TITLE: BOOK EXCHANGE PLATFORM

Group: 23

Problem Statement:

Book lovers frequently accumulate a collection of books they have read and look for other recommendations. They are always eager to explore new reading material. Traditional methods of exchanging books, such as local book swaps or lending among friends, are limited in scope and accessibility. Therefore, it is imperative to have a digital platform that can facilitate book exchanges on a larger scale. This platform should connect users with similar reading interests, enabling them to trade books easily and efficiently. This project aims to develop a full-stack web application that serves as a centralized platform for users to exchange, lend, and borrow books with other users. The platform should provide a user-friendly interface, robust search and recommendation features, and secure transaction capabilities.

Features:

1. **User Authentication**: Implement a secure user authentication system to allow users to register, log in, and manage their accounts.
2. **Book Listing**: Enable users to list books they want to exchange or lend, including details such as title, author, genre, condition, and availability status.
3. **Book Search**: Provide users with advanced search and filtering options to discover books based on criteria like author, title and availability.
4. **Exchange Requests**: Allow users to send and receive exchange requests for specific books, including negotiation options for terms such as delivery method and duration.
5. **Messaging System**: Implement a messaging system to facilitate user communication regarding book exchanges, including negotiation details, logistics, and scheduling. (Mock API s can be used)
6. **User Profiles**: Enable users to create profiles with information about their reading preferences, favorite genres, and books they currently own or wish to acquire.
7. **Transaction Management**: Provide tools for users to track the status of their exchange transactions, including pending requests, accepted exchanges, and completed transactions.

Expected outcome

The book exchange platform should provide users with a convenient and efficient way to discover new reading material, share their favorite books with others, and connect with fellow book enthusiasts in their community. The platform aims to promote a reading culture and foster a sense of community among users by facilitating book exchanges.

Group 23 Members:

1. **Sayali Kadam (2023MT93165)**
2. **Aryan Kaushik (2023MT93174)**
3. **Sachin Mejari (2023MT93173)**
4. **Khush Virmani (2023MT93176)**

Division of Work

Activity	Marks	Responsibility	Tasks done
Design:	10%	All Group member	
Frontend:	10%	Group Member 1 and 2	1. Registration of customers (User Authentication) 2. Book Listing 3. Book Search 4. Messaging System
Frontend:	10%	Group Member 3 and 4	1. User Profiles 2. Transaction Management 3. Exchange Requests 4. Cart Management
Backend:	10%	Group Member 1 and 2	1. Registration of customers (User Authentication) 2. Book Listing 3. Book Search 4. Messaging System
Backend:	10%	Group Member 3 and 4	1. User Profiles 2. Transaction Management 3. Exchange Requests 4. Cart Management
Integration:	10%	All Group members.	All Members

4. Introduction

Online book exchange platform, "***BookShare***" aids users to share their favorite books with others and connect with fellow book enthusiasts in their community. The platform aims to promote a reading culture and foster a sense of community amongst users by facilitating book exchanges.

5. Description of Application

BookShare has the following components.

- **Entities**
  - Customers : Readers / Book sharing enthusiasts
- **Features**

i. Registration of customers (User Authentication)
ii. Book Listing
iii. Book Search
iv. Purchase/Exchange Requests
v. Messaging System
vi. User Profiles
vii. Cart Management
viii. Transaction Management
ix. Light and Dark theme

6. User Stories and Acceptance criteria

EPIC 1: Login, Signup and User Profile

1.1. Signup:

User Story: As a new user, I should be able to sign up as a Reader / Book Owner so that I can use the platform accordingly

Acceptance Criteria:

- Input data fields to enter:
  - Name
  - Email id
  - Password
  - User Role: Reader / Book owner / Both

- After successful validation of all entered fields and on clicking Submit, show message indicating successful signup.

1.2. Login:

User Story: As a user I should be able to login with the username and password to the portal.

Acceptance Criteria:

- Input data fields to enter:
    - Username
    - Email id
    - Password
    - Role
- Indicate invalid usernames and passwords as alerts to the user.
- After successful validation of all entered fields and on clicking Submit, show message "Login Successful" and redirect to the platform Homepage.

1.3. User Profiles:

User Story: As a user I should be able to view my details so that I can view them and/or modify them.

Description:

Details like:

- Basic user information
- Address - City, Country
- Contact details
- Favourite books/bookmarks
- Premium user benefits

Acceptance Criteria: Via a click user is able to view their profiles if created and make modifications, else a form is opened for user to enter their details.

EPIC 2: Book Exchange Features

2.1. Book Listing:

User Story: As a user, I should be able to see books tagged with specific categories like Fiction, Non-Fiction, Humor, Education, Novels, etc.

Description: Pictorial description of the books with price and add-to-cart functionality.

Acceptance Criteria: The page should show default listings with multiple genres of book, a good-to-have would be to custom-tailor it for a user based on history.

2.2. Book Search:

User Story: As a user, I should be able to search book by its name.

Acceptance Criteria: When user searches for a specific book, it should show up at the top, otherwise "Book not found" should show up.

2.3. Exchange Requests:

User Story: As a user, I should be able to select multiple books of my choice so that I can checkout them all at once.

Acceptance Criteria:

- Add to cart functionality, multiple books from the listing can be added and for a common checkout.
- Each listing has cost mentioned, as decided by the seller, some can be free as well for mutual sharing of books in the community.

2.4. Messaging System:

User Story: As a user I should be able to talk to the book owner so that I can communicate, discuss interests and/or negotiate.

Description: Before checkout, user may communicate directly with the seller to initiate communication.

Acceptance Criteria: User is able to click the chat option for particular seller and is able to communicate with them without hassle.

EPIC 3: Transaction Management

3.1. Cart Management

User Story: As a user I should be able to view and modify my cart so that I can view all my selections at a glance, total cart amount and finalize transaction.

Acceptance Criteria: User should be able to view all their selections, and be able to modify the list by removing any item if required, and if cart is empty, then user should be able to se this.

3.2. Payments System

User Story: As a user I should be able to easily transact with my choice of mode of payment.

Acceptance Criteria: If payment is successful, "Order placed" should be visible with communication to buyer regarding confirmation of order, otherwise "Payment failed" should be highlighted with the reason of failure.

4. Design, Operations and Services

4.1. Frontend

- The frontend of the application will be responsible for providing a user-friendly interface for users.
- The frontend will communicate with the backend server through RESTful APIs to perform actions like login, searching books, viewing listings, placing orders, etc.
- Using **React** and **Express JS** as core technologies to implement frontend.

4.2. Backend

- The backend will handle all the business logic, process requests from the frontend, interact with the database, and coordinate with external services for payment processing and delivery status updates.

- It can be designed using a microservices architecture to improve modularity and scalability. Microservices will enable independent development and deployment of different application modules.
- The backend will expose RESTful APIs to be consumed by the frontend and any other potential clients.
- Using **Node JS** as core technology to implement backend.

4.3. Database

- The application's data will be stored in a reliable and scalable database system like Mongodb, etc.
- Books, book listing, user information is stored and fetched from the database.

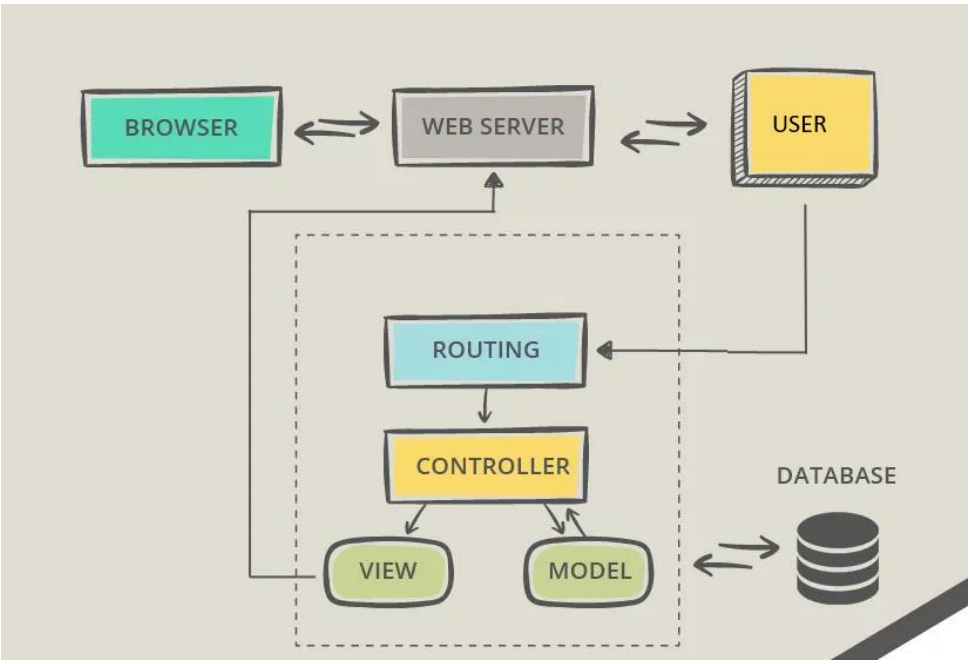
4.4. Authentication

- Users can access the platform and view listings, but to place order or communicate with others, they would have to register or login to maintain access to authentic users only.

4.5. Cart management and order placement

- Provide easy-to-use interface for users to add items to their cart and review items in cart and proceed to checkout for payment.
- Integration with secure and reliable payment gateway is critical to handle online transactions. Popular options include Razorpay, Gpay, PayPal.

5. Architecture



6. REST API endpoints with URI and HTTP methods

Create Book's Category

Endpoint POST: /books/category

Request

```
{
   "name": "",
   "description": ""
}
```

Response

```
{
   "status": "",
   "message": ""
}
```

Get Book's Categories

Endpoint GET - /books/categories

Response

```
{
   "status": "",
   "message": "",
   "categories": [
       {
           "id": 1,
           "name": "",
           "description": ""
       }
   ]
}
```

Create book

Endpoint POST - /books/new-book

Request

```
{
   "title": "",
   "description": "",
   "categoryId": 0,
   "price": 0.0,
   "rent": 0.0,
   "author": "",
   "authorMobile": "",
   "authorEmail": "",
   "coverImage": ""
}
```

Response

```
{
   "status": "",
   "message": ""
}
```

Books by category id

Endpoint GET - /books/category/{categoryId}

```
{
   "status": "",
   "message": "",
   "books": [
       {
           "id": 1,
           "name": "",
           "coverImage": "",
           "price": "",
           "rentPerDay": ""
       }
]
}
```

Book details

Endpoint GET - /books/book-details/{bookId}

```
{
   "status": "",
   "message": "",
   "books":{
           "id": 1,
           "name": "",
           "description": "",
           "author": "",
           "categoryId": 1,
           "category": "",
           "coverImage": "",
           "price": 0.0,
           "authorMobile": "",
           "authorEmail": "",
           "rentPerDay": 0.0
       }
}
```

Search by name, Author, categories

Endpoint GET - /books/all-books?filter = {name, Author, categories}


```
{
   "status": "",
   "message": "",
   "books": [
       {
           "id": 1,
           "name": "",
           "description": "",
           "author": "",
           "categoryId": 1,
           "category": "",
           "coverImage": "",
           "price": "",
           "authorMobile": "",
           "authorEmail": "",
           "rentPerDay": ""
       }
   ]
}
```


Cart

Endpoint POST: /cart/add-to-cart

Request


```
{
   "books": [
       {
           "bookId": 1,
           "purchaseType": "own/rent",
       }
   ]
}
```


Response


```
{
   "status": "",
   "message": "",
   "data": {
       "totalPrice": 0.0,
       "books": [
           {
               "bookId": 1,
```

```
                "bookName": "",
                "purchaseType": "",
                "price": 0.0
            }
        ]
    }
}
```

Create order

Endpoint POST: /cart/purchase

Request

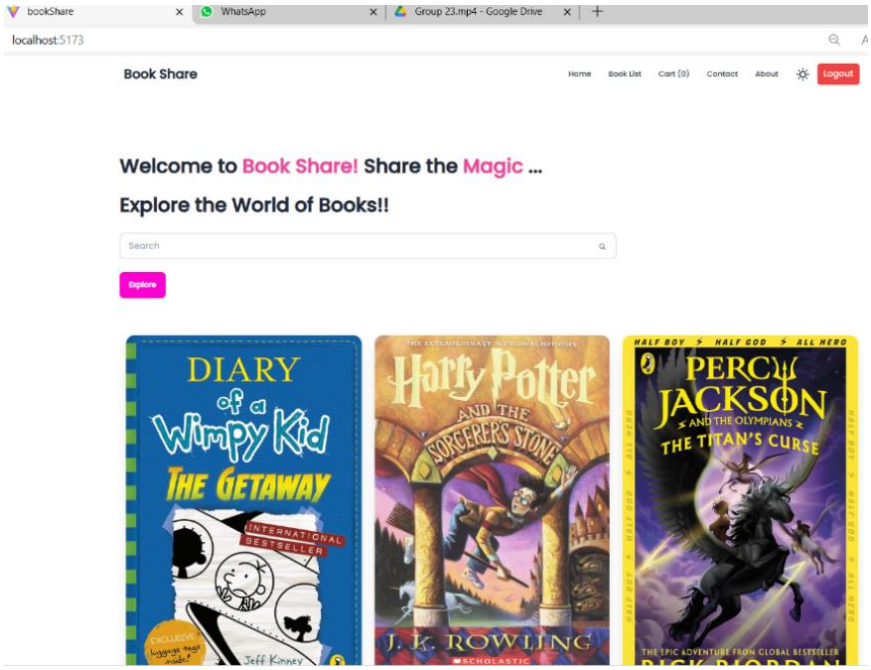
```
{
    "name": "",
    "mobile": "",
    "email": "",
    "address": "",
    "pincode": "",
    "totalPrice": 0.0,
    "payMode": "cod",
    "books": [
        {
            "bookId": 1,
            "bookName": "",
            "purchaseType": "",
            "price": 0.0,
            "quantity": 1,
            "days": 1
        }
    ]
}
```

Response

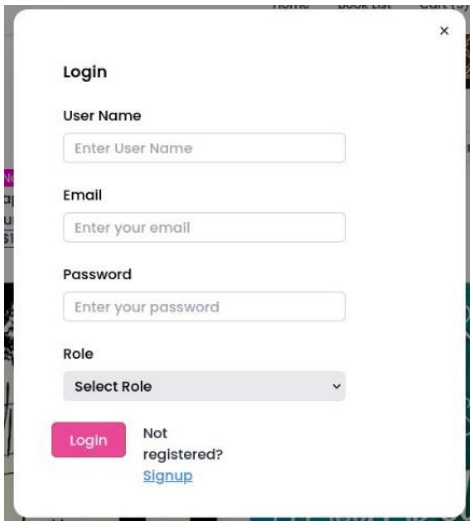
```
{
    "status": "",
    "message": "",
    "data": {
        "orderNo": ""
    }
}
```

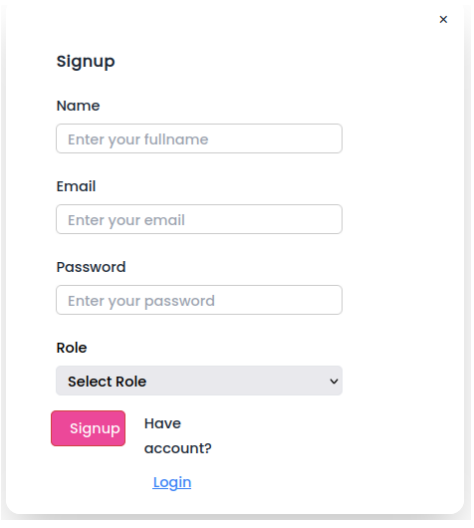
7. User Interface

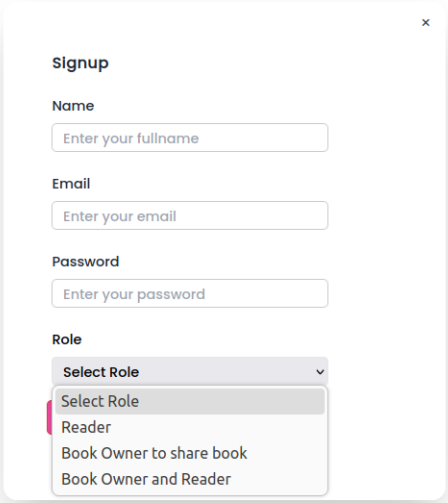
7.1. Home page



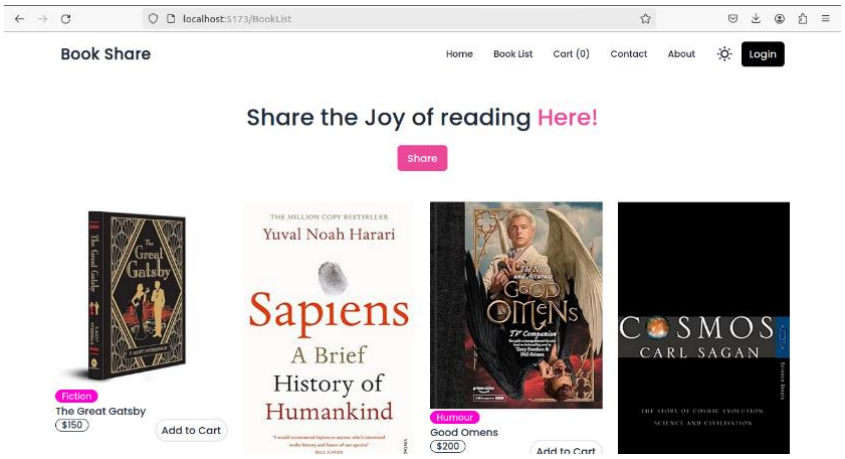
7.2. Login and Signup

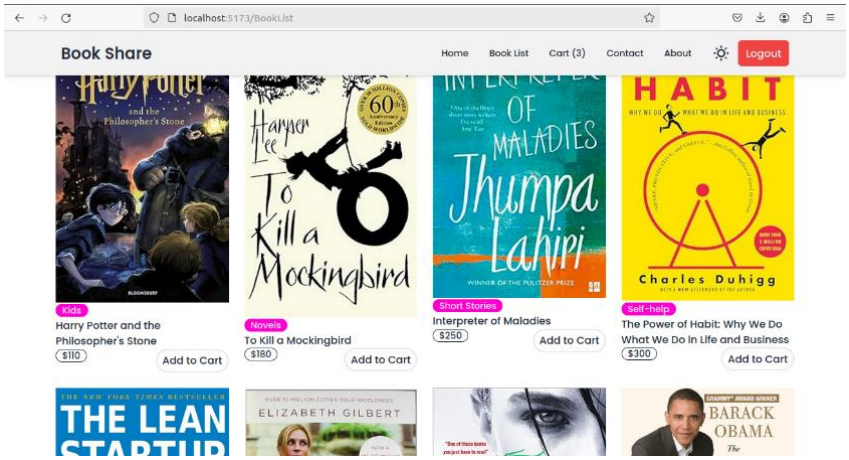




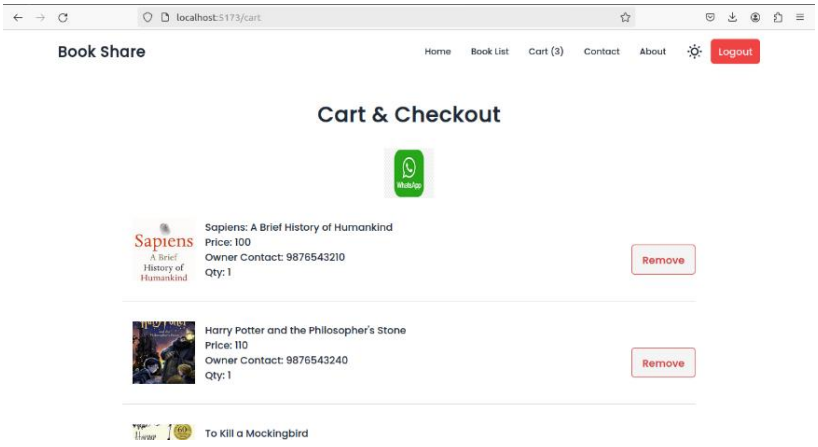


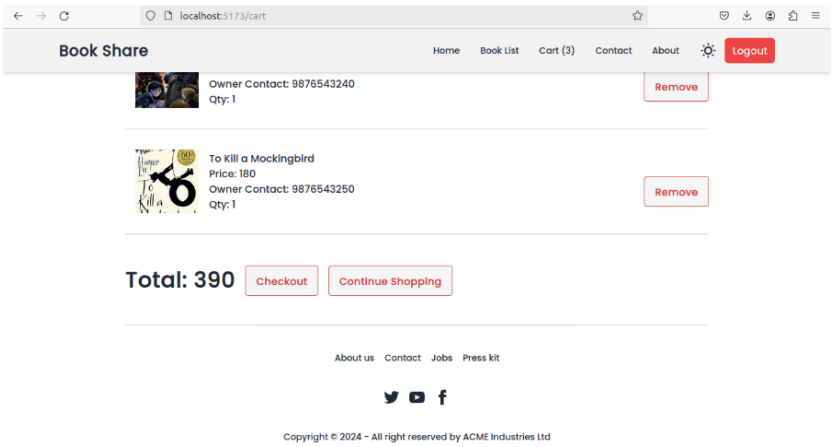
7.3. Book listing

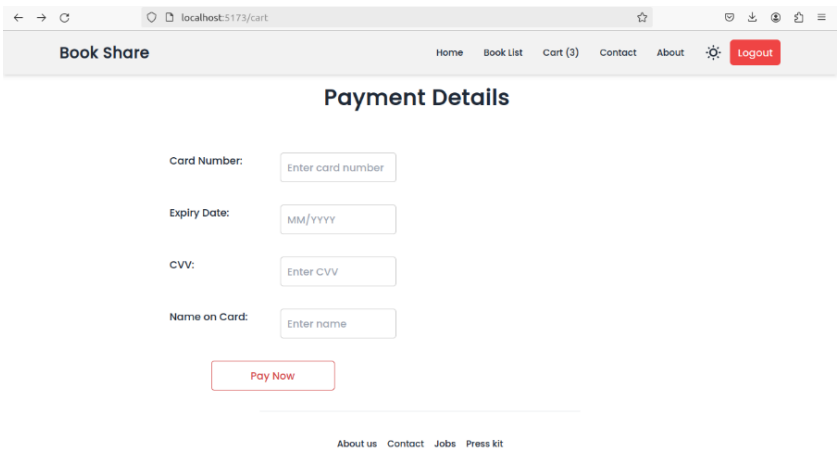




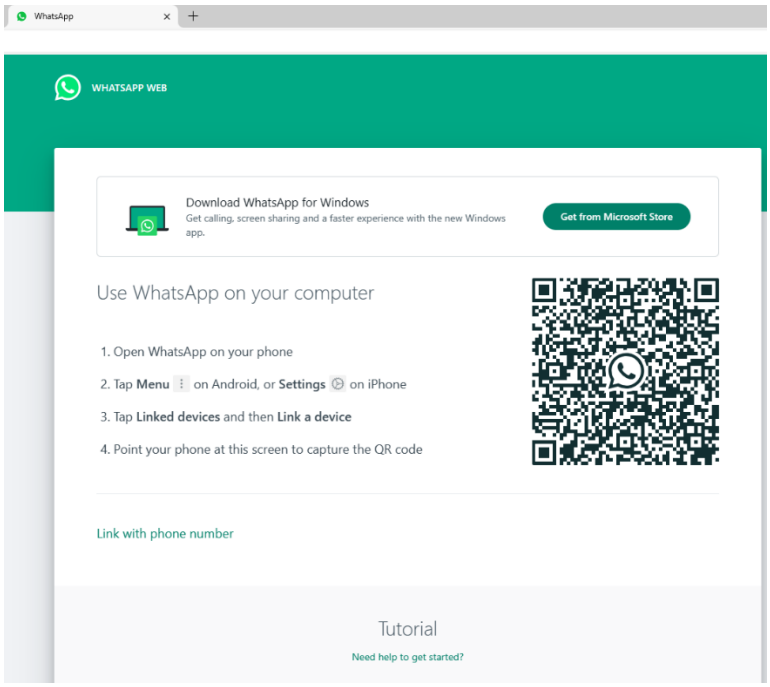
7.4. Cart and Checkout



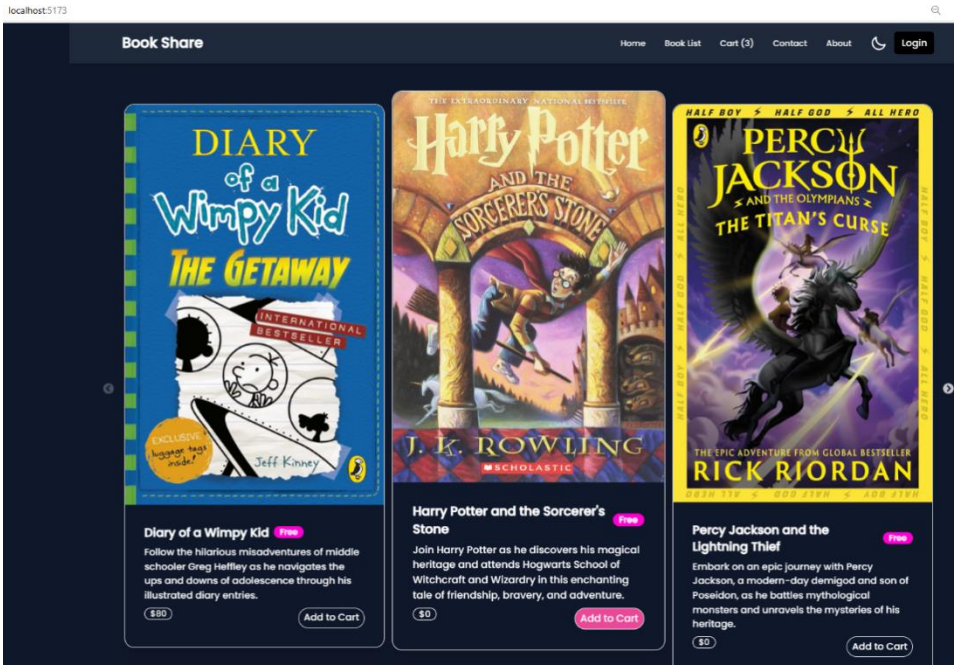




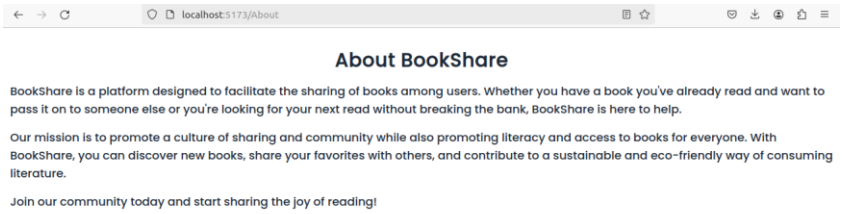
7.5. Messaging



7.6. Dark theme



7.7. About



8. Future Scope

- Modifiable user profile section
- Enhance transaction gateway, to support multiple payment modes
- Community chat rooms
- Add subscription model/charges and premium user benefits

9. GitHub Repositories

Link:

https://github.com/SKM5/FSAD_BookShare.git

10. Demonstration Video

Link:

https://drive.google.com/file/d/1-BVjB0nRulalye427l8-NgJx2omiqy3m/view?usp=sharing