

Project Report (Phase Documentation) - Book Store

Team Id - SWTID1743347281

Team Members

Komal Tripathi

Ayushi Sharma

Shreya Sharma

Srishti Jitpure

1. INTRODUCTION

1.1) Project Overview

The BookStore is a fully-featured online bookstore platform developed using the MERN stack. It allows users to browse, search, and purchase books while providing administrators with the ability to manage inventory efficiently.

1.2) Purpose

The primary purpose of the BookStore application is to offer a responsive, secure, and intuitive platform for online book sales with a clean and scalable codebase.

2. IDEATION PHASE

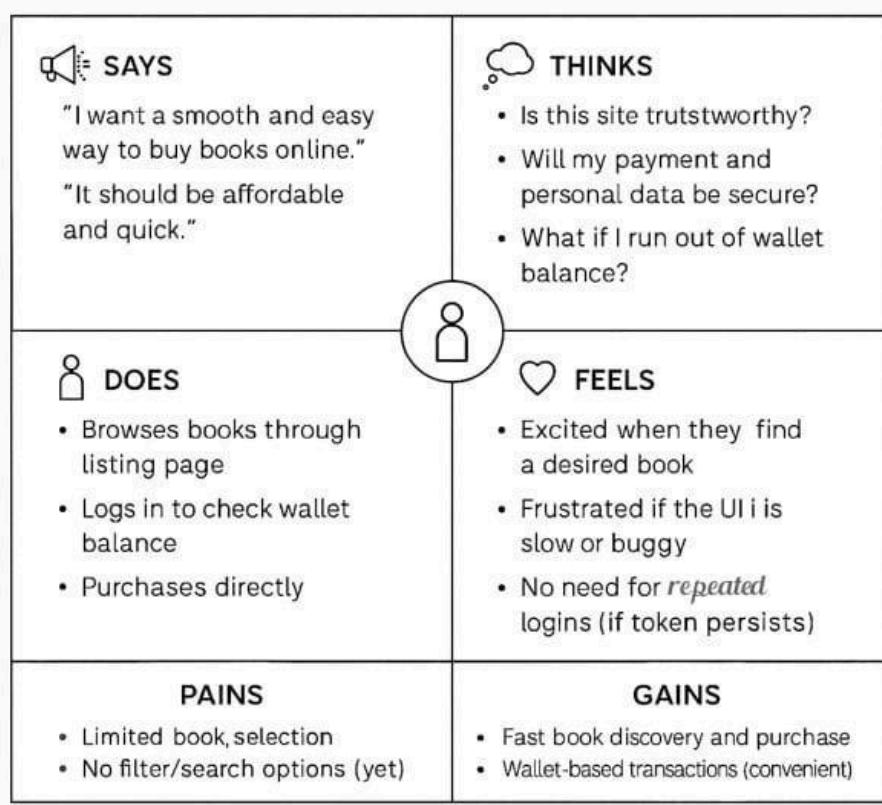
2.1) Problem Statement

Building an accessible and user-friendly platform for books with efficient inventory and user management capabilities.

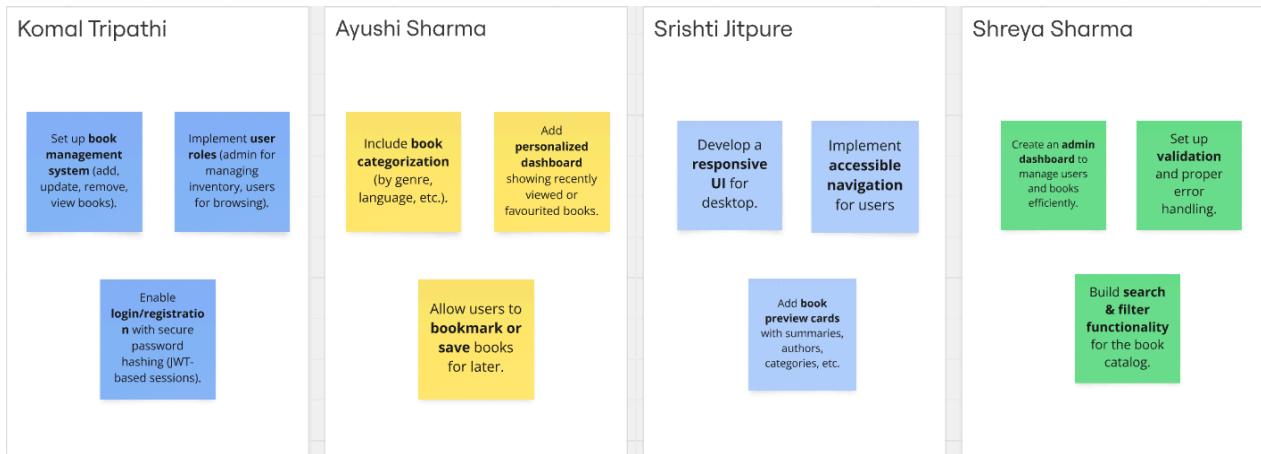
I am	I'm trying to	But	Because	Which makes me feel
a student who loves reading	buy books online quickly and easily	I can't find a reliable and fast platform	Most websites are either outdated, lack variety, or are hard to navigate	Frustrated and disappointed

I am	I'm trying to	But	Because	Which makes me feel
a student preparing for competitive exams	find the right set of reference books quickly	I waste a lot of time searching different websites	there is no personalized recommendation system for study materials	Anxious and stressed

2.2) Empathy Map Canvas



2.3) Brainstorming



3. REQUIREMENT ANALYSIS

3.1) Customer Satisfaction Map

Users visit the platform, register/login, browse/search for books, view details, and place orders. Admins manage book listings and can see the users and popular books.

Scenario: Making a bookstore website	ENTICE		ENTER		ENGAGE		EXIT		EXTEND	
STEPS	Wanting to read a book	Browsing books or wanting to create own library	Finding numerous available book options	Purchasing book from given wallet amount	Using wallet amount to purchase book	Or as admin adding the required books to DB	Using wallet amount to purchase book	Using wallet amount to purchase book	Using wallet amount to purchase book	Using wallet amount to purchase book
INTERACTIONS	Login/Logout or registration page	Selecting the book to be purchased	Payment Done Dialogue box	Checking the wallet balance dialogue box	Interacting with website for purchase	Interacting with website to check popular books	Successfully buying a book	Admin experiences the book trends	User has a purchased book	Admin is updated with trends to implement
GOALS & MOTIVATIONS	Help me in finding my book of choice	Help me in adding new books for user	Help in payment without much hassle	Help me login and logout easily	Help me feel good about buying the book	Help me putout books in demand	Help me leave satisfied with purchase	Admin leaves with proper user knowledge	User easily purchases book of choice	Admin has better understanding of user
POSITIVE MOMENTS	Finding different types of books	The process of finding user choices	Excitement about the purchase	Simple purchase implementation	User finds it easy to navigate books	Admin finds it easy to consider books to be kept	People leave website knowing they've made a successful purchase and will only be signed out once they click the button	User retains the book and can login to use it anytime		
NEGATIVE MOMENTS	Not finding book of choice	Book being more expensive than the money user has left in wallet		Admin unable to delete or modify users		Unsatisfied customer due to lack of book choices		Very less users might come back if they don't find book of choice		
AREAS OF OPPORTUNITY	Providing short summary of books	Show Genres of book below	Implemented a Search bar for ease of finding books	Implemented popular books for admin to see trends	How can we implement customer-admin communication to know if books might be available soon?			Whether it is possible for admin to handle user accounts to avoid piracy or any other activity from user?	Implementing a review system for customer	

3.2) Solution Requirement

Functional Requirements

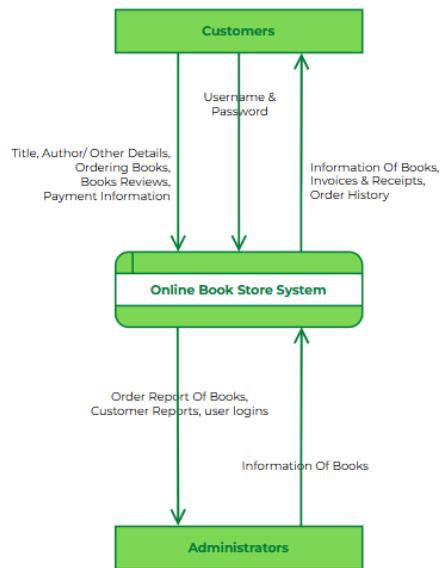
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Users should be able to sign up using email and password.
FR-3	Book Management	Add Book Edit Book Delete Book View Book List
FR-4	Authentication	Login Logout Password Encryption Token Generation
FR-5	Book Browsing	Search Book Filter by Genre/Price View Book Details
FR-6	Admin Dashboard	View Users View Book Stats Manage Inventory

Non-functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system should provide an intuitive and user-friendly interface.
NFR-2	Security	User data should be protected with encryption using dotenv jwt key authentication.
NFR-3	Reliability	The system should function correctly under expected usage conditions.

NFR-4	Performance	The application should load pages and respond to requests within 2 seconds.
NFR-5	Availability	The system should be available 99% of the time during working hours.
NFR-6	Scalability	The system should be able to handle increasing numbers of users and books.

3.3) Data Flow Diagram



3.4) Technology Stack

Table 1: Application Components

S.No	Component	Description	Technology
------	-----------	-------------	------------

1	User Interface	Web UI for user and admin interaction	HTML, Tailwind CSS, JavaScript, React.js
2	Application Logic-1	Backend logic for user registration, login, and book operations	Node.js, Express.js
3	Application Logic-2	Not applicable	—
4	Application Logic-3	Not applicable	—
5	Database	Stores user data, book data, and transactions	MongoDB (NoSQL)
6	Cloud Database	Cloud-hosted database solution	MongoDB Atlas
7	File Storage	Stores book images and related files	Local FileSystem
8	External API-1	Used for validating user emails or book metadata (if any)	Google Books API (optional)
9	External API-2	Not applicable	—

Table-2: Application Characteristics

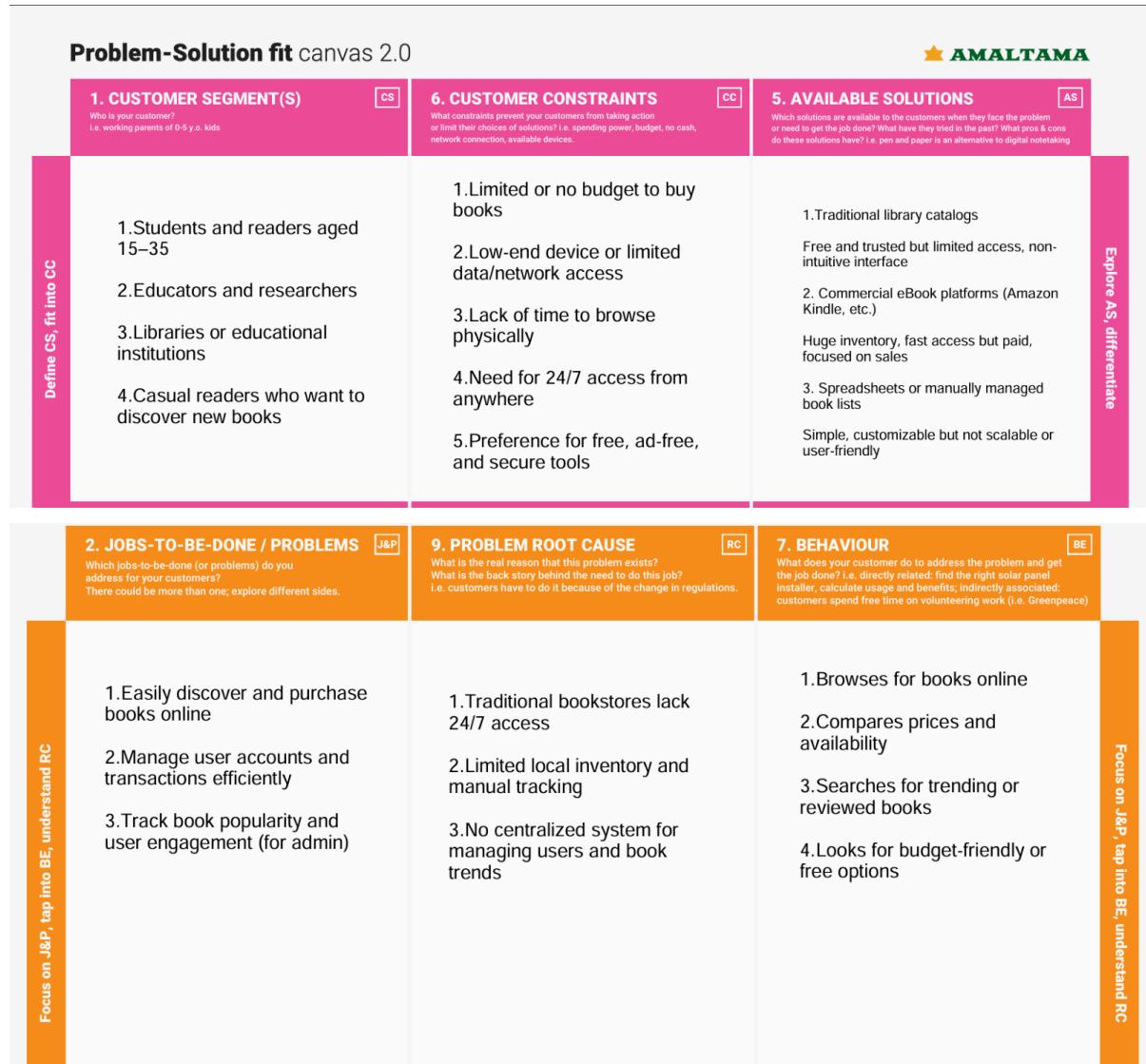
S.No	Characteristics	Description / Technology
1	Open-Source Frameworks	MERN Stack: React.js, Node.js, Express.js, MongoDB
2	Security Implementations	cors, dotenv, mongoose, nodemon, bcrypt password hashing, role-based access (, bcrypt, Helmet.js)
3	Scalable Architecture	3-tier architecture: UI, API server, database (MERN Stack, REST APIs)
4	Availability	Frontend on Vite (auto-scaling), backend on Node.js + Express.js

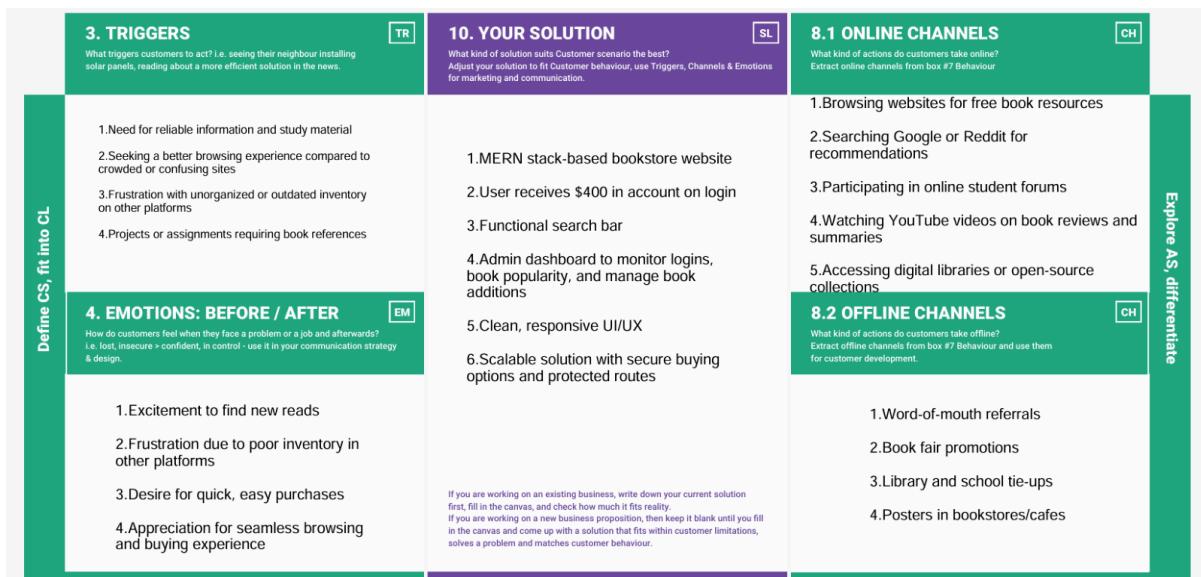
5	Performance	Optimized queries, search with debounce, pagination, MongoDB indexing, React lazy loading, CDN
---	-------------	--

4. PROJECT DESIGN

4.1) Problem Solution

Fit the project aligns with the need for a streamlined and accessible online bookstore, with scalable tech and features.





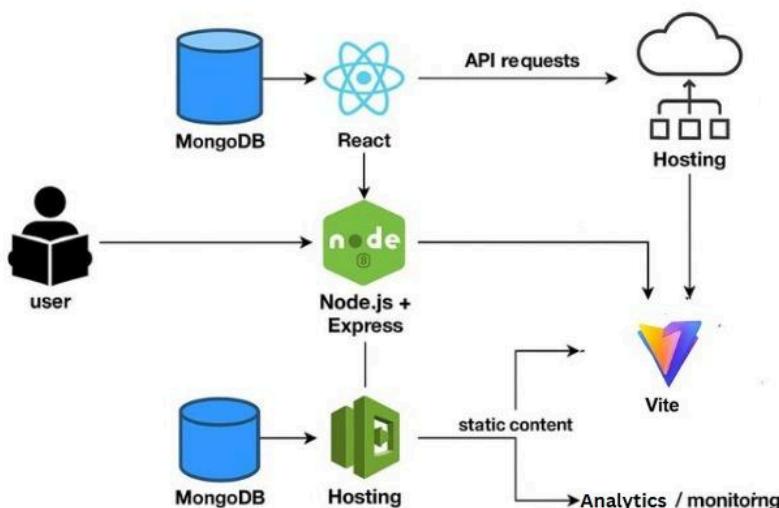
4.2) Proposed Solution

S. No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Lack of an accessible, streamlined, and secure platform for users to browse, manage, and purchase books online with user-specific and admin-level features.
2	Idea / Solution Description	MERN stack-based web application offering a full-fledged bookstore with book browsing, filtering, cart management, admin dashboard, secure login, and more.

3	Novelty / Uniqueness	Combines scalable technologies, modern UI/UX. Role-based access and real-time updates enhance the experience.
4	Social Impact / Customer Satisfaction	Enables easier access to educational and recreational reading. Promotes literacy by providing an intuitive, user-friendly platform for all demographics.
5	Scalability of the Solution	Designed for future growth with scalable architecture using cloud deployment (e.g., MongoDB Atlas, Vercel). Easily extendable to mobile apps and APIs.

4.3) Solution Architecture

Solution Architecture Diagram:



5. PROJECT PLANNING & SCHEDULING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Authentication	USN-1	As a user, I can register and log in to the application using email and password	2	High	Komal Tripathi
Sprint-1	Authentication	USN-2	As a user, I can log in securely and access features based on my role, such as browsing or managing books.	1	High	Shreya Sharma
Sprint-2	Authentication	USN-3	As an admin, I can manage user roles and access restrictions	2	Medium	Ayushi Sharma
Sprint-2	Book Management	USN-4	As an admin, I can add, update, or delete books in the inventory	3	High	Srishti Jitpure
Sprint-2	Book Browsing	USN-5	As a user, I can browse and search for books	2	High	Shreya Sharma
Sprint-3	Admin Dashboard	USN-6	As an admin, I can view a dashboard to track inventory and users	2	Medium	Komal Tripathi
Sprint-3	Admin	USN-7	As an admin, I can view the most purchased book	1	Low	Ayushi Sharma
Sprint-4	Testing	USN-9	As a developer, I can run backend and frontend test cases	2	Medium	Srishti Jitpure

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Project Overview:

- **Project Name:** Bookstore
- **Project Description:** A MERN-stack online bookstore platform providing a seamless user experience for book purchases with admin monitoring tools.
- **Project Version:** 1.0
- **Testing Period:** 2025-04-04 to 2025-04-06

Testing Scope:

- **Features to be Tested:**
 - User Registration & Login (Manual and Social)
 - Book Search & Display
 - Book Purchase using Wallet (\$400)
 - Admin Dashboard: Login Logs, Popular Book Stats
 - Add New Book (Admin)
- **User Stories to be Tested:**
 - USN-1 to USN-6 (User registration, login, search, and purchase features)
 - ADM-1 to ADM-3 (Admin functionalities)

Testing Scope:

Feature / Functionality	Description
User Authentication	Login, Registration, dotenv jwt key Authentication
Dashboard	Access after login, view balance, and available books
Search Functionality	Real-time book search
Book Purchase	Buying books using initial \$400 credit
Admin Features	View login logs, book popularity stats, add new books

Test Cases:

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	User Registration	1. Open Register Page 2. Enter Email, Password, Confirm Password 3. Click Submit	User gets registered and redirected to dashboard	User gets registered and redirected to dashboard	[✓]
TC-002	User Login	1. Open Login Page 2. Enter valid credentials 3. Click Login	Redirected to dashboard with token saved	Redirected to dashboard with token saved	[✓]
TC-003	Book Search	1. Navigate to Search Bar 2. Type Book Name	Matching books appear dynamically	Matching books appear dynamically	[✓]
TC-004	Purchase Book	1. Select Book 2. Click Buy 3. Confirm Purchase	\$ deducted, book added to profile	\$ deducted,	[✗]
TC-005	Admin Adds Book	1. Login as Admin 2. Go to Admin Panel 3. Fill book form, submit	New book is visible in catalog	New book is visible in catalog	[✓]

Bug Tracking:

Bug ID	Bug Description	Steps to Reproduce	Severity	Status	Additional Feedback
BG-001	Purchase button not responding	1. Select book 2. Click Buy 3. Nothing happens	High	In Progress	Check button Handler
BG-002	Admin stats not updating	1. Login as Admin 2. View stats after purchases	Low	Open	Might be cache issue

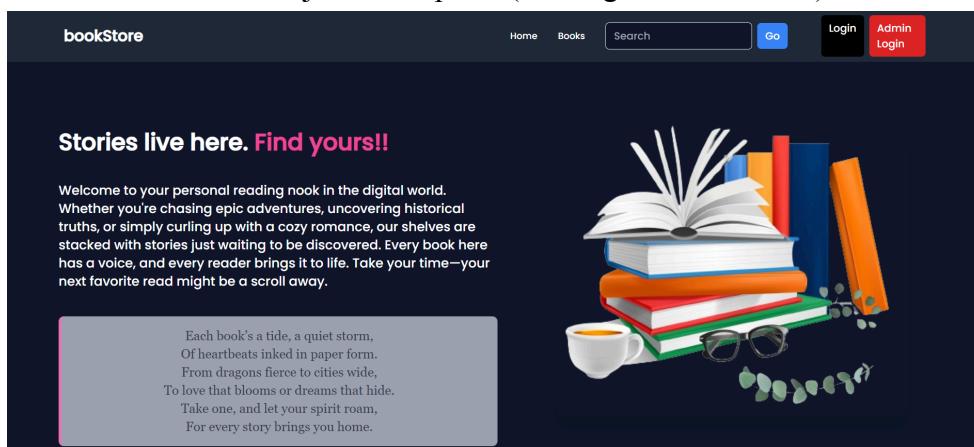
Sign-off:

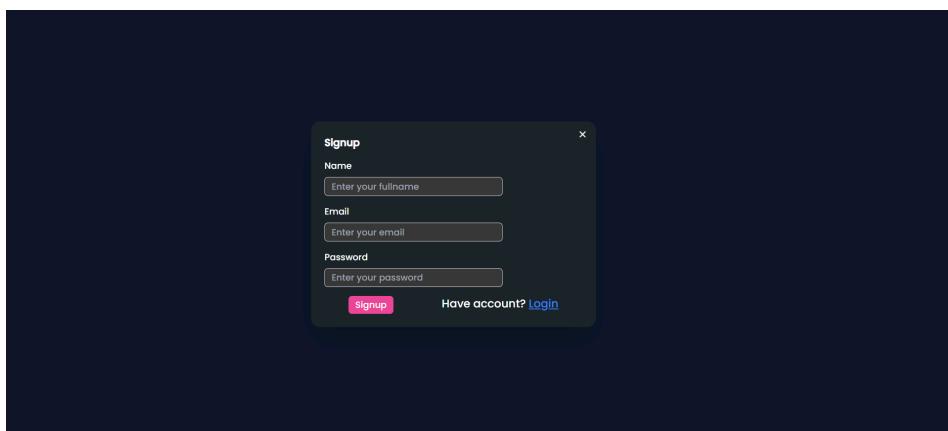
- Tester Name: Komal Tripathi
- Date: 6-Apr-2025

7. RESULTS

7.1 Output

- Frontend: Vite (locally hosted / deployed manually)
- Backend: Node.js with Express (running on local server)





bookStore

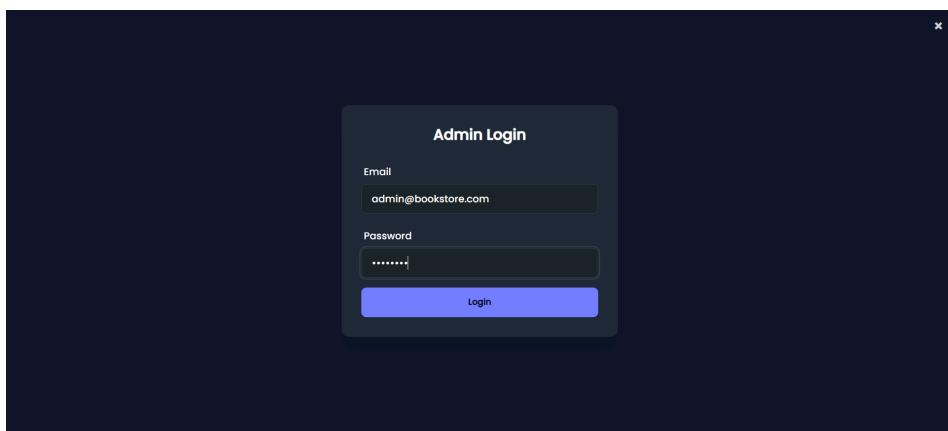
Home Books Search Go Admin Login

Stories live here. Find yours!™

Welcome to your personal reading nook in the clouds. Whether you're chasing epic adventures, uncovering hidden truths, or simply curling up with a cozy romance, our digital shelves are stacked with stories just waiting to be discovered. Every book has a voice, and every reader brings it to life. Your next favorite read might be a scroll away.

Each book's a tide, a quiet storm,
Of heartbeats inked in paper form.
From dragons fierce to cities wide,
To love that blooms or dreams that hide.
Take one, and let your spirit roam,
For every story brings you home.

Login Not registered? Signup

A decorative background image featuring a stack of colorful books, an open book showing text, a white coffee cup, and a pair of glasses resting on a surface.

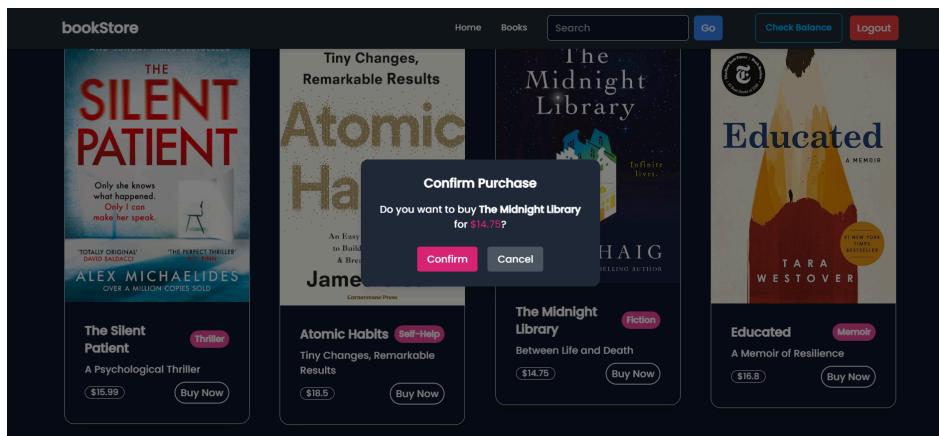
bookStore

Home Books Search Go Check Balance Logout

Four book covers are displayed in a row:

- 1984** by George Orwell (75th Anniversary Edition)
- To Kill a Mockingbird** by Harper Lee (Winner of the PULITZER PRIZE)
- THE SUBTLE ART OF NOT GIVING A F*CK** by MARK MANSON (A COUNTERINTUITIVE APPROACH TO LIVING A GOOD LIFE)
- The Great Gatsby** by F. SCOTT FITZGERALD (A Counterintuitive Approach to Living a Good Life)

Below each book cover are small buttons: "Buy Now", "Classic", "self-Help", and "Close".



Admin Dashboard

Hello, Admin [Logout](#)

[Books](#) [Add Book](#) [Users](#) [Book Popularity](#)

The Silent Patient Price: ₹15.99	Delete
Atomic Habits Price: ₹18.5	Delete
The Midnight Library Price: ₹14.75	Delete
Educated Price: ₹16.8	Delete
1984 Price: ₹12.99	Delete
To Kill a Mockingbird	Delete

Admin Dashboard

Hello, Admin [Logout](#)

[Books](#) [Add Book](#) [Users](#) [Book Popularity](#)

Ayush - ayushi@gmail.com Wallet: ₹230,589,999,999,999
Shreya - shreya@gmail.com Wallet: ₹386.5
Srishti - srishti@gmail.com Wallet: ₹363.21

Admin Dashboard

Hello, Admin [Logout](#)

[Books](#) [Add Book](#) [Users](#) [Book Popularity](#)

The Great Gatsby
13.50
category
image
title

[Add Book](#)

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Modular codebase
- Admin controls
- Responsive design

Disadvantages:

- No pagination yet
- Admin user must be created manually
- No password reset or email verification

9. CONCLUSION

The BookStore project effectively delivers a foundation for a scalable and user-friendly platform using the MERN stack.

10. FUTURE SCOPE

- Order workflows and payments
- Book reviews and ratings
- Advanced filtering/search
- Analytics for admins
- Internationalization support

11. APPENDIX Source Code:

Available at GitHub Dataset Link: N/A GitHub & Project Demo Link:
<https://github.com/KomalTripathi/SmartBridge>