OUSL BOOK SHOP

Website Report

2024.11.08

Contents

1. Introduction

1.1 Background of the OUSLBookStore

OUSLBookStore is a small-scale online bookstore that seeks to enhance its operational capabilities by transitioning to a web-based system for managing book sales. The current manual or limited digital processes have constrained their ability to efficiently handle customer orders, book inventory, and payment processing. To stay competitive and improve customer experience, the bookstore requires a solution that allows customers to browse, purchase books, and make payments online.

The new web-based platform will allow the bookstore to reach a larger audience, cater to a more diverse customer base, and streamline internal management processes, specifically in inventory and sales tracking. This initiative will also address the scalability needs of the bookstore, enabling the system to handle multiple users simultaneously and provide a seamless experience across desktop and mobile devices.

1.2 Objectives of the System

The primary objective of the OUSLBookStore system is to develop a web-based platform that manages the end-to-end process of online book sales. The platform must offer a user-friendly interface where customers can:

- Create accounts,
- Browse and search for books,
- Add books to a shopping cart,
- Complete secure online payments.

Additionally, the system will include an administrative interface for bookstore staff to:

- Manage the book inventory (add, update, and remove books),
- Generate sales reports to track orders and revenue.

Security, scalability, and responsiveness are key factors in the system design, ensuring a smooth and secure experience for customers and efficient management for administrators. The system is expected to handle up to 1,000 users simultaneously and be accessible on both desktop and mobile devices.

The report outlines the development process of this system, detailing the requirements, system design, implementation, and the final testing and deployment plan to meet the business goals of OUSLBookStore.

2. System Requirements

2.1 Functional Requirements

Must Have

1. Customer registration and login functionality.
2. Book browsing by title, author, and genre.
3. Shopping cart to add/remove books before purchase.
4. Secure online payment gateway for checkout.
5. Admin interface for adding, updating, or removing books from inventory.
6. Admin functionality to generate sales reports to track orders and revenue.

Should Have

1. Search functionality with filters like genre, author, and rating.
2. Mobile and desktop responsive design.

Could Have

1. Book rating and review system by customers.
2. Recommendation system based on previous purchases or preferences.

Won't Have

1. Social media integration for now.
2. Advanced analytics beyond basic sales reports.

2.2 Non-Functional Requirements

Must Have

1. Secure transactions with encryption (e.g., HTTPS, SSL).
2. Handle up to 1,000 concurrent users.
3. Ensure availability across both mobile and desktop devices.
4. User-friendly design for all features.

Should Have

1. System scalability to handle future traffic growth.
2. 24/7 availability and minimal downtime.

Could Have

1. Multi-language support for users from different regions.

3. System Design

3.1 UML Diagrams

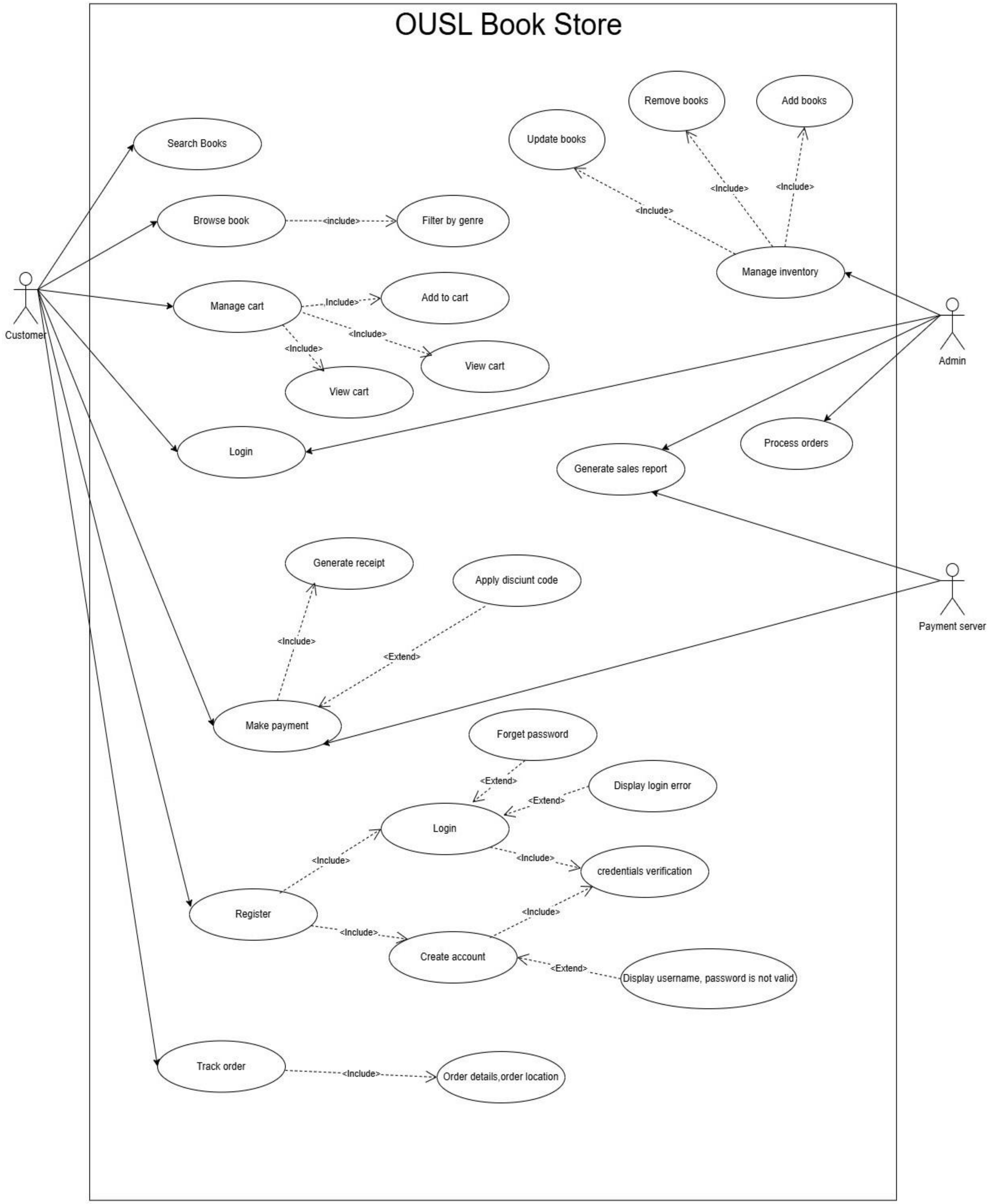
3.1.1 Use Case Diagram

3.1.2 Class Diagram

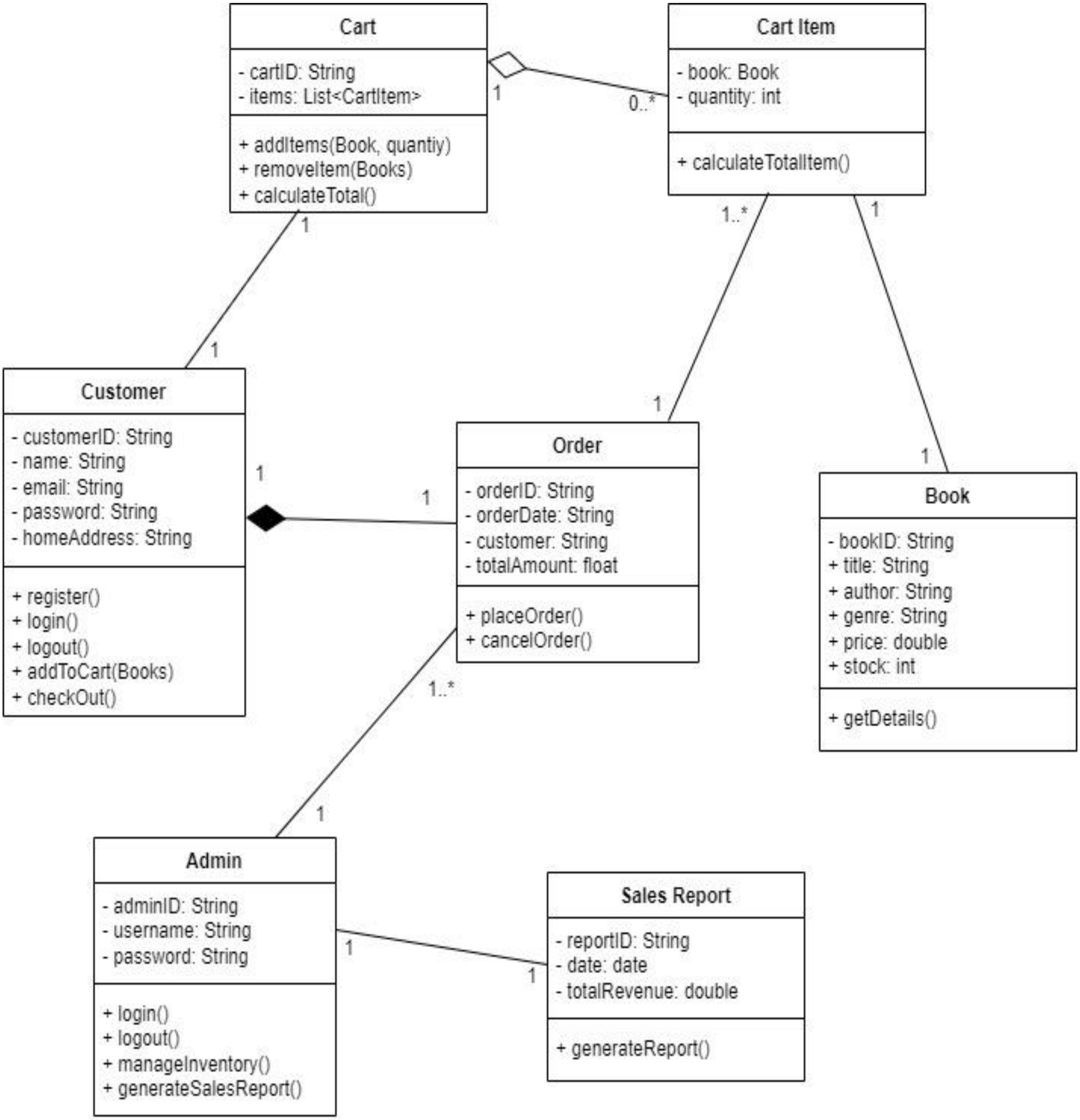
3.1.3 Sequence Diagram

3.1.4 Activity Diagram

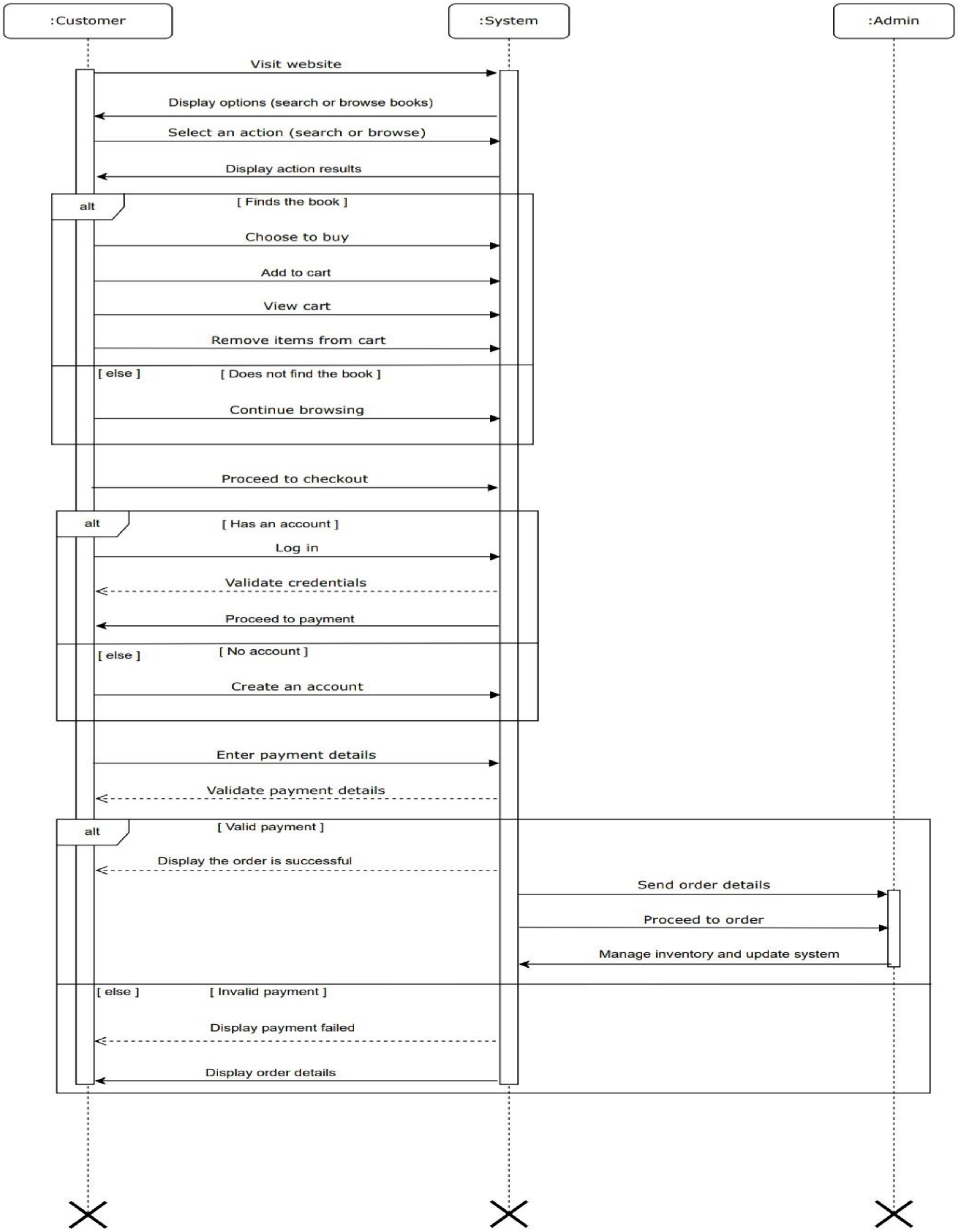
3.1.1 Use Case Diagram



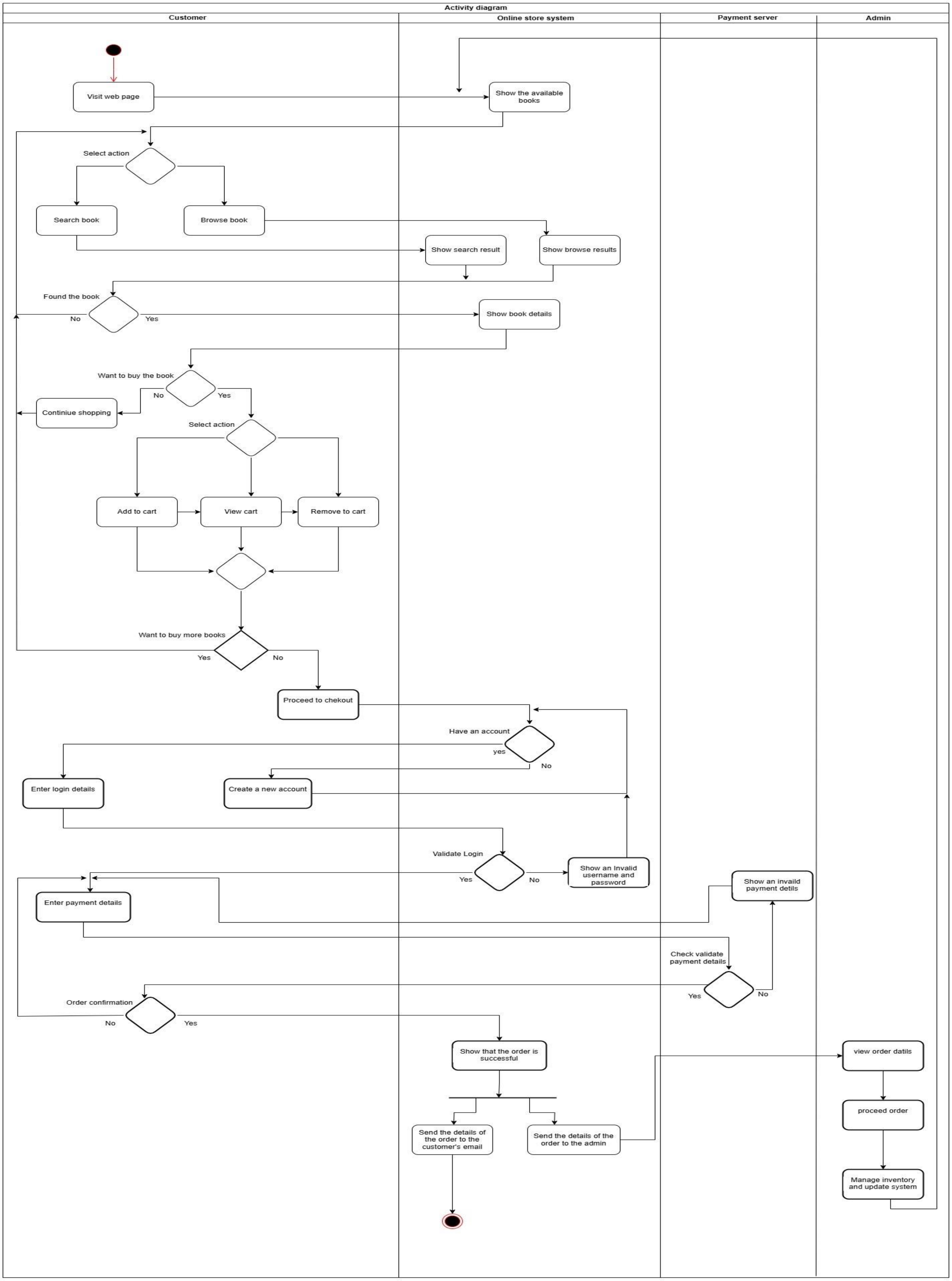
3.1.2 Class Diagram



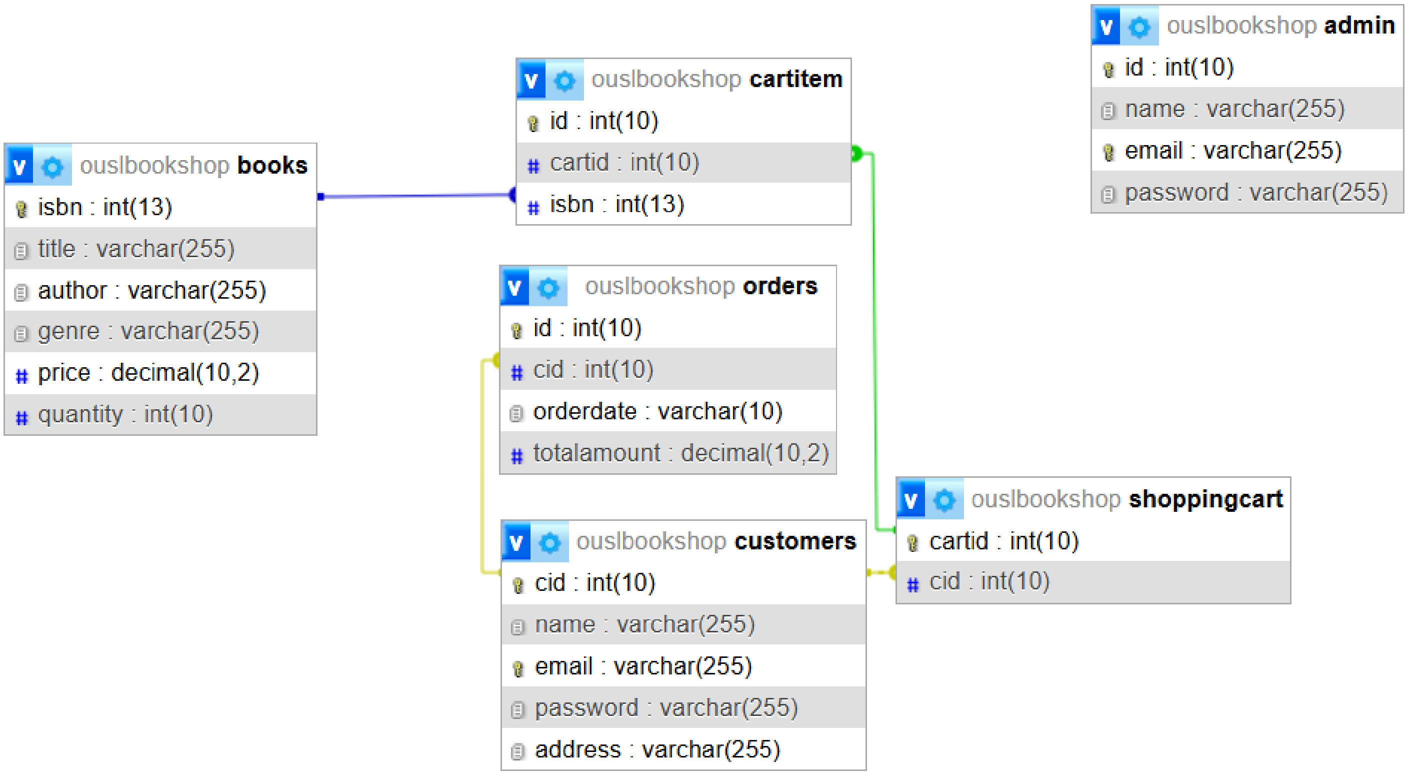
3.1.3 Sequence Diagram



3.1.4 Activity Diagram



4..1 Database Schema



ouslbookshop **books**
isbn : int(13)
title : varchar(255)
author : varchar(255)
genre : varchar(255)
price : decimal(10,2)
quantity : int(10)

ouslbookshop **cartitem**
id : int(10)
cartid : int(10)
isbn : int(13)

ouslbookshop **orders**
id : int(10)
cid : int(10)
orderdate : varchar(10)
totalamount : decimal(10,2)

ouslbookshop **customers**
cid : int(10)
name : varchar(255)
email : varchar(255)
password : varchar(255)
address : varchar(255)

ouslbookshop **admin**
id : int(10)
name : varchar(255)
email : varchar(255)
password : varchar(255)

ouslbookshop **shoppingcart**
cartid : int(10)
cid : int(10)

- **Customers:**

- id (Primary Key, int)
- name (String)
- email (String, unique)
- password (String)
- address (String)

- **Books:**

- id (Primary Key, int)
- title (String)
- author (String)
- genre (String)
- price (float)
- quantity (int)

- **ShoppingCart:**

- id (Primary Key, int)
- customerId (Foreign Key, int)

- **CartItems:**

- id (Primary Key, int)
- cartId (Foreign Key, int)
- bookId (Foreign Key, int)
- quantity (int)

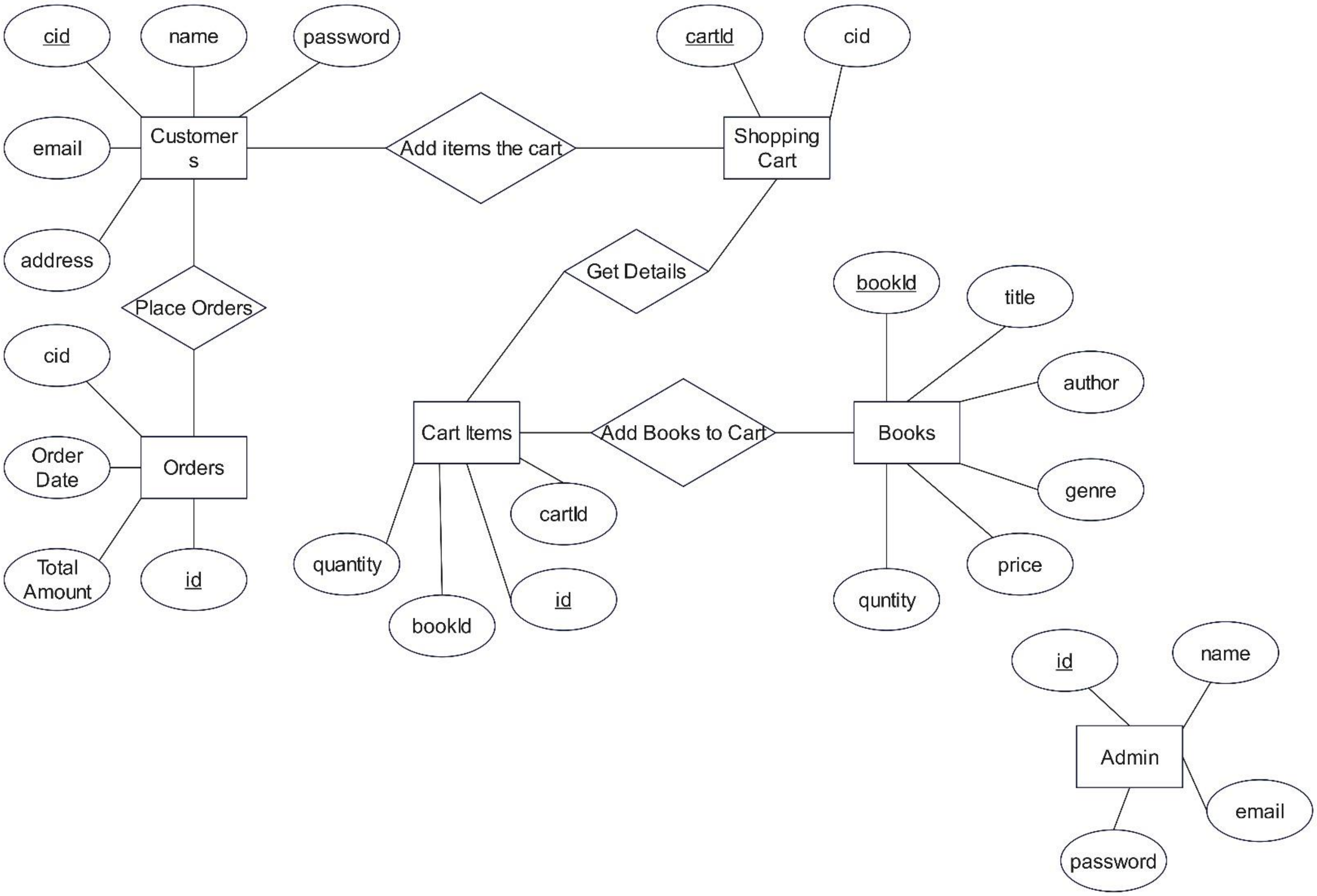
- **Orders:**

- id (Primary Key, int)
- customerId (Foreign Key, int)
- orderDate (Date)
- totalAmount (float)

- **Admins:**

- id (Primary Key, int)
- name (String)
- email (String, unique)
- password (String)

4.2 Entity-Relationship (ER) Model



5. Implementation Plan

5.1 Database Setup

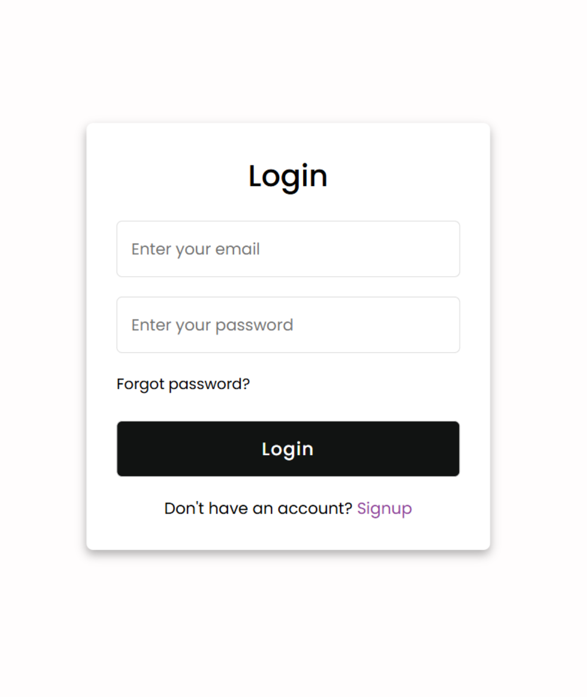
Task: Set up a relational database (MySQL/PostgreSQL).

- **Tables**: Customers, Books, ShoppingCart, CartItems, Orders, Admins.
- **Schema**:
  - Create each table following the database schema we defined earlier.
  - Define relationships (foreign keys) between tables, such as customerId in ShoppingCart and Orders.
- **Tools**: Use SQLAlchemy (for Python frameworks) or JPA (for Java frameworks) to manage database models.

5.2 Front-End Development

Task: Create a responsive user interface (UI) for the web-based system.

- **Customer Interface**:
    - Implement account registration and login.



Login

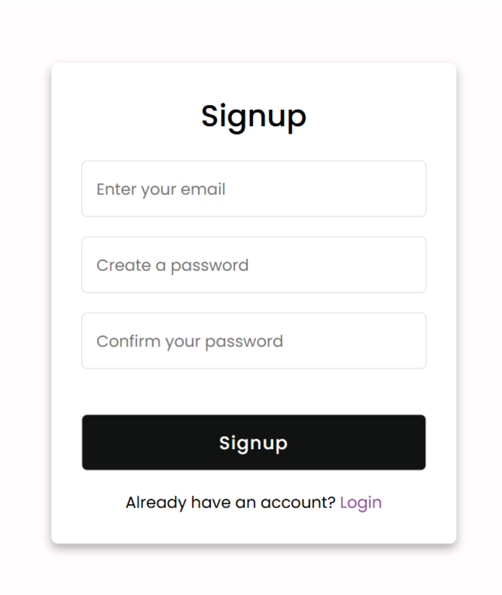
Enter your email

Enter your password

Forgot password?

Login

Don't have an account? Signup



Signup

Enter your email

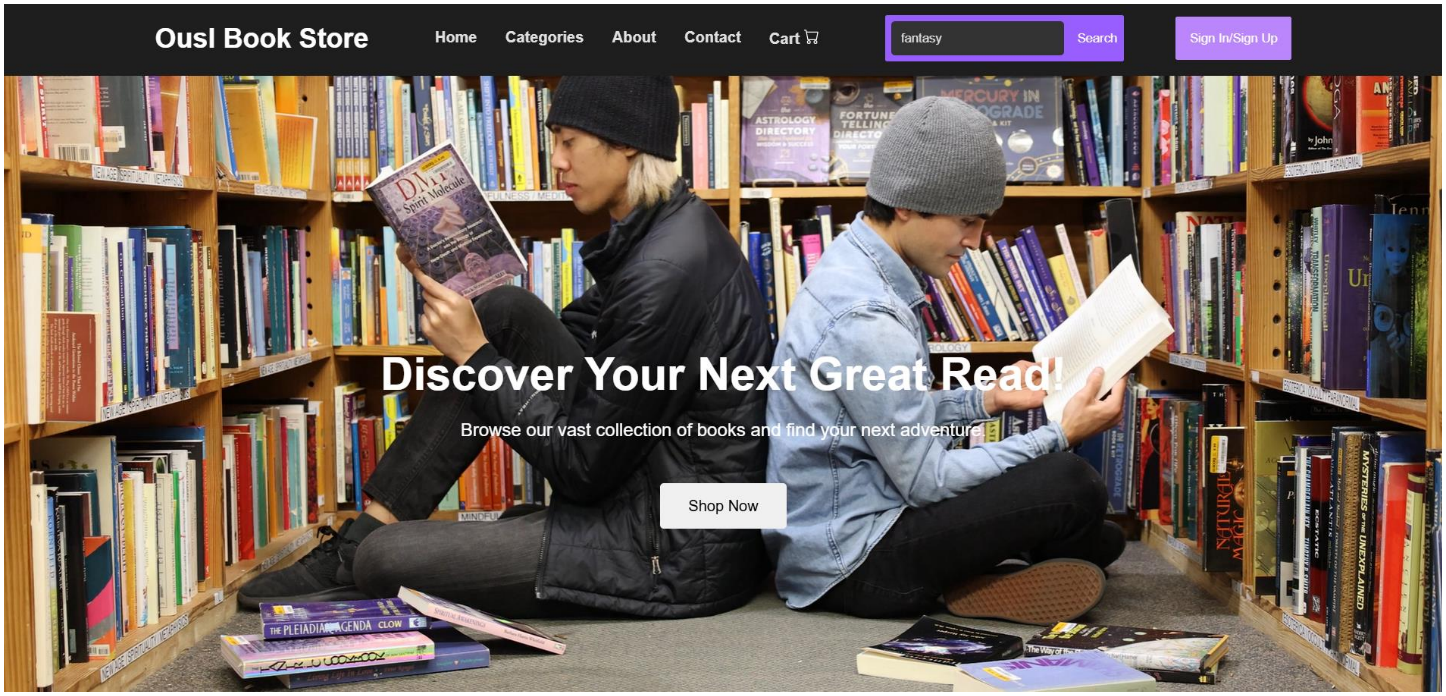
Create a password

Confirm your password

Signup

Already have an account? Login

-
    - Develop book browsing and search functionality.



Ousl Book Store

Home Categories About Contact Cart

fantasy

Search

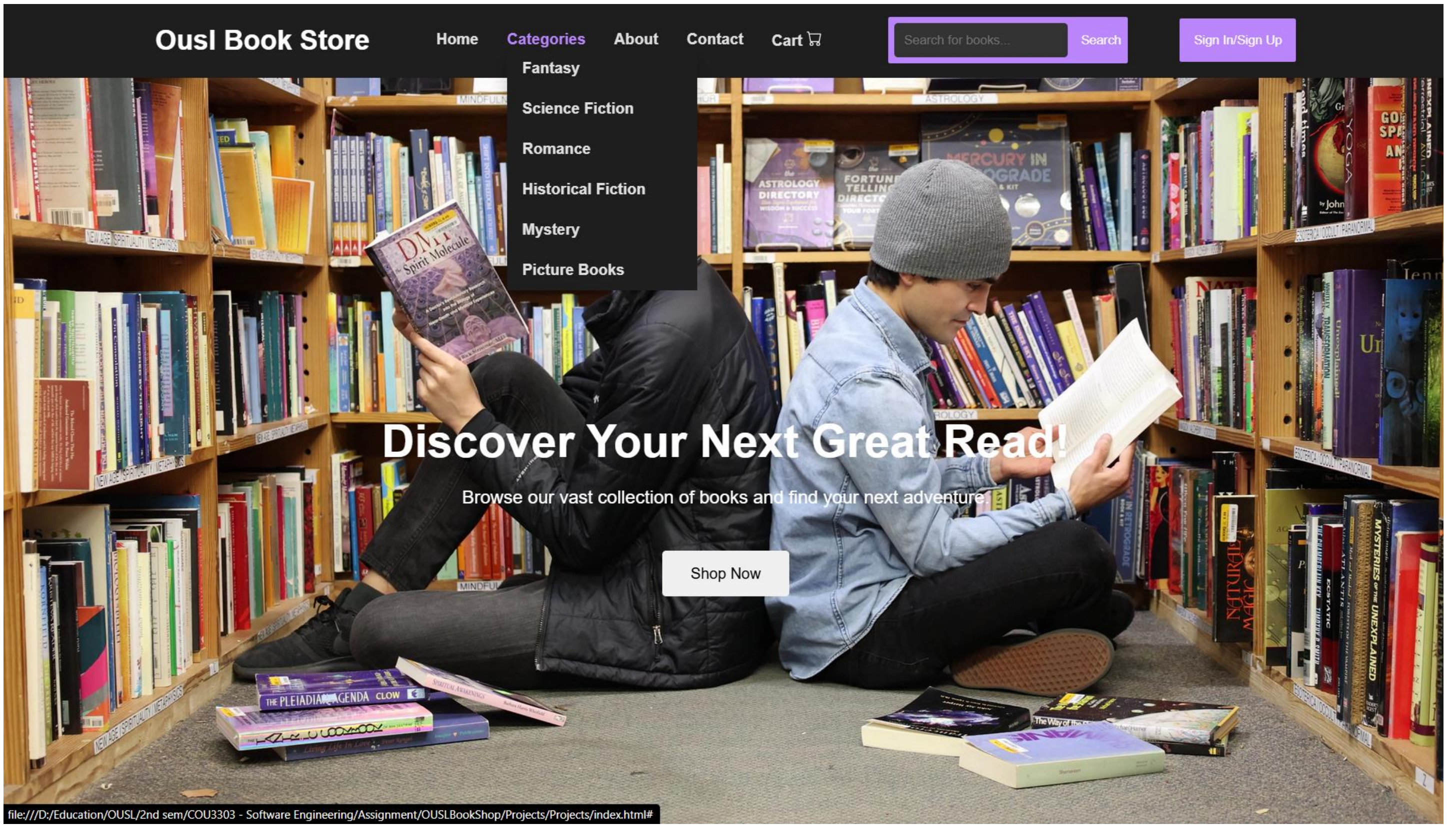
Sign In/Sign Up

Discover Your Next Great Read!

Browse our vast collection of books and find your next adventure

Shop Now

Home page with Search Function

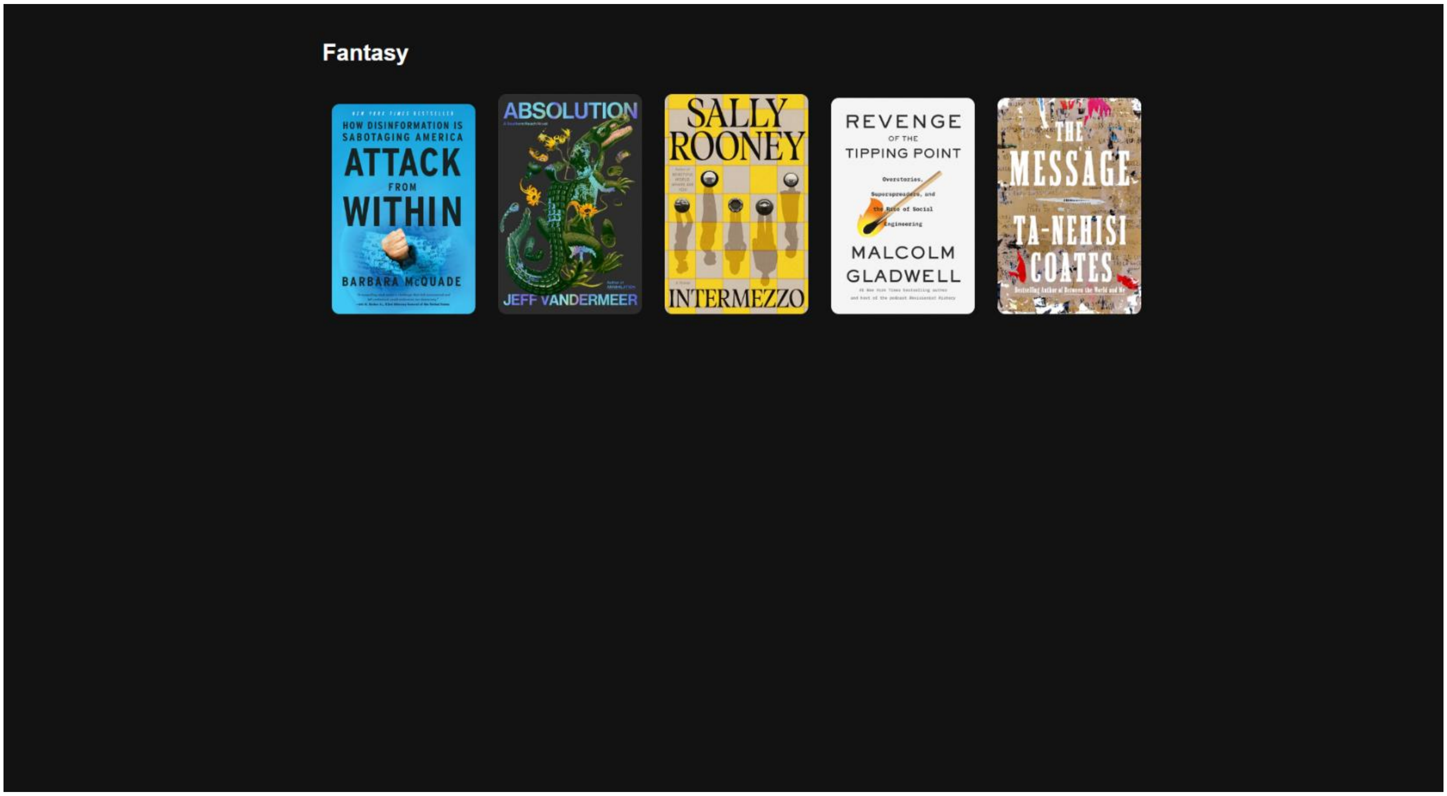


Discover Your Next Great Read!

Browse our vast collection of books and find your next adventure.

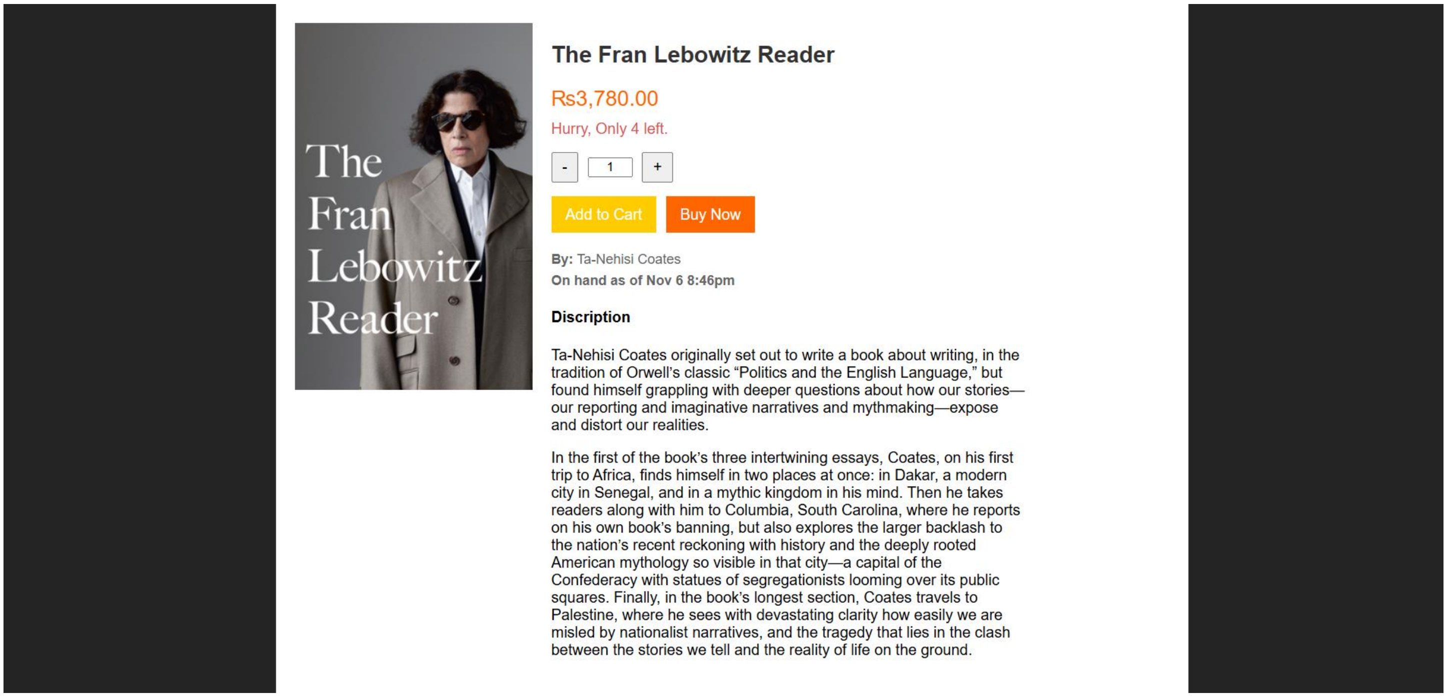
file:///D:/Education/OUSL/2nd sem/COU3303 - Software Engineering/Assignment/OUSLBookShop/Projects/Projects/index.html#

Filer by Genre



Filer by Fantasy

- Create a shopping cart interface to add/remove books and checkout.



The Fran Lebowitz Reader

Rs3,780.00

Hurry, Only 4 left.

By: Ta-Nehisi Coates

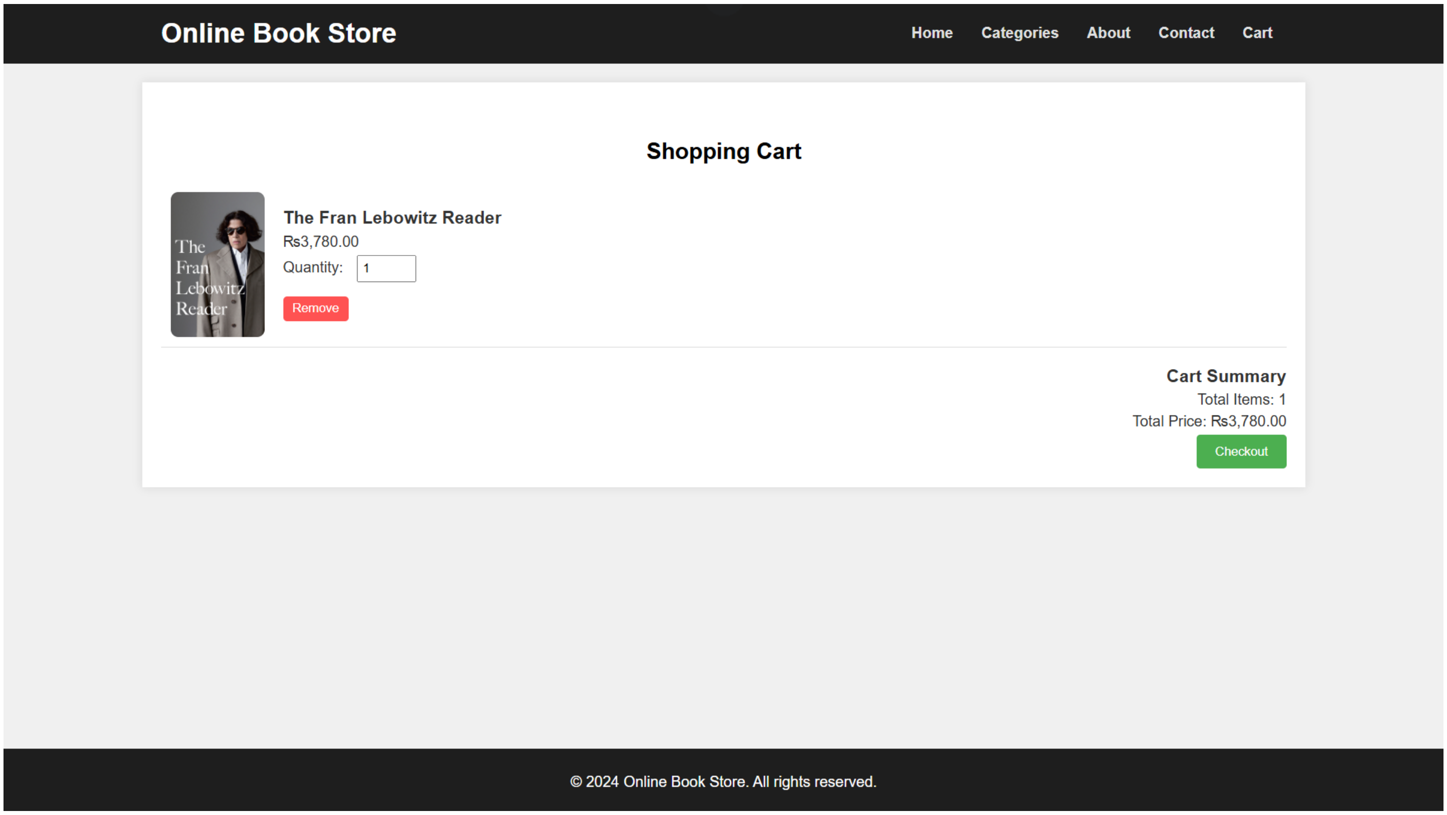
On hand as of Nov 6 8:46pm

Discription

Ta-Nehisi Coates originally set out to write a book about writing, in the tradition of Orwell's classic "Politics and the English Language," but found himself grappling with deeper questions about how our stories—our reporting and imaginative narratives and mythmaking—expose and distort our realities.

In the first of the book's three intertwining essays, Coates, on his first trip to Africa, finds himself in two places at once: in Dakar, a modern city in Senegal, and in a mythic kingdom in his mind. Then he takes readers along with him to Columbia, South Carolina, where he reports on his own book's banning, but also explores the larger backlash to the nation's recent reckoning with history and the deeply rooted American mythology so visible in that city—a capital of the Confederacy with statues of segregationists looming over its public squares. Finally, in the book's longest section, Coates travels to Palestine, where he sees with devastating clarity how easily we are misled by nationalist narratives, and the tragedy that lies in the clash between the stories we tell and the reality of life on the ground.

Book Details



Shopping Cart

The Fran Lebowitz Reader

Rs3,780.00

Quantity: 1

Cart Summary

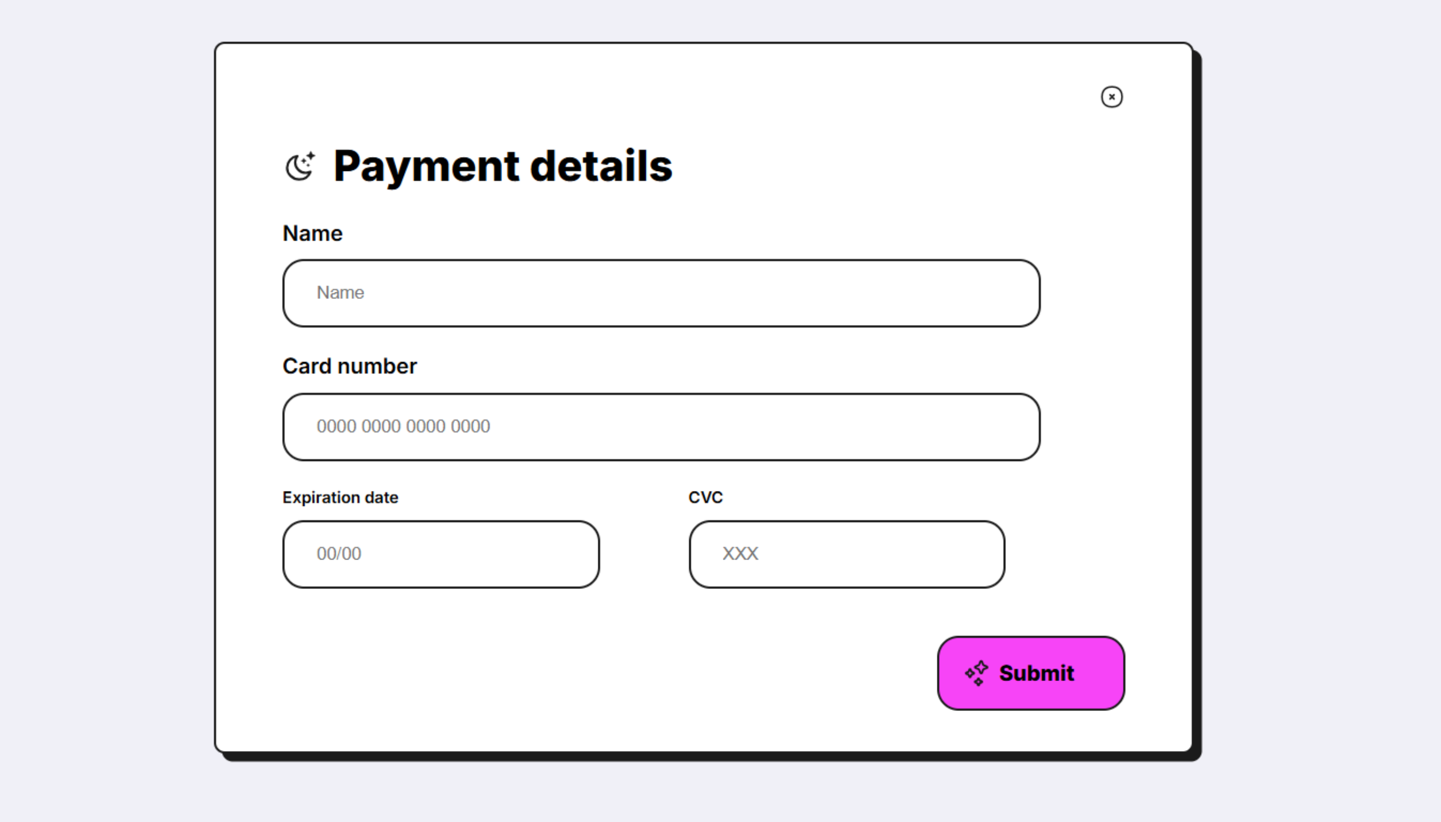
Total Items: 1

Total Price: Rs3,780.00

© 2024 Online Book Store. All rights reserved.

Shopping Cart

- Build payment integration on the checkout page.



Payment details

Name

Name

Card number

0000 0000 0000 0000

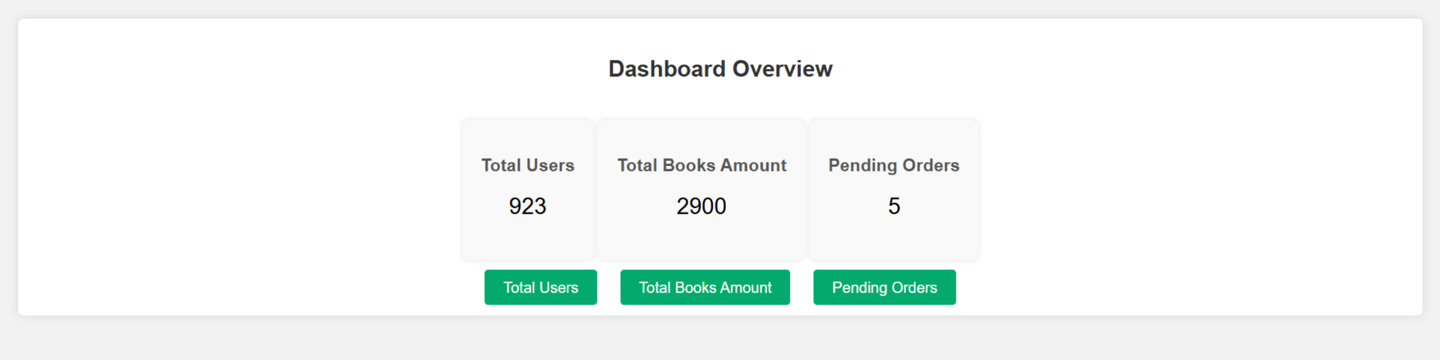
Expiration date

00/00

CVC

XXX

- **Admin Interface**:
    - Build a dashboard for administrators to add, update, and remove books.



Dashboard Overview

Total Users	Total Books Amount	Pending Orders
923	2900	5

    - Implement a reporting section for generating sales reports.
- **Tools**: Use HTML, CSS, JavaScript, with frameworks like React or Vue.js.
- **Mobile Responsiveness**: Ensure the UI works on both desktop and mobile devices (using CSS media queries or a responsive framework like Bootstrap).

5.3 Back-End Development

- **Task**: Develop the back-end API to handle business logic and database operations.
    - **Customer Routes**:
        - Registration (/register)
        - Login (/login)
        - Book browsing (/books)
        - Shopping cart management (/cart)
        - Checkout (/checkout)
    - **Admin Routes**:
        - Book management (/admin/books)
        - Report generation (/admin/reports)
    - **Authentication**: Use JWT (JSON Web Tokens) for securing API endpoints.
    - **Tools**: Use Python (Django/Flask), Node.js (Express), or Java (Spring Boot).

5.4 Payment Gateway Integration

- **Task**: Integrate a third-party payment gateway to handle transactions securely.
    - **Steps**:
        1. Set up an account with Stripe or PayPal.
        2. Use the payment gateway's API to process payments in the checkout route.
        3. Handle success and failure callbacks from the payment provider.
        4. Store successful payment information in the Orders table.
    - **Tools**: Use the official Stripe/PayPal SDK for the chosen back-end technology.

5.5 Security Implementation

- **Task**: Secure the system for safe transactions and data protection.
    - **Steps**:

1. Use HTTPS to encrypt communications.
2. Secure the payment process using the payment gateway’s best practices (e.g., PCI compliance).
3. Encrypt sensitive customer information, such as passwords (using bcrypt or similar).
4. Implement input validation and protect against SQL injection or cross-site scripting (XSS).

5.6 Testing

- **Task**: Perform comprehensive testing on the system.
  - **Unit Tests**: Test back-end business logic (e.g., cart calculations, checkout).
  - **Integration Tests**: Ensure front-end and back-end work together correctly.
  - **Load Testing**: Simulate up to 1,000 concurrent users to test scalability.
  - **Payment Gateway Testing**: Use sandbox mode of payment gateways to simulate transactions.

5.7 Deployment

- **Task**: Deploy the system on a cloud platform for production.
  - **Steps**:
    1. Set up a server on AWS, Heroku, or DigitalOcean.
    2. Deploy the back-end API and front-end application.
    3. Set up the database in a managed service (AWS RDS, Heroku Postgres, etc.).
    4. Configure domain and SSL certificates for HTTPS.

5.8 Post-Deployment Monitoring

- **Task**: Monitor the system post-launch.
  - Use logging and monitoring tools (e.g., AWS CloudWatch, Loggly) to track errors and performance.
  - Ensure that the system is responsive and scalable based on traffic.

6. Milestones

1. **Week 1**: Database design and schema setup.
2. **Week 2**: Front-end development of customer features (browse, cart, checkout).
3. **Week 3**: Back-end API development and integration with the database.
4. **Week 4**: Admin interface and reporting tools.

5. **Week 5**: Payment gateway integration and security features.
6. **Week 6**: Testing and bug fixing.
7. **Week 7**: Deployment and monitoring setup.

7. Gathering Results

- **User Feedback**: Gather feedback on the usability of the customer and admin interfaces.
- **Performance Metrics**: Monitor the system's performance under high traffic to ensure it scales.
- **Sales Tracking**: Use the admin reports to track if sales data are being captured correctly.

8. Conclusion

The OUSLBookStore web-based system is designed to address the current limitations faced by the bookstore in managing online book sales. Through this project, a comprehensive platform was developed to provide a seamless experience for both customers and administrators. The system enables customers to create accounts, browse books, add them to a shopping cart, and complete secure online payments. The administrative interface provides tools for managing book inventory and generating sales reports, improving the efficiency of bookstore operations.

With a user-friendly design, secure payment integration, and the ability to handle up to 1,000 users simultaneously, the system ensures a robust and scalable solution that meets the bookstore's business goals. The implementation focuses on security, mobile responsiveness, and scalability, ensuring that the system is ready for production use and can grow alongside the business.

****

Team members:

- **B.M.K.U.W. Hemantha Bandaranayake – 123582243**

- **R.W.M.S.B Wickramasingha – 223593522**

- **Anton Ignacious Andrew – 723577003**

- **W.G.K.Kavinda - 623593635**