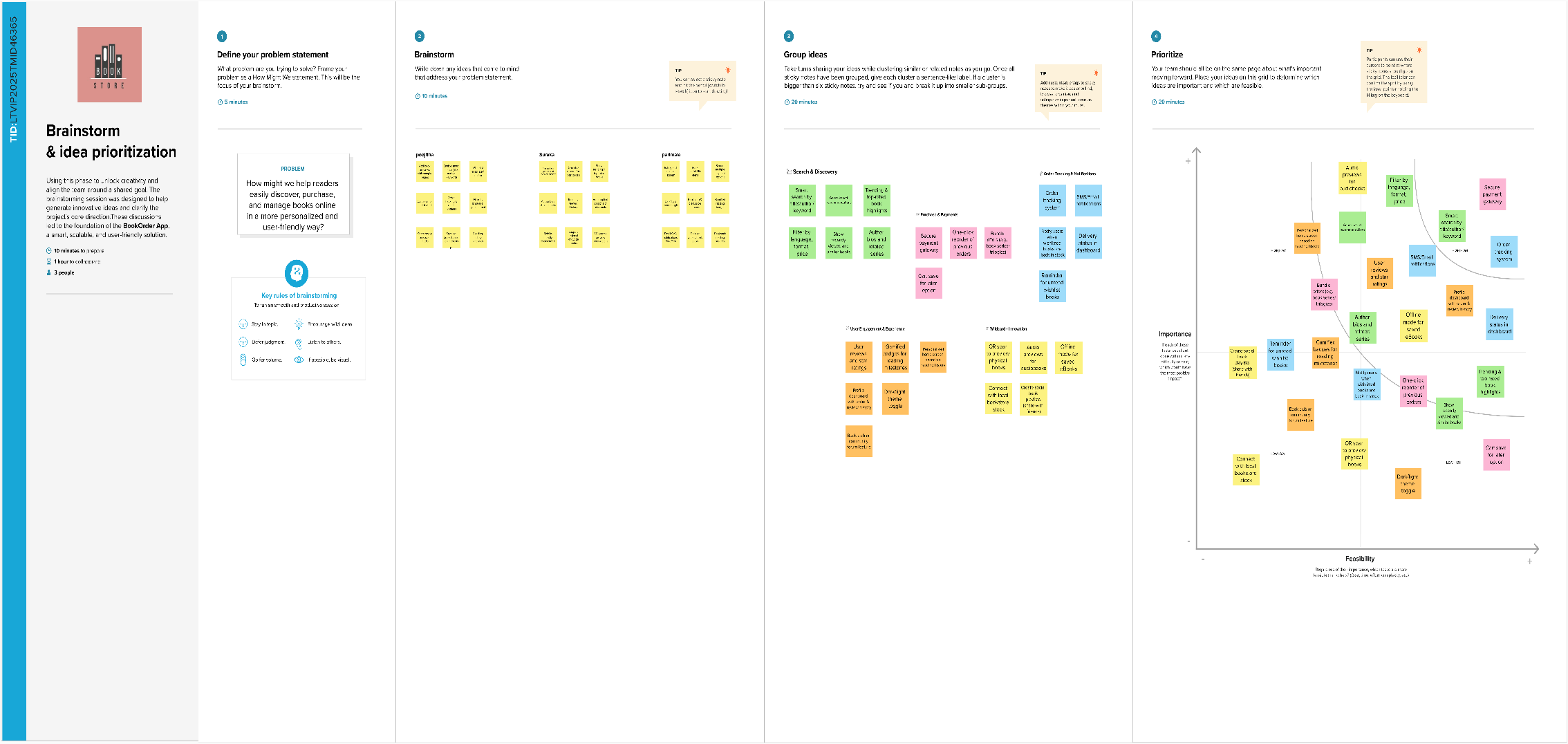
****

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

****

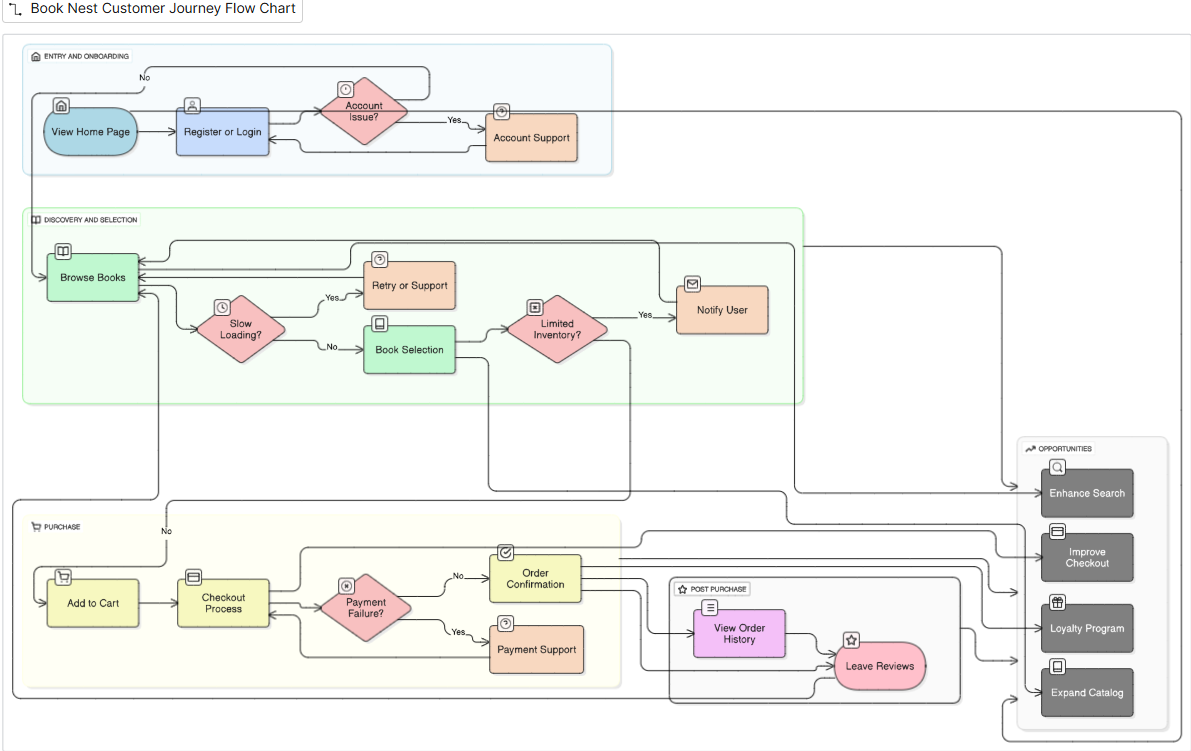


**Step-3: Idea Prioritization**



**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey map**

****

**3.2 Solution Requirement**

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

**Functional Requirements – Music Streaming App**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| **FR-1** | **User Authentication** | Sign up, Login, Password Reset, OAuth Login via Google/GitHub |
| **FR-2** | **Book Listings** | View available books, details including title, author, genre, price, and availability |
| **FR-3** | **Book Selection** | Filter by genre, author, rating, or popularity |
| **FR-4** | **Purchase Process** | Add to cart, specify quantity, proceed to secure payment |
| **FR-5** | **Order Confirmation** | Display order confirmation with order ID, price, book info |
| **FR-6** | **Order History** | View current and previous orders, track shipment, give ratings |
| **FR-7** | **Wishlist** | Add/remove books from wishlist |
| **FR-8** | **Admin Dashboard** | Manage users, books, inventory, sellers, analytics |
| **FR-9** | **Seller Dashboard** | Manage book listings, stock, pricing, orders |
| **FR-10** | **Notifications** | Email/SMS confirmations for orders and updates |

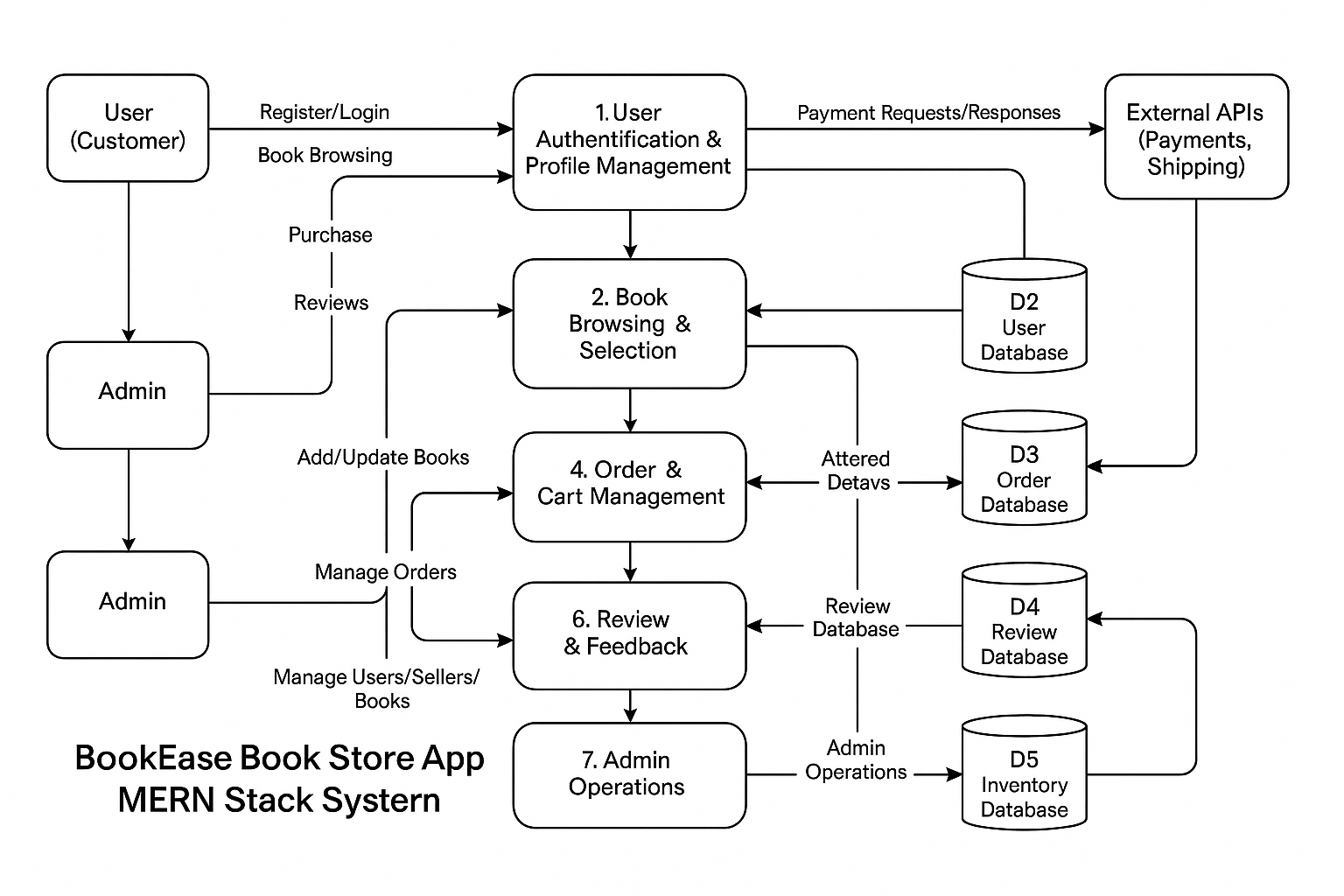
**Non-functional Requirements:**

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

|  |  |  |
| --- | --- | --- |
| **NFR No.** | **Requirement** | **Description** |
| **NFR-1** | **Usability** | UI should be intuitive and user-friendly for buyers, sellers, and admins |
| **NFR-2** | **Security** | All user and transaction data must be encrypted; implement role-based access and JWT auth |
| **NFR-3** | **Reliability** | All services including browsing, checkout, and notifications must be consistently available |
| **NFR-4** | **Performance** | Pages should load within 2 seconds; checkout and actions should respond instantly |
| **NFR-5** | **Availability** | Ensure at least 99.9% uptime with backup and recovery plans |
| **NFR-6** | **Scalability** | Must support growth in users, inventory, and orders without performance degradation |

**3.3 Data Flow Diagram**

**Data Flow Diagrams:**

****A **Data Flow Diagram (DFD)** illustrates how data moves within the Freelance Finder platform. It captures how users (freelancers and clients) interact with the system, how information flows between different components, and where the data is stored.

Example: DFD Level 0 (Industry Standard)

**User Story Table – Book Nest App**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| **User (Reader)** | **Account Management** | As a user, I can register and log in securely. | Account is created and login works correctly. | High | Sprint-1 |
|  | **Book Browsing** | As a user, I can search for books by title, author, or genre. | Relevant books are shown in the search results. | High | Sprint-1 |
|  | **Purchase Books** | As a user, I can add books to the cart and make purchases. | Orders are confirmed and inventory updates. | High | Sprint-2 |
|  | **Order History** | As a user, I can view my order history and details. | Past and current orders display correctly. | Medium | Sprint-2 |
|  | **Review System** | As a user, I can leave reviews for purchased books. | Reviews appear under respective books. | Medium | Sprint-2 |
|  | **Messaging** | As a user, I can chat with sellers or support in real-time. | Messages are sent and received instantly. | Medium | Sprint-2 |
| **Seller** | **Inventory Management** | As a seller, I can list new books and update stock. | Listings update immediately and reflect in store. | High | Sprint-1 |
|  | **Order Fulfillment** | As a seller, I can view and manage received orders. | Orders are updated with shipping and status. | High | Sprint-2 |
|  | **Profile Management** | As a seller, I can update my business profile. | Changes save successfully. | Medium | Sprint-2 |
| **Admin** | **User Management** | As an admin, I can manage user and seller accounts. | Admin can create, edit, or deactivate accounts. | High | Sprint-1 |
|  | **Book Moderation** | As an admin, I can approve or reject book listings. | Only approved books are visible in the store. | High | Sprint-1 |
|  | **Review Moderation** | As an admin, I can review reported reviews and take actions. | Actions are logged and review status is updated. | Medium | Sprint-2 |
|  | **Dashboard Management** | As an admin, I can access dashboards for analytics and performance tracking. | Metrics load correctly and update in real-time. | High | Sprint-2 |

**3.4 Technology Stack**

**Technical Architecture:**

**Book Nest** is designed with a scalable 3-tier architecture consisting of:  
● **Presentation Layer (Frontend):** User-friendly interface for users and sellers to browse and purchase books.  
**● Business Logic Layer (Backend):** Handles user authentication, book management, cart, orders, and notifications.  
● **Data Storage Layer:** Secure storage of user profiles, books, reviews, orders, and transaction records.  
The platform integrates with third-party APIs for notifications (SMS/email), payment processing, and book recommendation engines to enhance usability.

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

|  |  |  |
| --- | --- | --- |
| **S.No** | **Component** | **Description Technology** |
| **1.** | User Interface | Web and mobile-friendly interface for book browsing and purchases HTML, CSS, JavaScript / React.js |
| **2.** | Application Logic-1 | Book browsing, filtering, cart management, and order placing Node.js, Express.js |
| **3.** | Application Logic-2 | Admin and Seller panels, inventory and user management React.js, Node.js |
| **4.** | Database | Stores user profiles, books, orders, and reviews MongoDB |

**Table-2: Application Characteristics:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Characteristics** | **Description Technology** |
| **5.** | Open-Source Frameworks | Frontend frameworks React.js, Node.js, Bootstrap, Tailwind CSS |
| **6.** | Scalable Architecture | 3-tier architecture with RESTful APIs Microservices |

**References:**

[**React.js Documentation**](https://react.dev/)

[**Node js Best Practice**](https://nodejs.org/en/learn/getting-started/introduction-to-nodejs)

[**JSON Web Server Referance**](https://www.npmjs.com/package/json-server)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)

[MongoDB Docs](https://www.mongodb.com/docs/)

**4. PROJECT DESIGN**

**4.1 Problem Solution Fit**

**Problem – Solution Fit Overview:**

The Problem–Solution Fit ensures that Book Nest effectively addresses the challenges in finding, organizing, and enjoying books faced by avid readers and book lovers. This validation ensures the concept meets real user needs.

**Purpose:**

* Simplify and streamline the process of discovering and accessing books.
* Provide a centralized digital platform for organizing personal and community bookshelves.
* Enhance user engagement with recommendations, reviews, and social reading features.
* Support both casual readers and bibliophiles through personalized features and rich metadata.

### **Problem Statement:**

**Readers and book enthusiasts face challenges such as:**

* Scattered sources for discovering books and keeping track of what they’ve read or want to read.
* Lack of personalized book recommendations based on interest or history.
* Limited community features to share reviews, book lists, or participate in group discussions.
* Poor categorization and tagging in existing platforms for niche or non-mainstream books.
* Time-consuming process to manage physical and digital collections.

### **Solution:**

**Book Nest offers a seamless digital library and book discovery experience with:**

* Advanced search and filter by genre, author, tags, ratings, and more.
* Personalized dashboards to track current reads, wishlists, and completed books.
* Social features including book clubs, reviews, forums, and activity feeds.
* Recommendation engine powered by user behavior and preferences.
* Mobile and web access for syncing collections across devices.
* Telehealth integration for virtual consultations.

Admin controls for managing providers, appointments, and reporting

**4.2 Proposed Solution**

**Proposed Solution for Book Nest App**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Parameter** | **Description** |
| 1 | **Problem Statement (Problem to be solved)** | Readers often struggle to find specific books, face limited access to bookshops, and have difficulty managing purchases and order history. Book sellers face challenges in inventory tracking, order handling, and user engagement. |
| 2 | **Idea / Solution Description** | Book Nest is a full-stack online bookstore platform built with the MERN stack. It offers book discovery, recommendations, digital purchases, reviews, and inventory management. Admin and seller tools allow efficient book and order management. |
| 3 | **Novelty / Uniqueness** | • Real-time inventory updates • Role-based access for users, sellers, and admins • Secure checkout & payment integration • Wishlist and review system • Responsive UI for web and mobile |
| 4 | **Social Impact / Customer Satisfaction** | • Reduces need for physical store visits • Enables wider access to books across regions • Improves user satisfaction with reviews, wishlists, and tracking • Empowers small sellers via online presence |
| 5 | **Business Model (Revenue Model)** | • Freemium for basic sellers • Commission on every book sale • Premium seller subscriptions (analytics, visibility, ads) • B2B tie-ups with publishers and libraries |
| 6 | **Scalability of the Solution** | • Built for scalability using modular architecture • Global-ready with multi-language, multi-timezone, and multi-currency support • Mobile-first UX with future React Native or Flutter app • Modular microservices-based backend for flexibility |

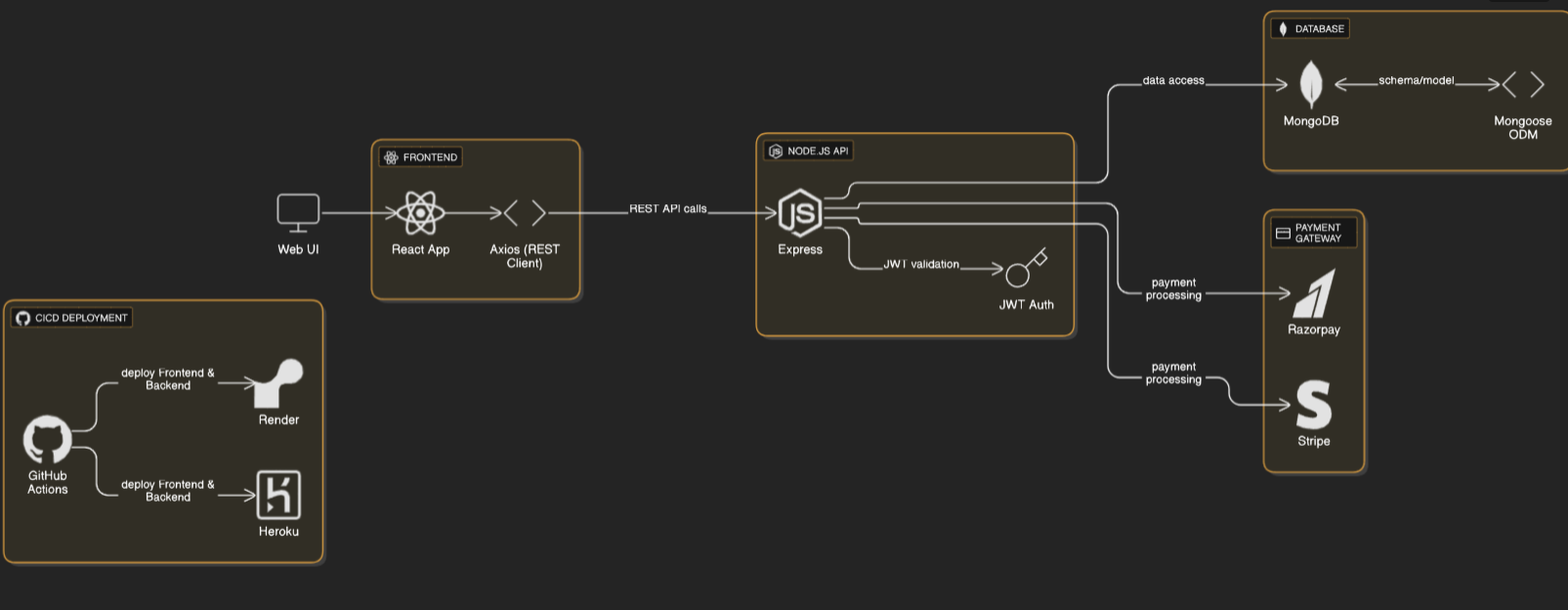
**4.3 Solution Architecture**

**Solution Architecture:**

To provide a user-friendly, secure, and scalable bookstore management system that bridges the gap between readers and sellers.

* Effortless Book Browsing and Discovery
* End-to-End Purchase & Order Management
* Scalable User Authentication and Role-Based Authorization
* Secure and Trackable Payment Transactions
* Real-time Order Tracking and Notifications

**Example - Solution Architecture Diagram:**

****

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

**Product Backlog & Sprint Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| **Sprint-1** | **User Registration & Authentication** | USN-1 | As a user, I can register and log in securely. | 3 | High | K.jayasurya  T.Ediga Brahma Koti  V.Teaja  P.Vignesh  C.Madhu Sudhakar |
|  |  | USN-2 | As a user, I can reset my password. | 2 | Medium | K.jayasurya  T.Ediga Brahma Koti  V.Teaja  P.Vignesh  C.Madhu Sudhakar |
| **Sprint-2** | **Book Browsing & Filtering** | USN-3 | As a user, I can search and filter books by genre/author. | 3 | High | K.jayasurya  T.Ediga Brahma Koti  V.Teaja  P.Vignesh  C.Madhu Sudhakar  C.Madhu Sudhakar |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | USN-4 | As a user, I can view detailed information of each book. | 2 | Medium | K.Poojitha,  T.Sureka,  K.Parimala |
| **Sprint-3** | **Cart & Purchase** | USN-5 | As a user, I can add books to the cart and proceed to checkout. | 3 | High | K.Poojitha,  T.Sureka,  K.Parimala |
|  |  | USN-6 | As a user, I can securely complete payment. | 3 | High | K.Poojitha,  T.Sureka,  K.Parimala |
| **Sprint-4** | **Order History & Reviews** | USN-7 | As a user, I can view my order history and track orders. | 2 | Medium | K.jayasurya  T.ediga Brahma Koti  V.Teaja  P.Vignesh  C.Madhu Sudhakar |
|  |  | USN-8 | As a user, I can leave reviews for purchased books. | 2 | Medium | K.jayasurya  T.ediga Brahma Koti  V.Teaja  P.Vignesh  C.Madhu Sudhakar |
|  |  | USN-9 | As a user, I can change my password. | 2 | Medium | K.jayasurya  T.ediga Brahma Koti  V.Teaja  P.Vignesh  C.Madhu Sudhakar |

**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** | 20 | 6 Days | 15 June 2025 | 20 June 2025 | 20 | 20 June 2025 |
| **Sprint-2** | 20 | 6 Days | 17 June 2025 | 22 June 2025 | 20 | 22 June 2025 |
| **Sprint-3** | 20 | 6 Days | 19 June 2025 | 24 June 2025 | 20 | 24 June 2025 |
| **Sprint-4** | 20 | 6 Days | 21 June 2025 | 26 June  2025 | 20 | 26 June  2025 |

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

**Project Overview:**

**Project Name:** BookNest  
**Project Description:** A user-friendly platform enabling users to explore, search, and purchase books online with ease. Features include book search, filtering, wishlist, cart, secure checkout, order history, and admin/seller dashboards.  
**Project Version:** v1.0.0  
**Testing Period:** 2025-06-15 to 2025-06-28  
**Testing Scope:**

* User registration and login
* Book search and filter by category, author, etc.
* Add to cart and wishlist
* Checkout and payment process
* Order confirmation and history
* Seller dashboard for managing listings
* Admin dashboard for managing users and content
* Notifications (email/SMS) for order and delivery
* Review and rating system

#### **Requirements to be Tested:**

* As a user, I want to securely register, login, and explore books.
* As a user, I want to search books and manage my wishlist and cart.
* As a user, I want a smooth and secure checkout and order tracking experience.
* As a seller, I want to add and manage my book listings.
* As an admin, I want to oversee platform users and content.

**Testing Environment:**

**URL:** https://booknest.example.com

**Credentials:**

**User:** test.user@example.com / user123

* **Seller:** test.seller@example.com / seller123
* **Admin:** admin@docspot.com / admin@12345

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Actual Result / Pass-Fail** |
| TC-001 | User Registration | 1. Visit site 2. Click "Sign Up" 3. Fill and submit form | Account created and redirected to dashboard | [Pass/Fail]: Pass |
| TC-002 | Book Search | 1. Enter book title/author 2. Click search | Books matching the criteria displayed | [Pass/Fail]: Pass |
| TC-003 | Add to Cart & Checkout | 1. Add book to cart 2. Proceed to checkout 3. Complete payment | Order placed successfully and confirmation shown | [Pass/Fail]: Pass |

**Bug Tracking:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Bug Description** | **Steps to reproduce** | **Severity** | **Status** | **Additional feedback** |
| BG-001 | Wishlist not saving items | 1. Add book to wish list 2. Refresh page | High | Open | Wishlist clears after reload |

**Sign-off:**

Tester Name: K.Parimala

Date: 28-06-2025

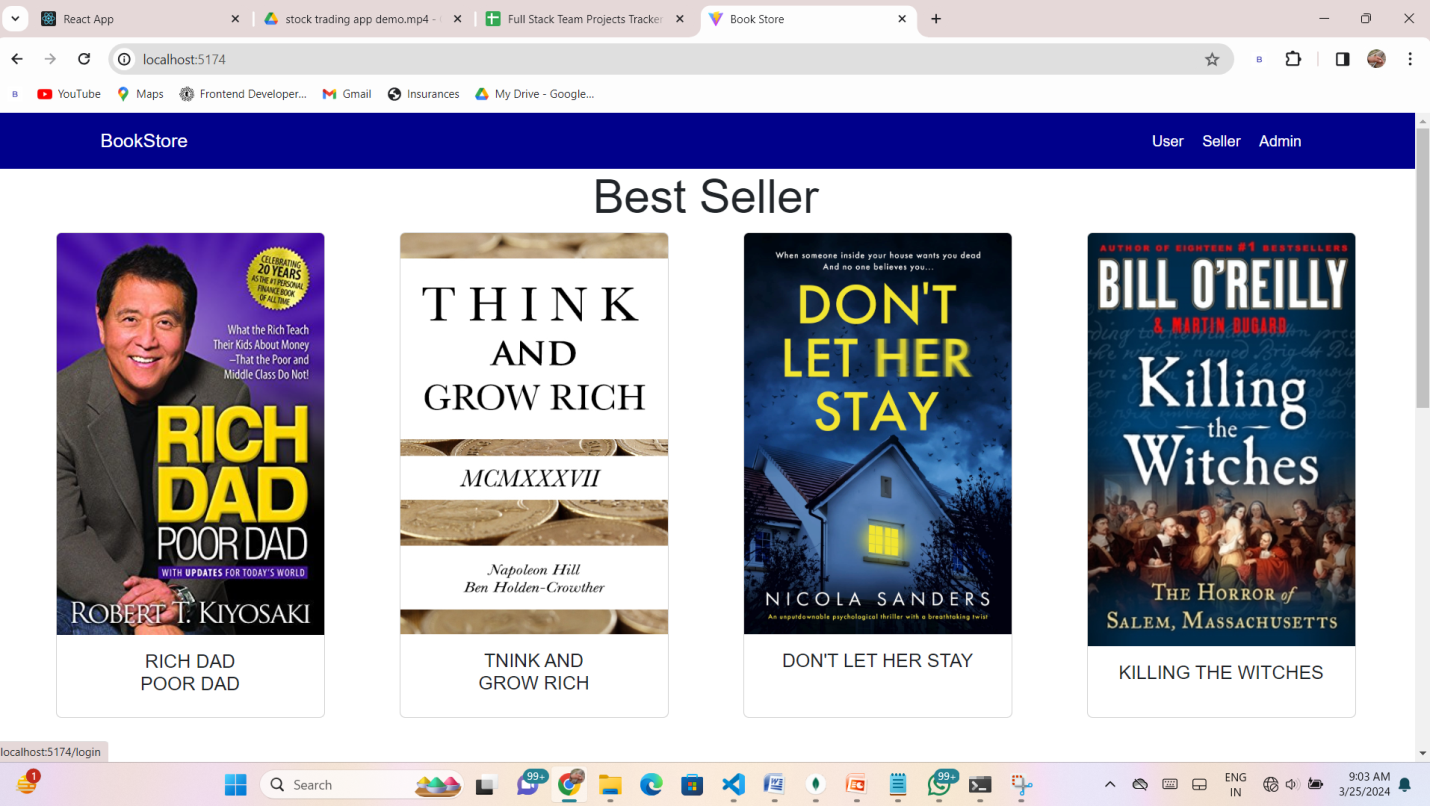
Signature: K.Parimala

**Notes:**

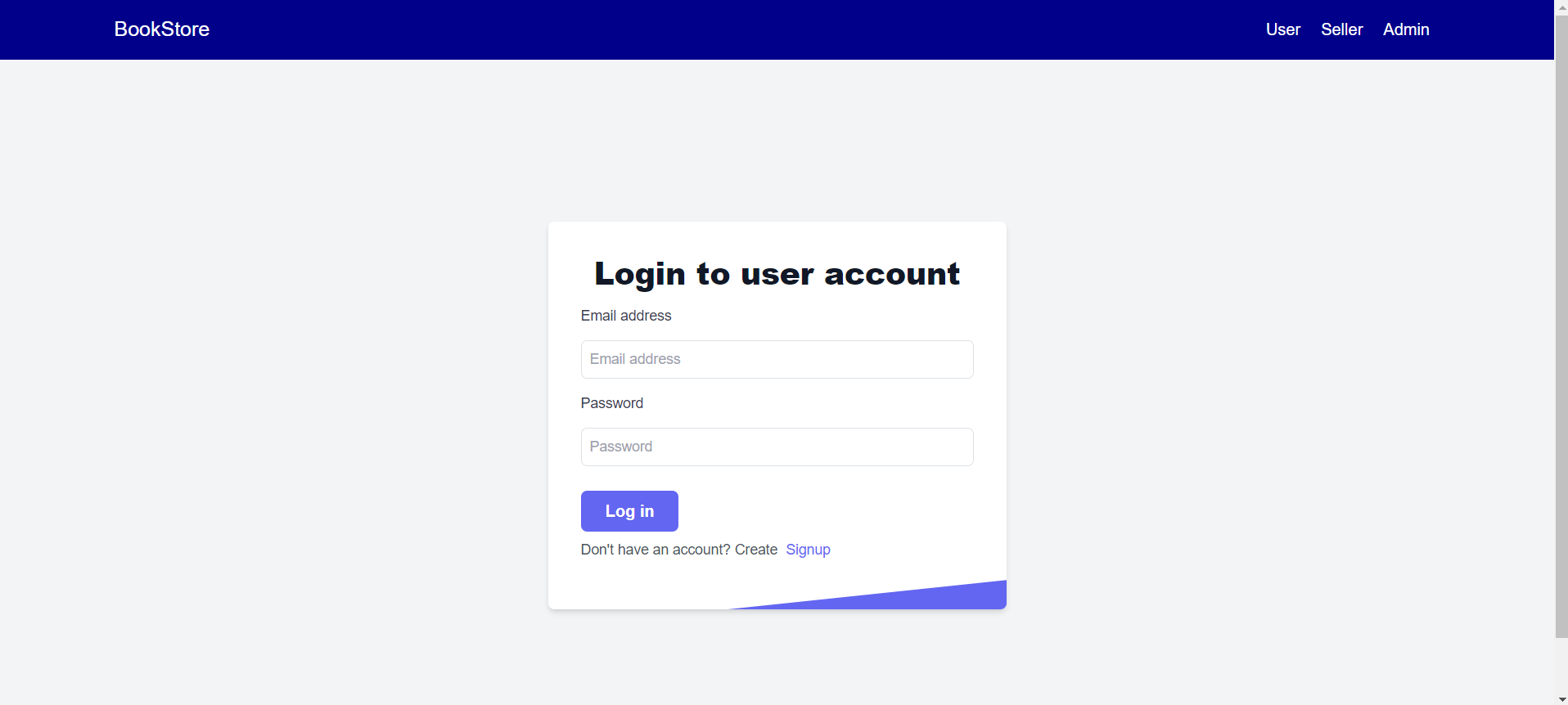
* Test across **multiple devices and browsers**.
* Cover **positive and negative** cases (e.g., invalid input, empty search, etc.).
* Track all bugs with reproduction steps and severity.
* Sign-off is required from the **project manager** and **product owner** before release.

**7. RESULTS**

**7.1 Output Screenshots**



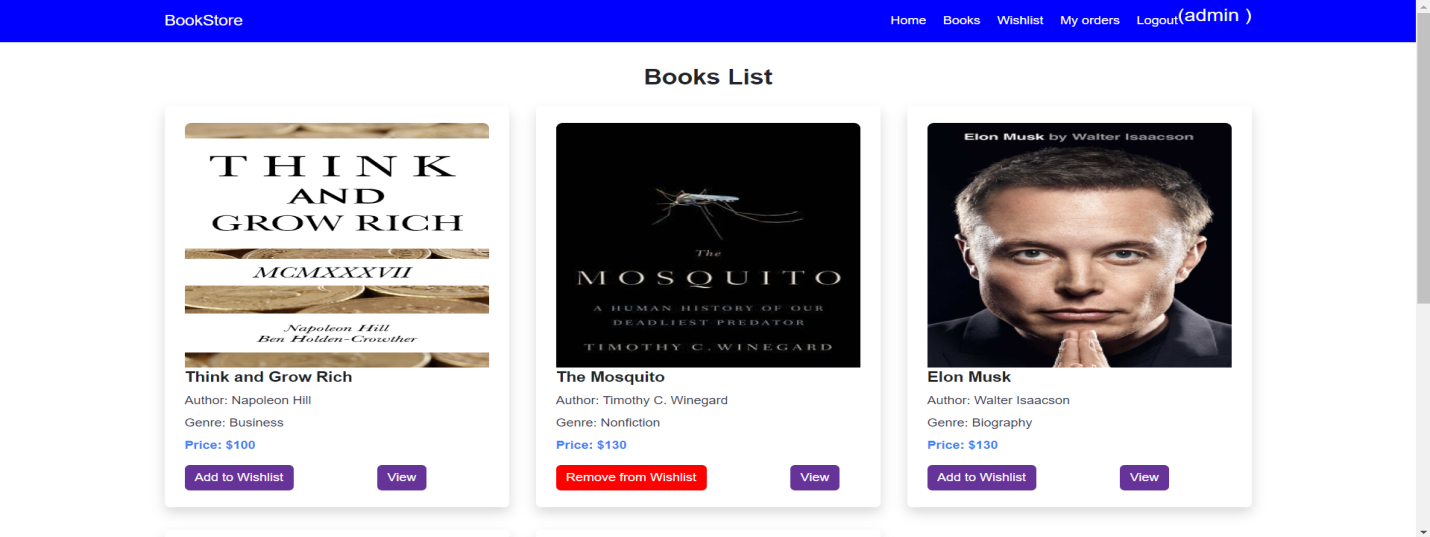
**Login Page:**



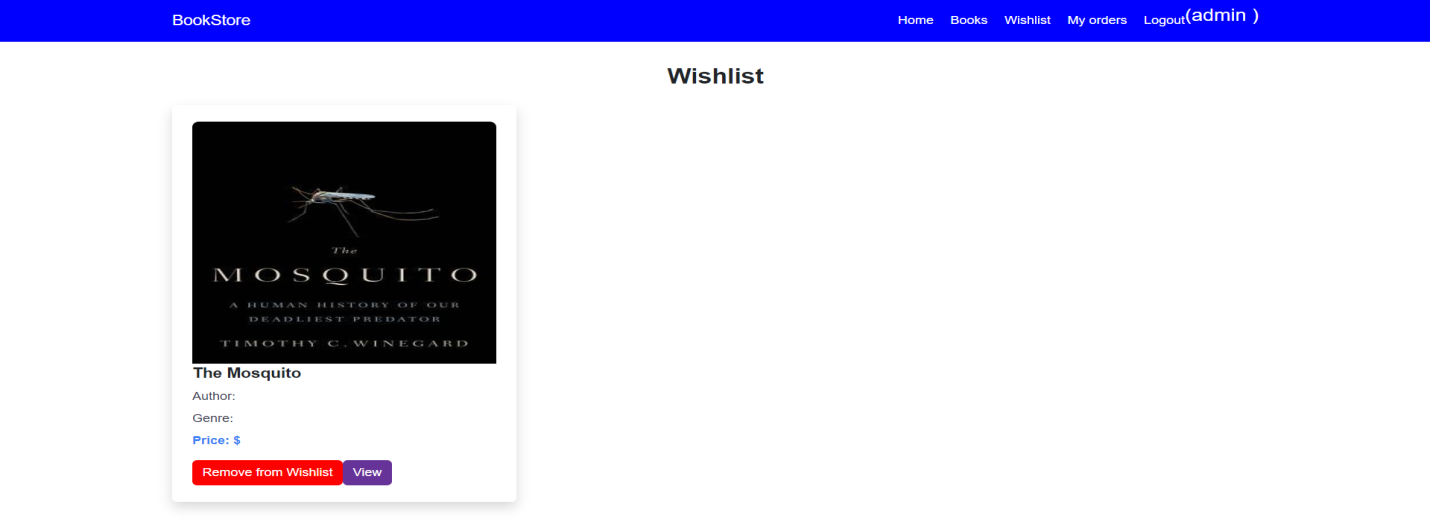
**Home Page:**



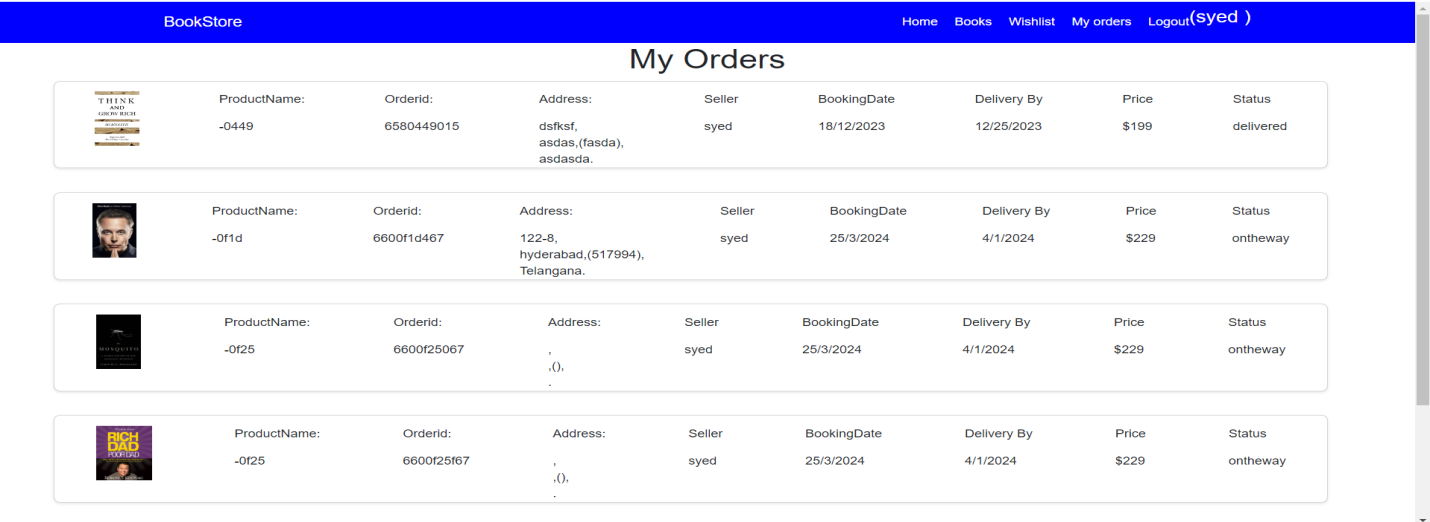
**Books Page:**



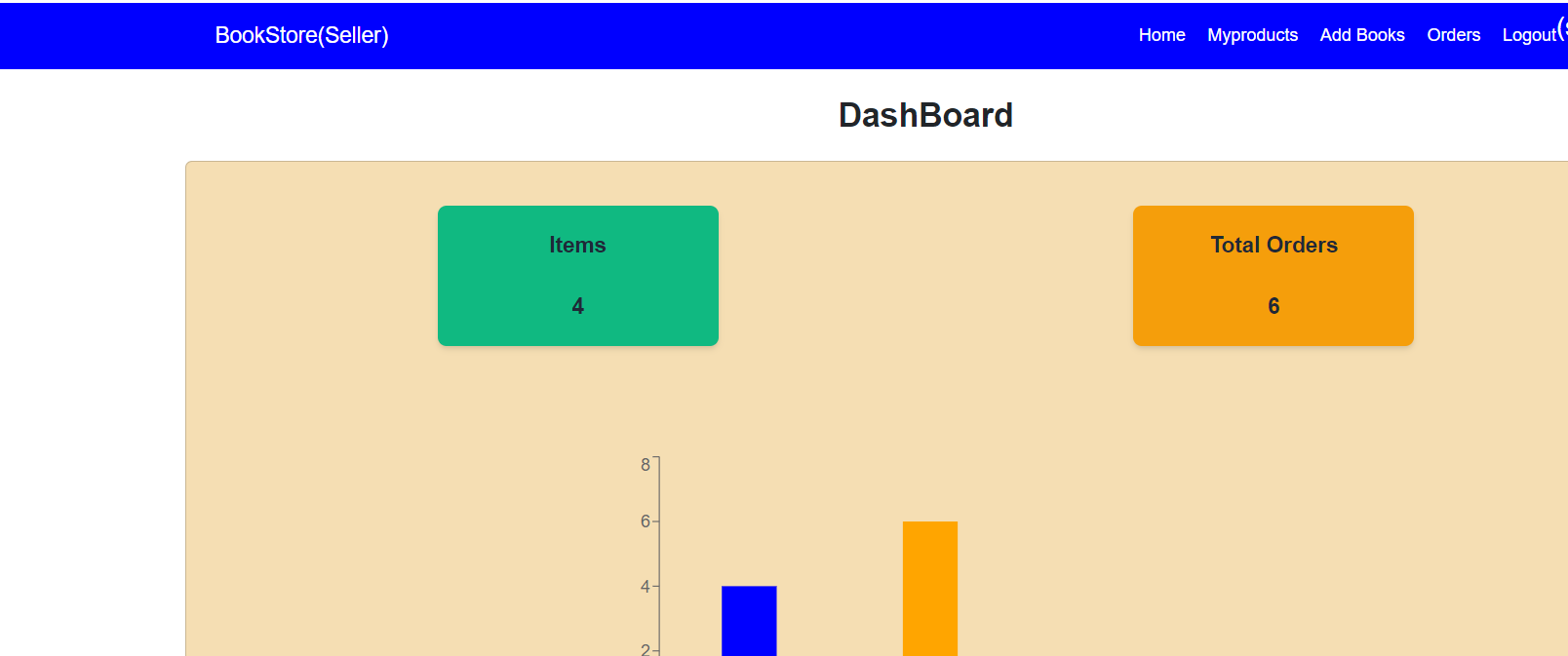
**Wishlist Page:**



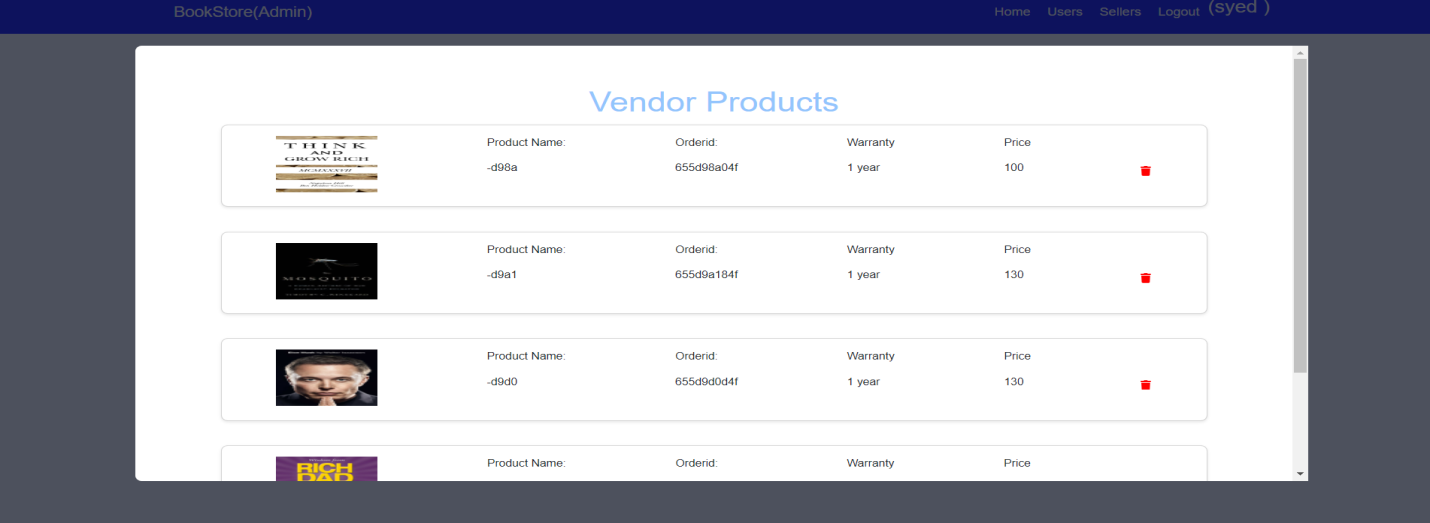
**My Bookings Page:**



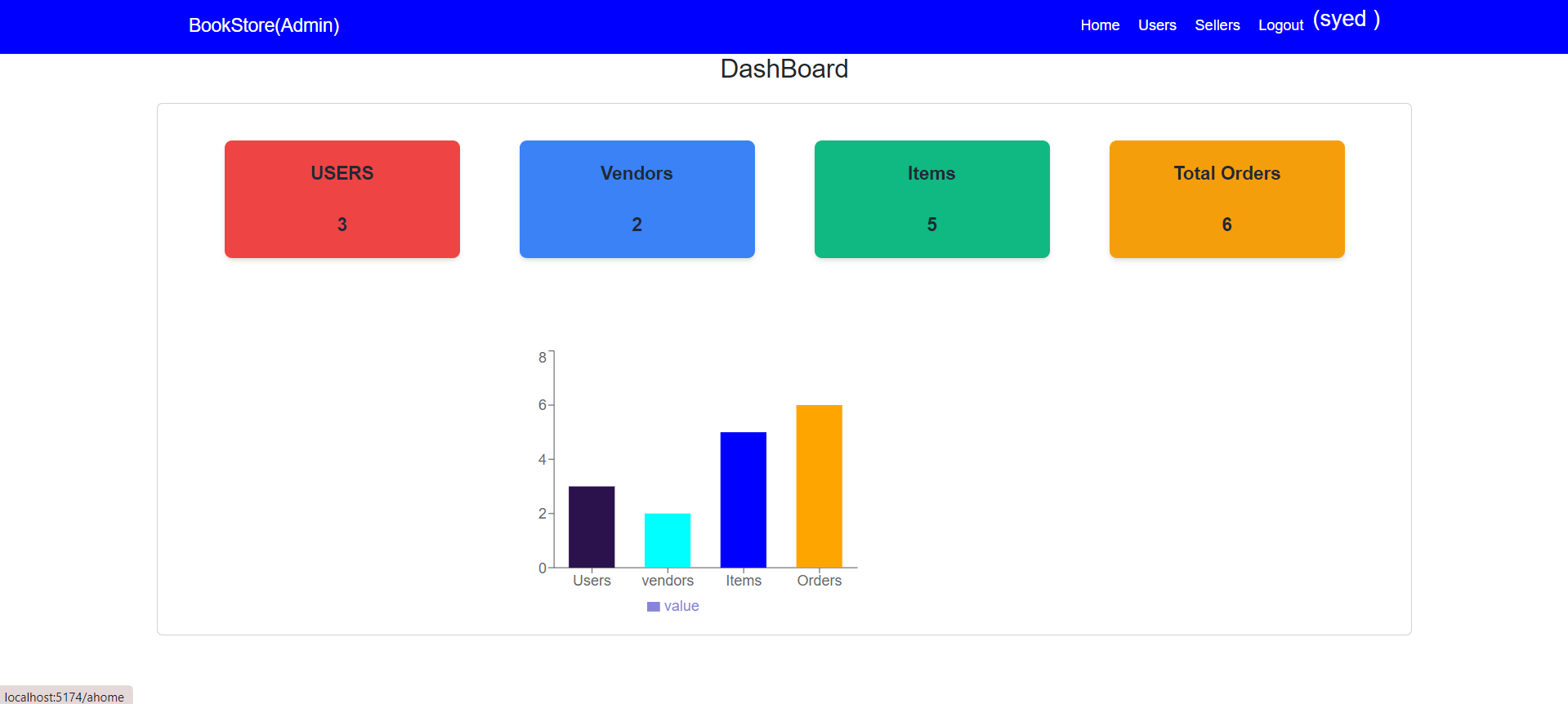
**Seller Dashboard:**



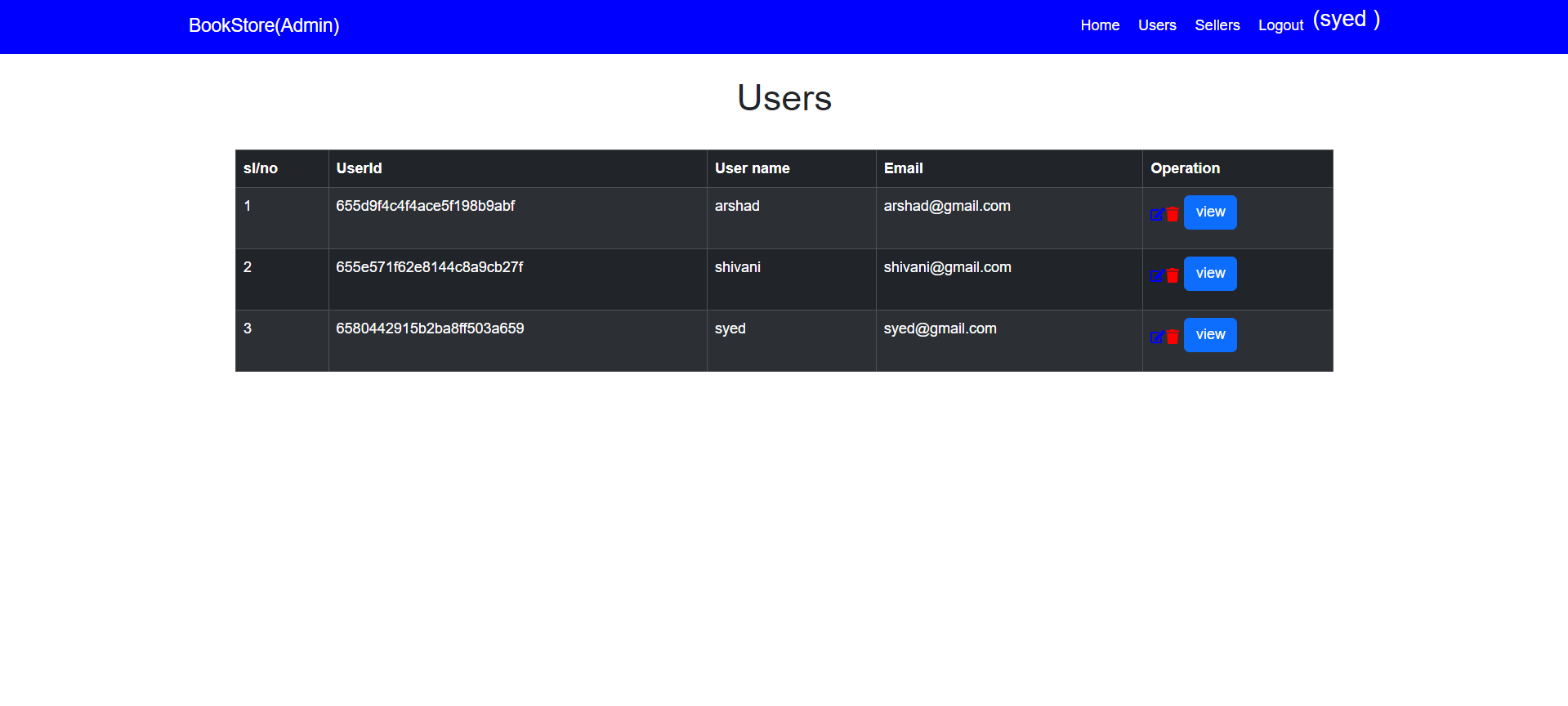
**Seller Items:**



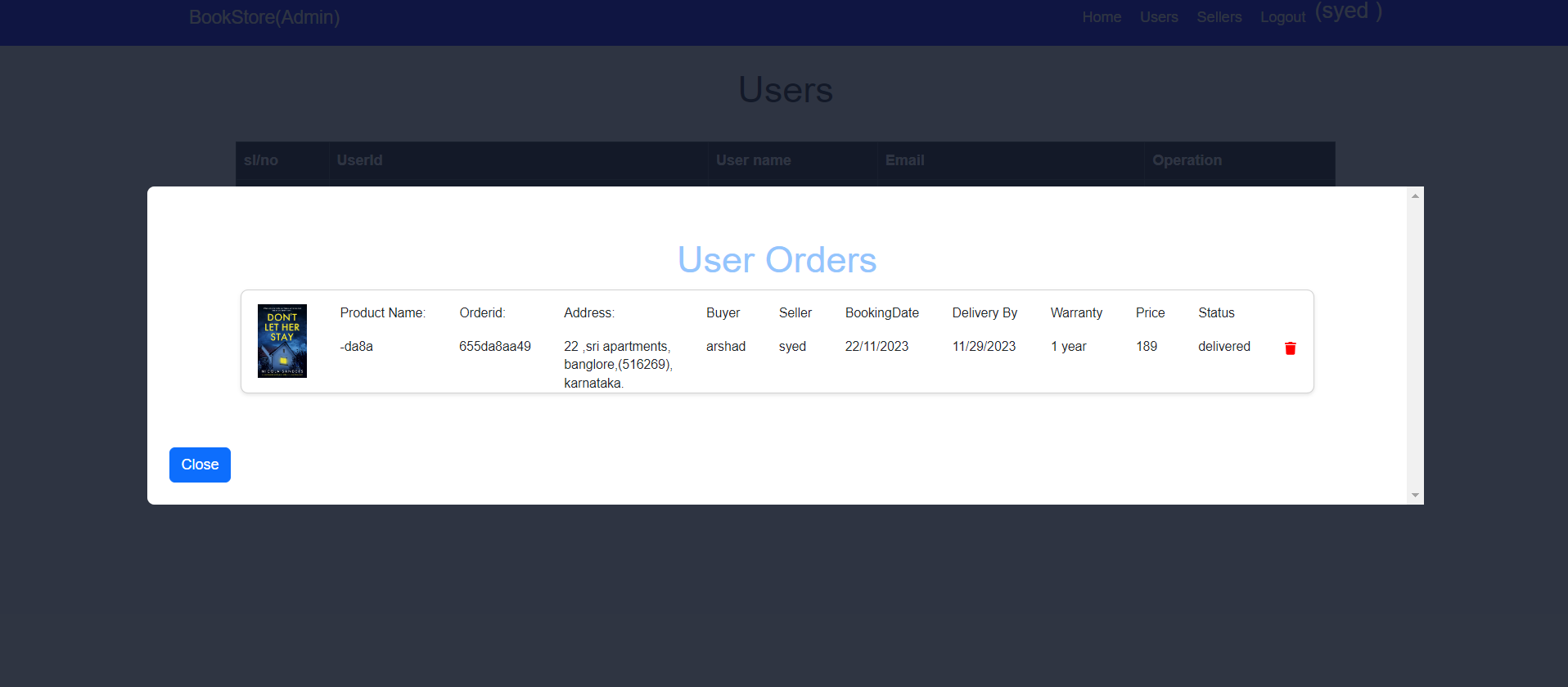
**Admin Dashboard:**



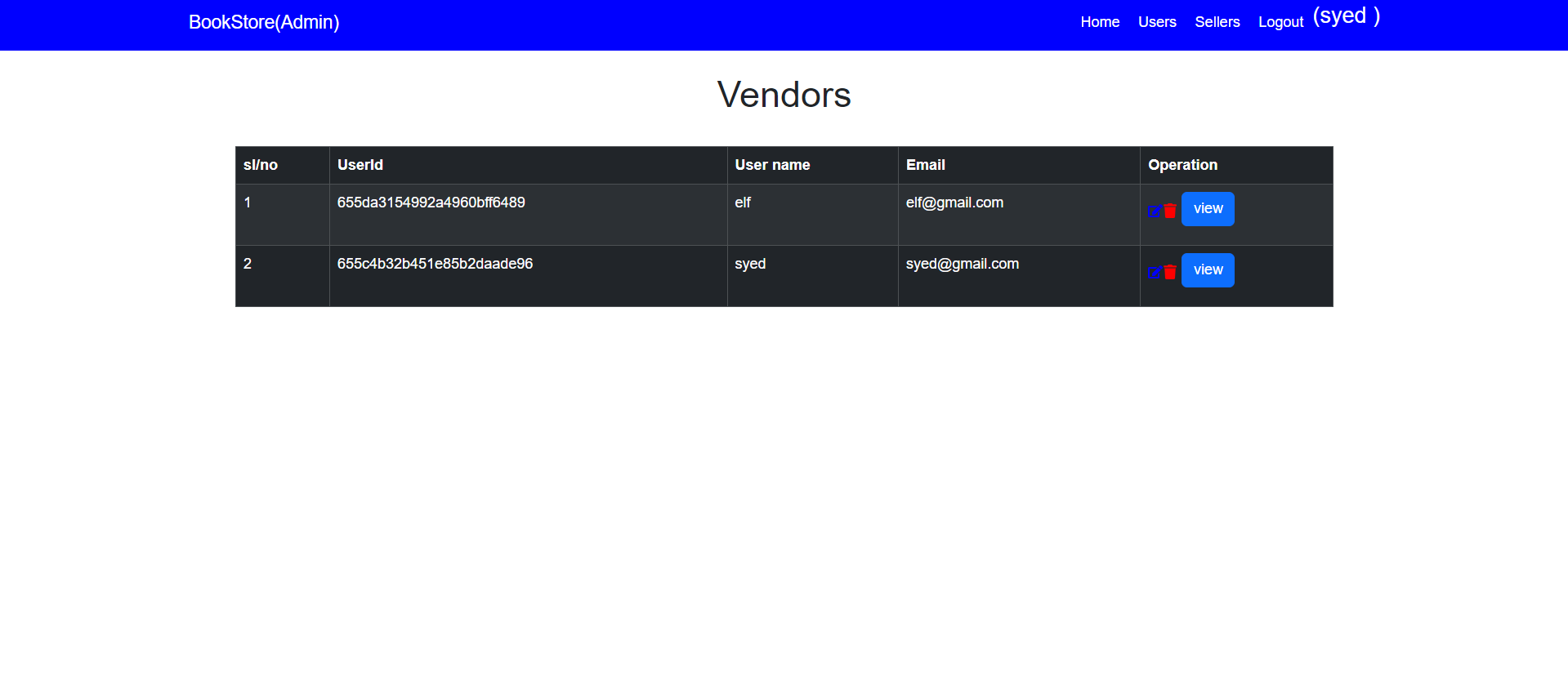
**Users Page:**



**User Orders:**



**Sellers Page:**

App demo link: [Demo](https://github.com/poojitha-kanipakam/Book-Nest-Where-Stories-Nestle-Team-ID-LTVIP2025TMID46365-/tree/main/Video%20Demo)

**8. ADVANTAGES & DISADVANTAGES**

**Advantages of Book Nest Project**

* **Modern Tech Stack**: Built with MERN (MongoDB, Express.js, React, Node.js), allowing a consistent JavaScript environment throughout.
* **Responsive UI**: React ensures fast, dynamic interfaces across devices.
* **Scalable Architecture**: MongoDB handles large datasets, and Node.js manages high concurrency efficiently.
* **Real-Time Operations**: Easily integrates real-time features like order tracking or live chat using tools like Socket.io.
* **Modular Design**: Separates concerns between frontend, backend, and database, simplifying maintenance.
* **Cross-Platform Support**: Works seamlessly across desktop and mobile browsers.
* **Extensible**: APIs can be expanded or connected with external services like payment gateways and recommendation engines.
* **Role-Based Access**: Clearly defined user, seller, and admin roles improve security and management.

**Disadvantages of Book Nest Project**

* **Steep Learning Curve**: Requires knowledge of four major technologies (React, Node.js, Express.js, MongoDB).
* **NoSQL Limitations**: MongoDB may not be ideal for complex relational data like multi-table joins.
* **Client-Side Performance**: Heavy frontend logic can slow down performance on low-end devices.
* **Security Management**: JWT, authentication, and user data must be handled carefully to avoid vulnerabilities.
* **SEO Limitations**: React single-page applications may struggle with search engine optimization without server-side rendering.
* **State Management Overhead**: Managing global state (e.g., with Redux) can become complex as the app grows.

**9. CONCLUSION**

The **Book Nest** project demonstrates the practical implementation of full-stack development using the powerful MERN stack — MongoDB, Express.js, React, and Node.js. It successfully bridges the gap between traditional bookstores and modern digital solutions, offering users a seamless experience for browsing, purchasing, and managing books online.

From intuitive user interfaces to efficient backend processing and robust database integration, Book Nest showcases a scalable, modular, and responsive application architecture. Through features like user authentication, real-time order management, admin control panels, and seller dashboards, the platform is capable of serving a wide range of stakeholders effectively.

While there are some challenges such as performance on low-end devices and the complexity of managing state and security, the overall design and execution of the project align with modern development standards. Book Nest not only solves a real-world problem but also serves as a strong foundation for future enhancements, such as advanced analytics, personalized recommendations, and integration with global marketplaces.

This project is a testament to how full-stack development can deliver complete, user-centric digital solutions in today’s rapidly evolving tech landscape.

**10. FUTURE SCOPE**

The **Book Nest** project has strong potential for expansion and enhancement. Below are some possible future developments:

**1. Advanced Search and Filters**

* Implement elastic search or fuzzy search capabilities.
* Add filters like price range, book condition (new/used), language, and publication year.

**2. Personalized Recommendations**

* Use machine learning models to recommend books based on user behaviour, purchase history, and ratings.

**3. Mobile App Integration**

* Develop native Android/iOS apps using React Native or Flutter for broader accessibility.

**4. Payment Gateway Integration**

* Integrate secure online payment gateways (e.g., Razor pays, Stripe, PayPal) for seamless transactions.

**5. Real-Time Order Tracking**

* Integrate with logistics APIs to provide users with live tracking of their book shipments.

**6. Wishlist & Favourites**

* Let users create wish lists, bookmark books, and receive alerts when those items are on sale or back in stock.

**7. Community and Reviews**

* Introduce a community forum, discussion threads, and verified book reviews.
* Enable social sharing of reading lists.

**8. Enhanced Security**

* Implement Two-Factor Authentication (2FA).
* Add CAPTCHA and stronger encryption for sensitive data.

**9. Analytics Dashboard**

* Build dashboards for sellers and admins to track sales, traffic, top-rated books, and user engagement.

**10. Multi-Language & Multi-Currency Support**

* Localize the app to support different languages and currencies for international expansion.

**11. APPENDIX**

**Source Code**

[**Code**](https://github.com/poojitha-kanipakam/Book-Nest-Where-Stories-Nestle-Team-ID-LTVIP2025TMID46365-/tree/main/Code)

**GitHub & Project Demo Link**

[**GitHub Link**](https://github.com/poojitha-kanipakam/Book-Nest-Where-Stories-Nestle-Team-ID-LTVIP2025TMID46365-)

[**Project Demo Link**](https://drive.google.com/drive/folders/1CFjcssACyAQ8XuS9M19LT4xWgCOUDYca)

[**Google drive Link**](https://drive.google.com/drive/folders/1ZxZGdcR8apN3IgB-An2nBTomhK6u_9VK?usp=drive_link)