

Multi-Seller Bookstore App - Project Report

Title: Delhi Book Market.in App

Name: Joshua Deepu Mathew

Github: <https://github.com/Joshua633>

Email: josmindeepu23@gmail.com

Submission Date : 11 September 2025

1. Project Overview

The Multi-Seller Bookstore App is a comprehensive mobile application designed to facilitate a marketplace for books with multiple sellers. The application supports two distinct user roles: Buyers who can browse, search, and purchase books, and Sellers who can list books, manage inventory, and process orders. The system is built using a modern tech stack ensuring scalability, maintainability, and a seamless user experience.

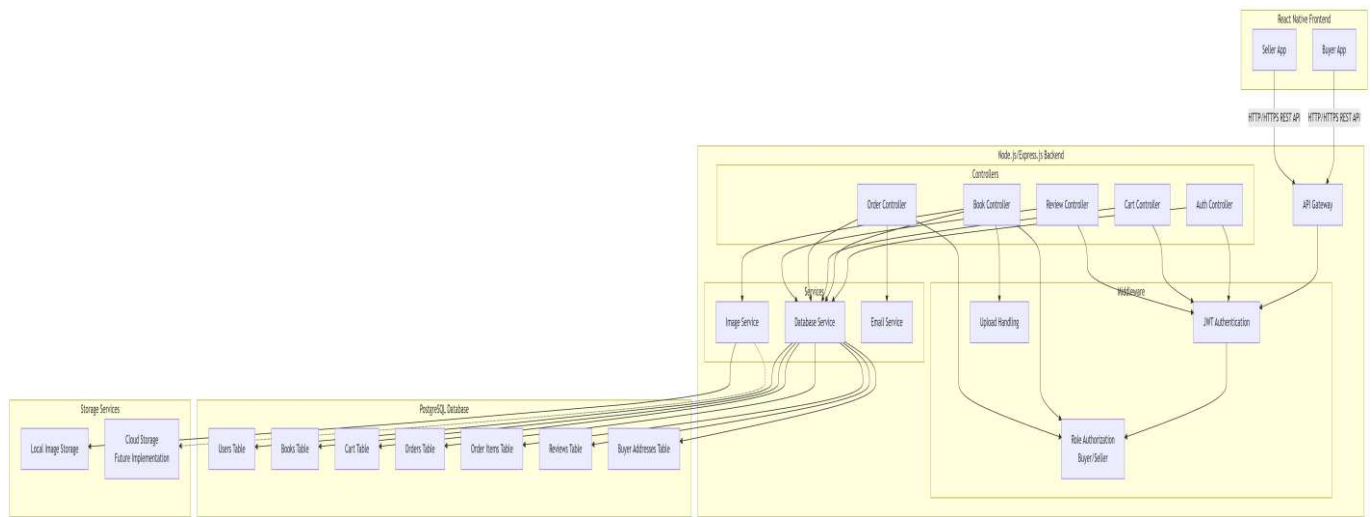
2. Technology Stack

- **Frontend:** React Native with Expo
- **Backend:** Node.js with Express.js
- **Database:** PostgreSQL
- **Authentication:** JWT (JSON Web Tokens)
- **Image Storage:** Multer for local storage
- **State Management:** React Context API

3. System Architecture

The application follows a client-server architecture with a clear separation between frontend and backend components. The React Native mobile app communicates with the Node.js/Express.js backend via RESTful APIs, while the PostgreSQL database handles data persistence.

3.1 Architecture Diagram



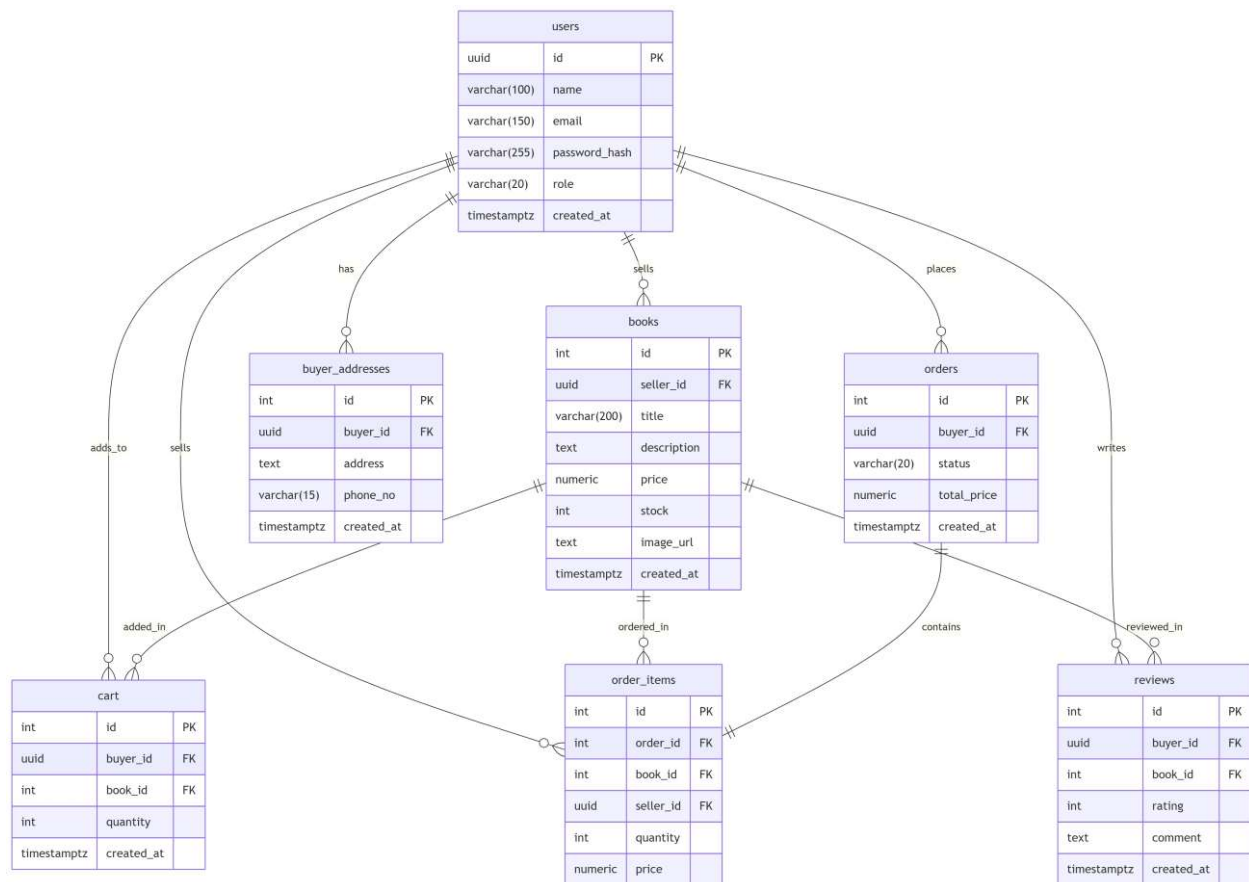
The architecture consists of:

- Frontend Layer (React Native)
- Backend Layer (Node.js/Express.js)
- Database Layer (PostgreSQL)
- Storage Layer (Local/Cloud)

4. Database Design

The application uses a well-structured PostgreSQL database with tables designed to support all required functionality.

4.1 Entity-Relationship Diagram



4.2 Database Tables

- **users**: Stores user information with role-based access
- **books**: Contains product listings with seller information
- **cart**: Manages shopping cart items
- **orders**: Stores order headers with status information
- **order_items**: Contains individual items within orders
- **reviews**: Manages product ratings and reviews
- **buyer_addresses**: Stores customer shipping information

5. Functional Requirements

5.1 Buyer Side Features

- Storefront feed displaying all books
- Product detail pages with add to cart functionality
- Shopping cart management
- Order placement and tracking
- Product reviews and ratings

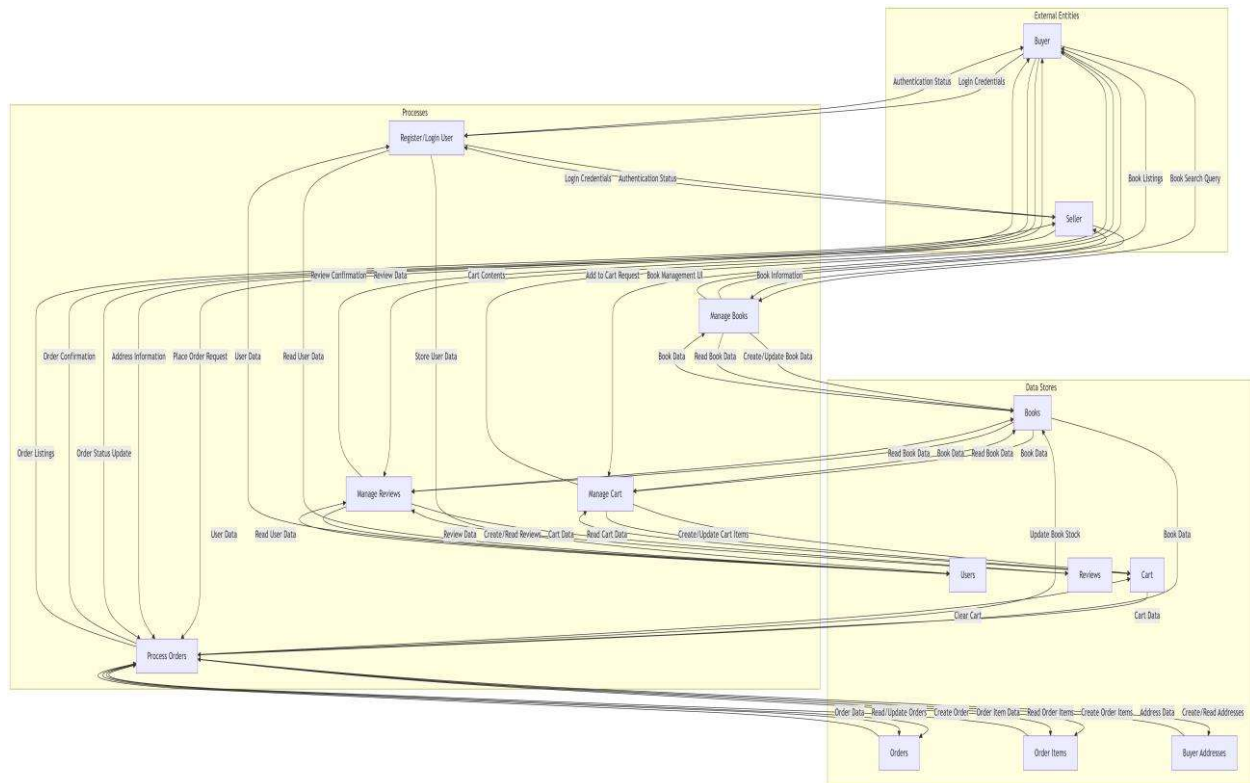
5.2 Seller Side Features

- Book listing management (CRUD operations)
- Sales dashboard with order overview
- Order status management (Pending → Shipped)
- Inventory management

5.3 Backend APIs

- Authentication endpoints (register, login, profile)
- Book management endpoints
- Cart management endpoints
- Order processing endpoints
- Review management endpoints

6. Data Flow Diagram



The data flow diagram illustrates how information moves through the system:

- External entities (Buyers and Sellers)
- Processes (User management, Book management, Cart management, etc.)
- Data stores (Database tables)
- Data flows between components

7. Implementation Details

7.1 Frontend Implementation

The React Native frontend uses:

- React Navigation for screen transitions

- Context API for state management
- Custom UI components for consistency
- API integration for backend communication

7.2 Backend Implementation

The Node.js/Express.js backend implements:

- RESTful API design
- JWT-based authentication
- Role-based authorization
- Database integration with proper error handling
- Image upload handling with Multer

7.3 Database Implementation

The PostgreSQL database features:

- Proper table relationships with foreign keys
- Indexing for performance optimization
- Data validation constraints
- Timestamp tracking for all records

8. Setup Instructions

8.1 Prerequisites

- Node.js (v14 or higher)
- PostgreSQL database
- Expo CLI
- Git

8.2 Backend Setup

1. Clone the repository: git clone https://github.com/Joshua633/Delhi_book_market.git
2. Install dependencies: npm install
3. Create a PostgreSQL database
4. Configure environment variables in .env file
5. Run database migrations
6. Start the server: npm run dev

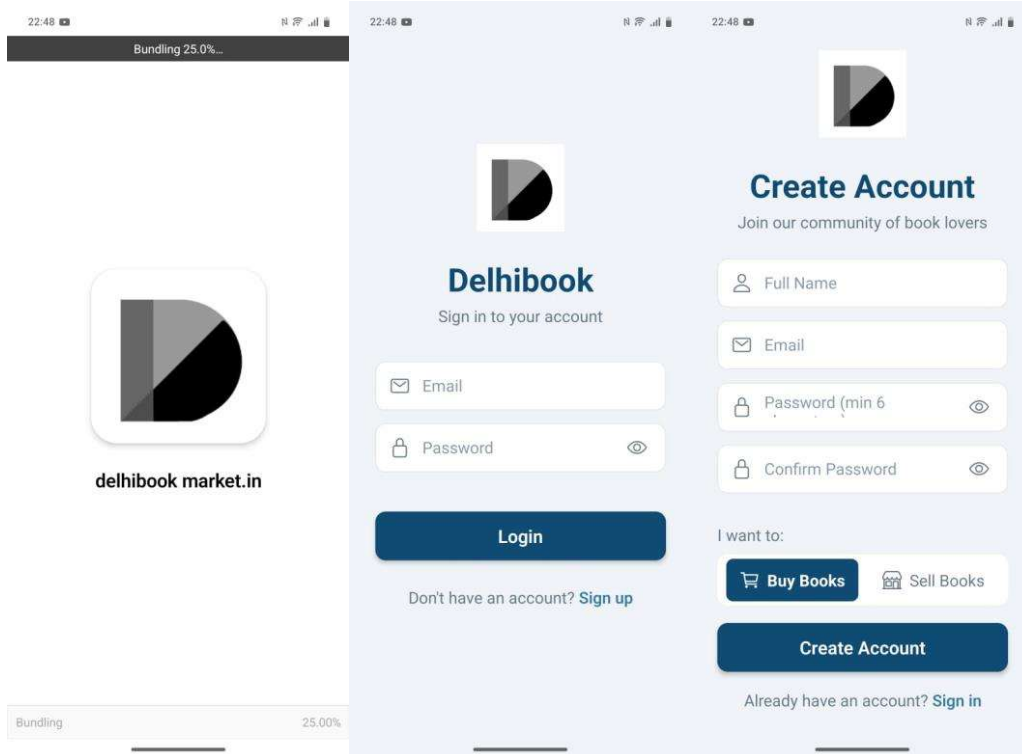
8.3 Frontend Setup

1. Navigate to the frontend directory
2. Install dependencies: npm install
3. Configure API base URL
4. Start the application: npx expo start

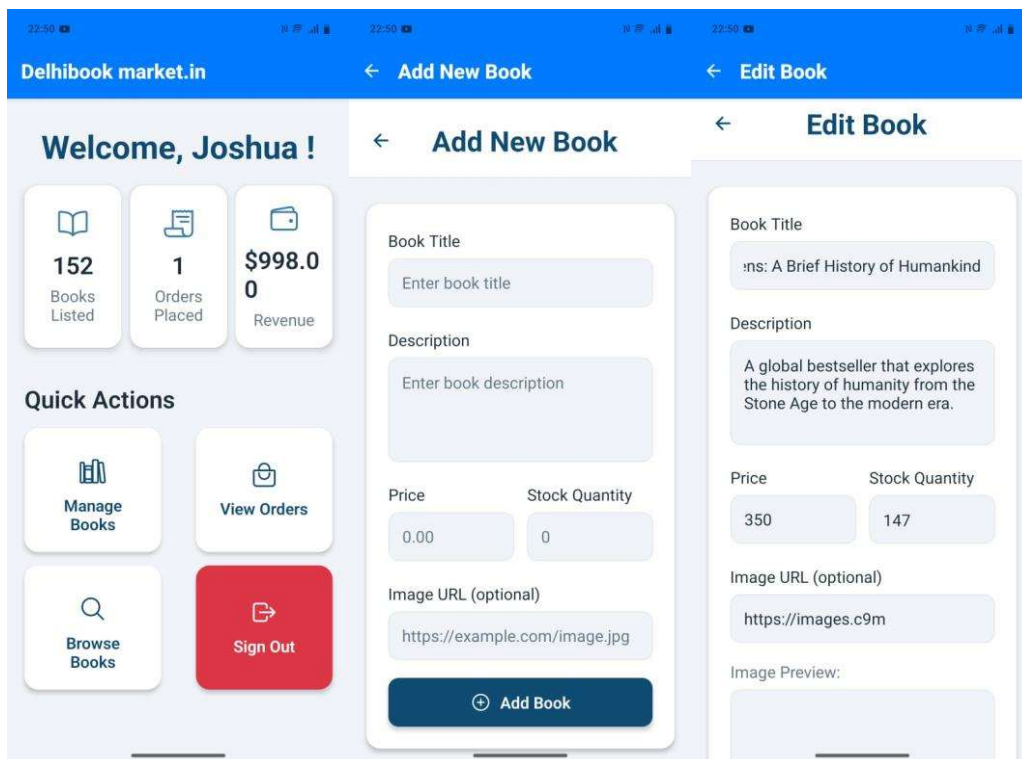
8.4 Environment Variables

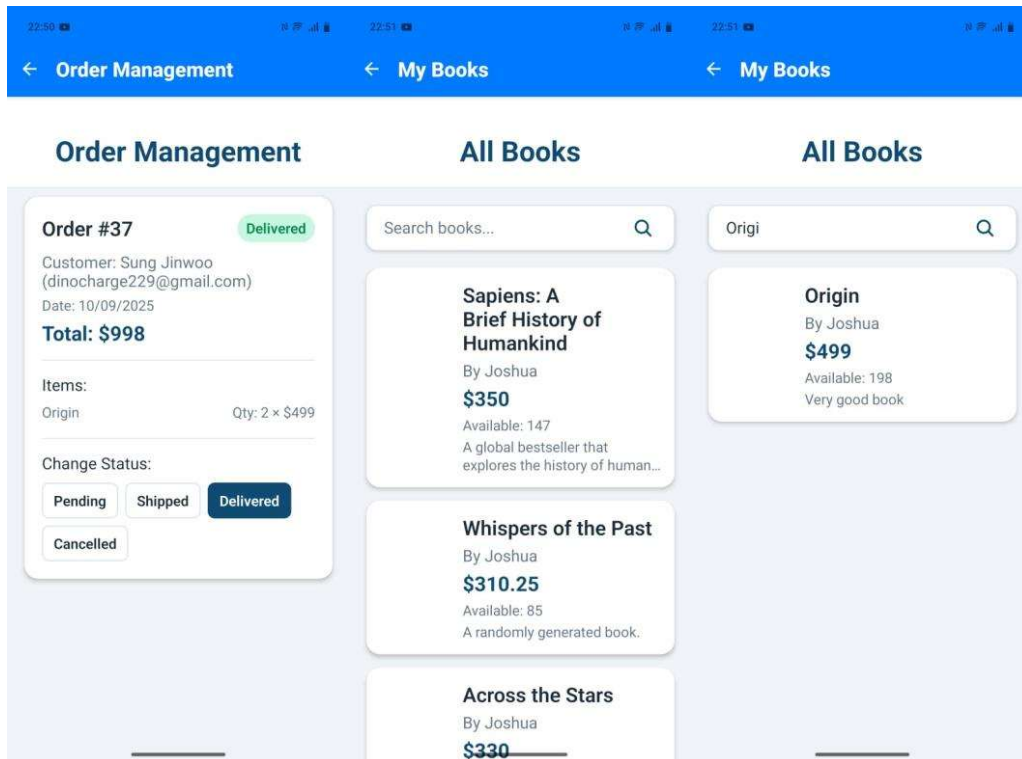
```
DB_HOST=localhost  
DB_PORT=5432  
DB_NAME=your_database_name  
DB_USER=your_username  
DB_PASSWORD=your_password  
JWT_SECRET=your_jwt_secret  
UPLOAD_PATH=./uploads
```

9. Screenshots

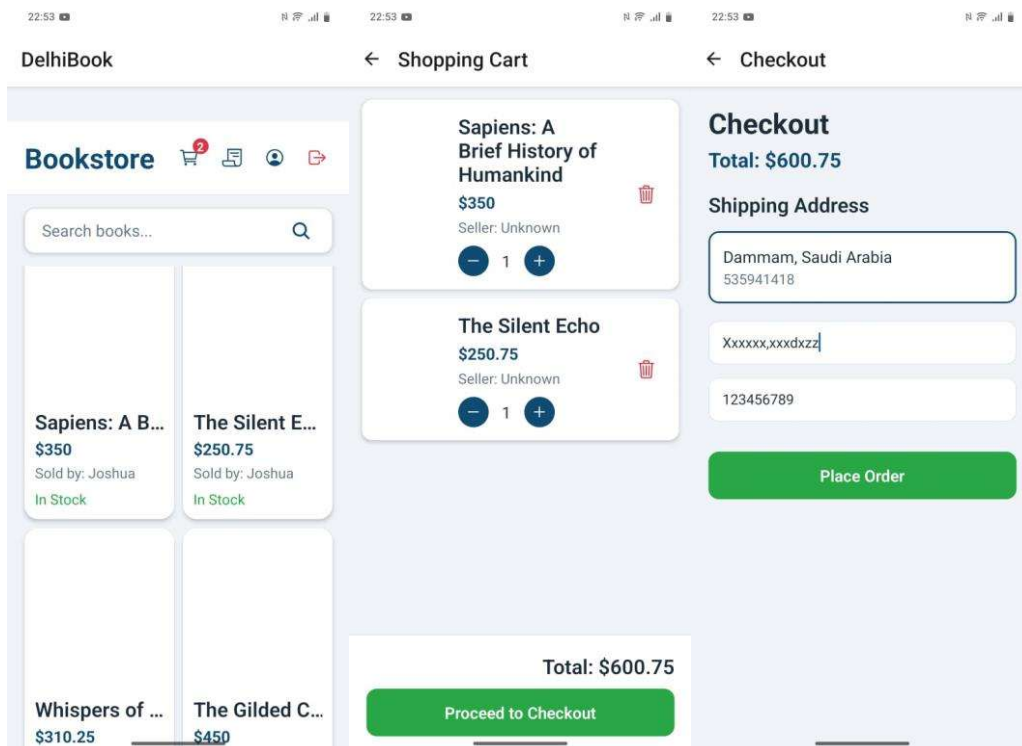


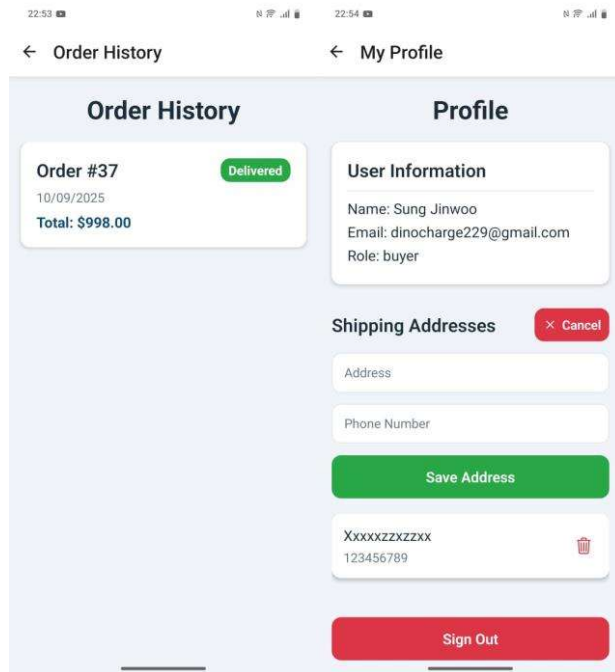
Seller Screens





Buyer Screens





10. Conclusion

The Multi-Seller Bookstore App successfully implements all required features with a clean, maintainable codebase. The application demonstrates proper separation of concerns between frontend and backend, role-based access control, and efficient database design. The use of React Native provides a cross-platform solution with native-like performance, while the Node.js/Express backend ensures scalability and maintainability.

The project meets all assignment requirements and provides a solid foundation for a production-ready multi-vendor marketplace application. Future enhancements could include payment integration, push notifications, advanced search features, and social functionality.

GitHub Repository: https://github.com/Joshua633/Delhi_book_market.git