

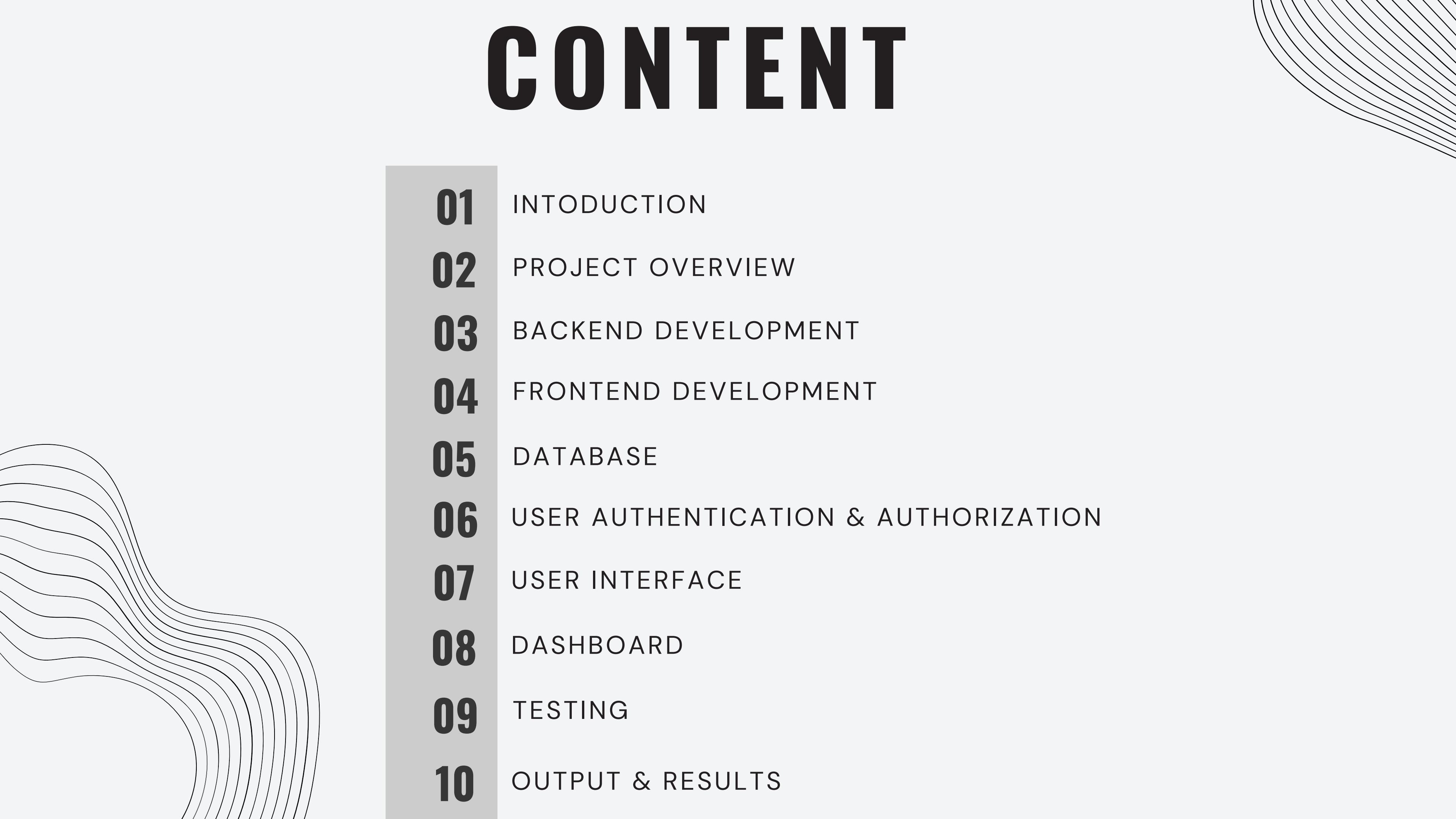
JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BOOK STORE USING MERN

SUPERVISED BY
Mrs.JEEVITHA D (M.E)
Assistant Professor-CSE

CONTENT

- 
- 01** INTRODUCTION
 - 02** PROJECT OVERVIEW
 - 03** BACKEND DEVELOPMENT
 - 04** FRONTEND DEVELOPMENT
 - 05** DATABASE
 - 06** USER AUTHENTICATION & AUTHORIZATION
 - 07** USER INTERFACE
 - 08** DASHBOARD
 - 09** TESTING
 - 10** OUTPUT & RESULTS

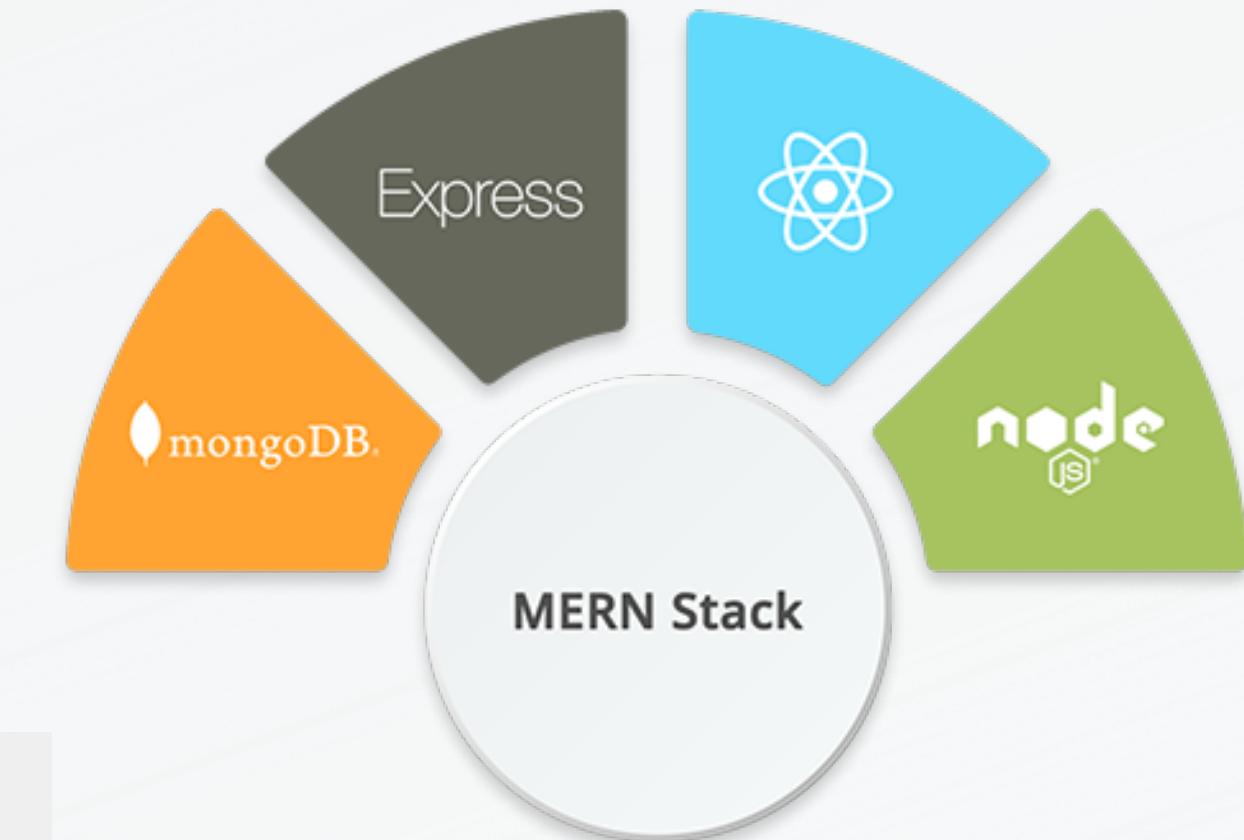
INTRODUCTION



- **What is MERN?** A powerful combination of technologies for full-stack web development.
- MongoDB: A flexible, NoSQL database.
- Express.js: A minimalist web framework for Node.js.
- React.js: A JavaScript library for building user interfaces.
- Node.js: A JavaScript runtime environment.



- **Why MERN?** Efficient development workflow.
- Strong community and extensive resources.
- Scalability and performance.
- Cross-platform compatibility.



PROJECT OVERVIEW

CORE FUNCTIONALITIES

- User registration and authentication
- Book browsing and search
- Product details and reviews
- Shopping cart and checkout
- Order tracking and history
- Admin dashboard for managing products, orders, and users

TECHNOLOGY STACK

- **Frontend:** React.js, , Material-UI, Axios
- **Backend:** Node.js, Express.js.
- **Database:** MongoDB.

BACKEND

```
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
require('dotenv').config();

const app = express();
const PORT = process.env.PORT || 5000;

// Middleware
app.use(cors());
app.use(express.json());

// MongoDB Connection
mongoose.connect(process.env.MONGO_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
.then(() => console.log('MongoDB connected'))
.catch((err) => console.log('MongoDB connection error: ', err));

// Basic Route
app.get('/', (req, res) => {
  res.send('Book Store API is running');
});

// Start the server
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

- Express.js is a web application framework for Node.js. Install Express.js to handle server-side routing, middleware, and API development.
- Installation: Open your command prompt or terminal and run the following command: **npm install express**

React js: React is a JavaScript library for building client-side applications.

And Creating Single Page Web-Appliaction

Getting Started

Create React App is an officially supported way to create single-page React applications. It offers a modern build setup with no configuration.

FRONTEND

HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations.

Front-end Library: Utilize React to build the user-facing part of the application, including products listings, booking forms, and user interfaces for the admin dashboard.

Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

- Git: Download and installation instructions can be found at: <https://git-scm.com/downloads>

```
const fetchBooks = async () => {
  const response = await axios.get('http://localhost:5000/api/books');
  console.log(response.data);
```

DATABASE

Key Roles of MongoDB in a Book Store:

1. Storing Book Information:

- Core details: Title, author, ISBN, publication year, genre, price, and description.
- Additional metadata: Publisher, language, dimensions, weight, and page count.
- Rich content: Reviews, ratings, and user comments.
- Images: Store image file paths or references to cloud storage.

2. Managing User Data:

- User profiles: Name, email, password (hashed), address, phone number, and preferences.
- Order history: Purchase details, shipping address, payment information, and order status.
- Wishlist: Books added to a user's wishlist.
- Cart: Items added to a user's shopping cart, including quantity and price.

3. Handling Inventory:

- Stock levels: Track the quantity of each book in inventory.
- Low-stock alerts: Trigger notifications when stock levels fall below a certain threshold.
- Warehouse management: Manage multiple warehouses and track inventory across locations.

USER INTERFACE



UI COMPONENTS

Home page
Book page
Authentication page
Book Details page
User profile page
Admin panel

RESPONSIVE DESIGN

- The UI is styled using CSS, Bootstrap, or Tailwind CSS for responsiveness.
- Collapsible Navbar for small screens.
 - Grid layouts adjust dynamically for different screen sizes.

STATE MANAGEMENT

- React Context API or Redux is used to manage the application state efficiently.
- Examples:
 - User authentication status is stored globally to display dynamic navbar links.
 - Shopping cart data is stored and updated in real-time.

ERROR HANDLING

- Custom error pages (e.g., 404 for invalid routes).
- User-friendly error messages for API failures (e.g., "Unable to fetch books").

USER AUTHENTICATION & AUTHORIZATION

- Authentication is an essential part of the MERN Book Store application. It ensures secure access to protected resources, such as adding books, placing orders, and viewing user-specific data. The application uses JWT (JSON Web Token) for secure, stateless authentication.
- **Authentication Features**
 - User Registration: Allows new users to sign up.
 - User Login: Verifies user credentials and issues a JWT token.
 - Token Verification: Validates the JWT token for accessing protected routes.
 - Role-Based Access Control (RBAC): Restricts certain actions (like adding or deleting books) to admin users only.
- **Workflow**
 - **Registration:**
 - A new user provides a username, email, and password.
 - The server hashes the password using bcrypt and stores it securely in the database. After successful registration, the user can log in.
 - **Login:**
 - The user submits their email and password.
 - The server validates the credentials against the database.
 - If valid, the server generates a JWT token and returns it to the user.
 - The user includes this token in the Authorization header of subsequent requests to access protected routes.
 - **Token Validation:**
 - Each protected route checks for a valid JWT token.
 - If the token is missing, invalid, or expired, access is denied.

DASHBOARD

Visualizations

- **Sales Over Time:** Line chart showing sales trends over a specific period (daily, weekly, monthly, yearly).
- **Top-Selling Products:** Bar chart highlighting the best-selling products.
- **Customer Demographics:** Pie chart or bar chart showing customer distribution by age, gender, or location.
- **Order Status Distribution:** Pie chart illustrating the distribution of orders by status (pending, processing, shipped, delivered).
- **Revenue by Category:** Bar chart showing revenue generated by different book categories.
- **Customer Segmentation:** Segmentation of customers based on purchase behavior (e.g., frequent buyers, occasional buyers).



TESTING

Testing ensures the reliability and functionality of the Book Store application. The testing process includes both manual and automated tests across various levels of the application.

- Purpose: Test individual components and functions in isolation.
- Tools: Jest and React Testing Library.

UNIT TESTING

- Purpose: Ensure that different parts of the application work together.
- Tools: Jest and Supertest.

INTEGRATION TESTING

- Purpose: Simulate user interactions to test the entire application flow.
- Tools: Cypress or Playwright.

E2E TESTING

FUTURE ENHANCEMENT

Integration of Payment Gateway: The application will integrate secure payment gateways like PayPal, Stripe, and Razorpay for seamless purchase experiences, supporting multiple payment methods like credit/debit cards, digital wallets, and net banking.

- Advanced Search and Recommendation Engine

The website will improve search functionality with natural language processing, advanced filtering, and a machine learning-powered recommendation engine, enhancing user engagement and personalizing the shopping experience.

- Mobile Application Development

A mobile app for Android and iOS will enhance the Book Store's accessibility, offering offline browsing, push notifications, and a user-friendly interface.

- User Reviews and Ratings

The Book Store will introduce a review and rating system, enabling users to provide feedback on their reading experiences, fostering an interactive community.

- Admin Dashboard for Inventory and User Management

The admin dashboard will enhance operational efficiency by allowing administrators to manage inventory, view analytics, and oversee user activity, providing valuable insights into user behavior and sales trends.

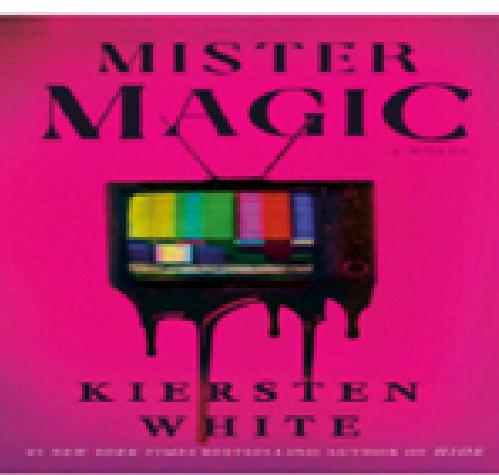
OUTPUT

localhost:5173/wishlist

BookStore

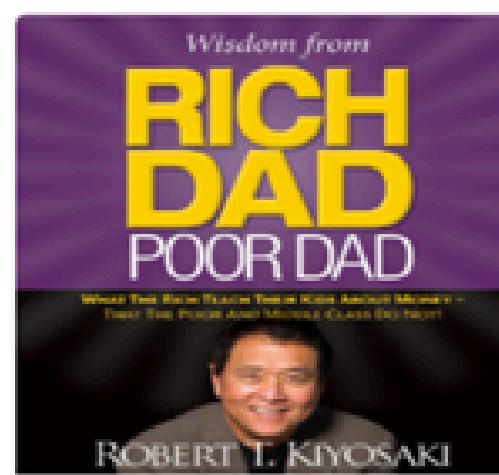
Home Books Wishlist My orders LogOut

Wishlist



Mister Magic
Author: Kiersten White
Genre: Fantasy
Price: \$

[Remove from Wishlist](#) [View](#)



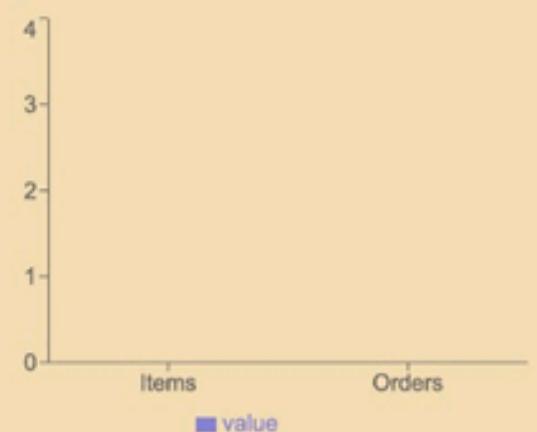
Rich Dad Poor Dad
Author: Robert T. Kiyosaki
Genre: Finance
Price: \$

[Remove from Wishlist](#) [View](#)

Dashboard

Items: 0

Total Orders: 0



BookStore(Admin)

Home Users Sellers Logout (umar)

Vendors

sl/no	UserId	User name	Email	Operation
1	67480b82561a8e54b92d8227	seller	seller@example.com	  View

STRATEGY N°1

STRATEGY N°2

Orders

	ProductName:	Orderid:	Customer Name	Address:	BookingDate	Delivery By	Warranty	Price	Status
		6738687e6b	Sanjai Customer	26, ewrer,(63565), ewrew.	16/11/2024	11/23/2024	1 year	429	ontheaway
	ProductName:	Orderid:	Customer Name	Address:	BookingDate	Delivery By	Warranty	Price	Status
		673868916b	Sanjai Customer	213, Neyveli,(600119), Tssmil.	16/11/2024	11/23/2024	1 year	654	ontheaway

RESULT

The "Book Store" platform is a comprehensive online solution for book enthusiasts, offering a seamless shopping experience. Built with a user-friendly front-end powered by React and a robust back-end utilizing Node.js, Express.js, and MongoDB, the platform allows users to browse through a vast collection of books, add items to their cart, and complete secure checkout.

While the current platform provides essential functionalities, there's potential for improvement in areas such as real-time inventory updates, personalized recommendations, and advanced search capabilities. Future enhancements, including integration with popular e-readers, social media sharing, and a loyalty program, will further elevate the user experience and drive customer engagement.

THANK YOU

