# A PROJECT ON

**“ONLINE BOOK STORE”**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF

DIPLOMA IN ADVANCED COMPUTING FROM CDAC



#### SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

Hinjawadi

**SUBMITTED BY:**

Dipam Mokal

Vaibhav Yadav

Piyush Bokade

Shivansh Asati

Additya Dapke

**UNDER THE GUIDENCE OF:**

Nilesh Ghule

Snehal Jadhav

Sunbeam Institute of Information Technology, PUNE.

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Dipam Mokal

Vaibhav Yadav

Piyush Bokade

Shivansh Asati

Additya Dapke

PGDAC March 2023 Batch,

SIIT Pune

**CERTIFICATE**

This is to certify that the project work under the title ‘Online Book Store’ is done by Vaibhav Yadav, Dipam Mokal, Piyush Bokade, Shivansh Asati, Additya Dapke in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

**Nilesh Ghule Mr. Yogesh Kolhe**

**Project Guide Course Co-Coordinator**

Date: 01/09/2023

**TABLE OF CONTENTS**

1. Introduction To Project……………………………………1
2. Functional Requirement……………………………………2-9
   1. User Account…………………………………………..2-6
   2. Admin Account………………………………………...7-9
3. Design……………………………………………………….10-22
   1. Class Design……………………………………………10
   2. Database Design………………………………………..11-15
   3. User Interface…………………………………………..16-22
      1. Registration And Login Interface………………….16-17
      2. Admin Home Page…………………………………18-19
      3. Customer Home Page………………………………20
      4. Shopping Cart Page……………………………...…21-22
4. Coding Standards……………………………………………..23
5. Test Report…………………………………………………..24-26
   1. Development Tools And Technology……………..……27
   2. System Testing…………………………………….……28
   3. Testing Method…………………………………………28
      1. Stock Module………………………………..……..28
      2. Web Module………………………………..………29
6. Reference…………………………………………...………..30

# INTRODUCTION TO PROJECT

The web based “Online Book Store” project is an attempt to stimulate the basic concepts of Web Based e-bookstore website The objective of this project is to develop an e- book store where books can be bought from the comfort of home through the Internet. An online book store is a virtual store on the Internet where customers can browse the catalog and select books of interest. The selected books may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address.

Customers may shop for books online using a web browser thanks to the Online Book Store Project. A customer can create an account, log in, sort books by category, add books to a shopping basket, and pay their bill using their credit card information. When compared to a regular user, the Administrator will have more options. He can edit the author, publisher, book categories, book details, and member information, as well as confirm an order.

The following are the three main components of the software:

1. Implementation of a new user registration and login process.

2. Allow the user to select any book.

3. Allow the user to purchase books.

The motivation to create this project has many sources

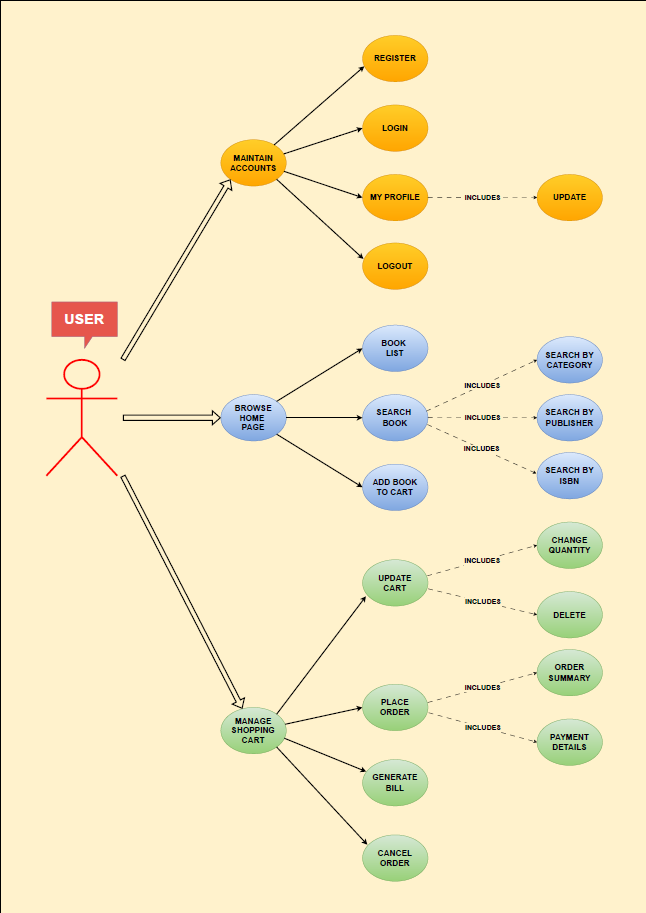
• Interest to develop a good user friendly website with many online transactions using a Database.

• To increase my knowledge horizon in technologies like React, Spring Boot, Mysql

• To gain good experience in web before joining in a full time job.

# 2.FUNCTIONAL REQUIREMENTS

## **2.1 User Account**

****

**Use Cases:**

**Browse Books**

**1) Search for a Book**

• **Purpose**: If the user wants to perform an advanced search he can search for a

book of his choice by selecting category, title, author . Then a

select query is used to retrieve data from the database and display the selected

information.

• **Actor:** User

• **Input:** The user will select a category and enter title, author, and price range in a

text box provided.

• **Output:** The system will display the books which matches the selected search

criteria. A dataset is created as a result of select query. Later the dataset is binded

to the data repeater to display the selected data.

**3) Give rating to a book**

• **Purpose**: If the user wants to give rating according to his opinion for a book he

can select either Excellent, Very good, good, regular or deficient. The final rating of a book will depend on all the individual user rating.

• **Actor:** User

• **Input:** The user will select a rating based on his opinion.

• **Output:** The system will display the rating of a book and the total number of votes received. Below is the display for various rating.

\*\*\*\*\* Excellent

\*\*\* Good

\*\* Regular

\* Deficient

**Maintain Account**

**1) Register**

• **Purpose**: If the user doesn’t have an account then he will be asked to register.

• **Actor:** User

• **Input:** The user will enter details in the registration form according to the

required fields. The fields include

1. First Name

2. Last Name

3. Email

4. Password

5. Mobile Number

• **Output:** After registration the user will be directed to the main home page.

**2) Login**

• **Purpose**: If the user wants to get access to all the functionalities of Online Book

Store he should login using his username and password.

• **Actor:** User

• **Input:** The user will enter his email and password.

• **Output:** If it is a successful login the user will be directed to the main home page.

Else if the user enters invalid information he will be asked to check the entered

information.

**3) Update Profile**

• **Purpose**: If the user wants to change his personal account information then he can

update his selected fields and the entire data will be updated in the data base

through an update query.

• **Actor:** User

• **Input:** The user will update his account information.

• **Output:** The system will update the entered information in the database using an

update query.

**4) Logout**

• **Purpose**: If the user wants to end his session and sign out of the website then he

can use the logout option.

• **Actor:** User

• **Input:** The user will click the logout button.

• **Output:** The user’s account session comes to an end and he should login again if

he wants to enter into the website.

**Manage Shopping Cart**

**1) Place an order**

• **Purpose**: If the user wants to purchase a book then he can place an order by

selecting the add to shopping cart button and entering the quantity required under

the book description.

• **Actor:** User

• **Input:** The user will enter the quantity required and click the add to shopping cart

button.

• **Output:** The order will be added to the user’s shopping cart

**2) Update Shopping Cart**

• **Purpose**: If the user wants to change the quantity of a book or change a book then

he can update his shopping cart.

• **Actor:** User

• **Input:** The user will click the details button in the shopping cart summary to edit

and update his order details..

• **Output:** The updated order details are reflected in the shopping cart summary.

**3) View Shopping Cart**

• **Purpose**: If the user wants to view the items he added to the shopping cart then he

can click the shopping cart link at the top of the page.

• **Actor:** User

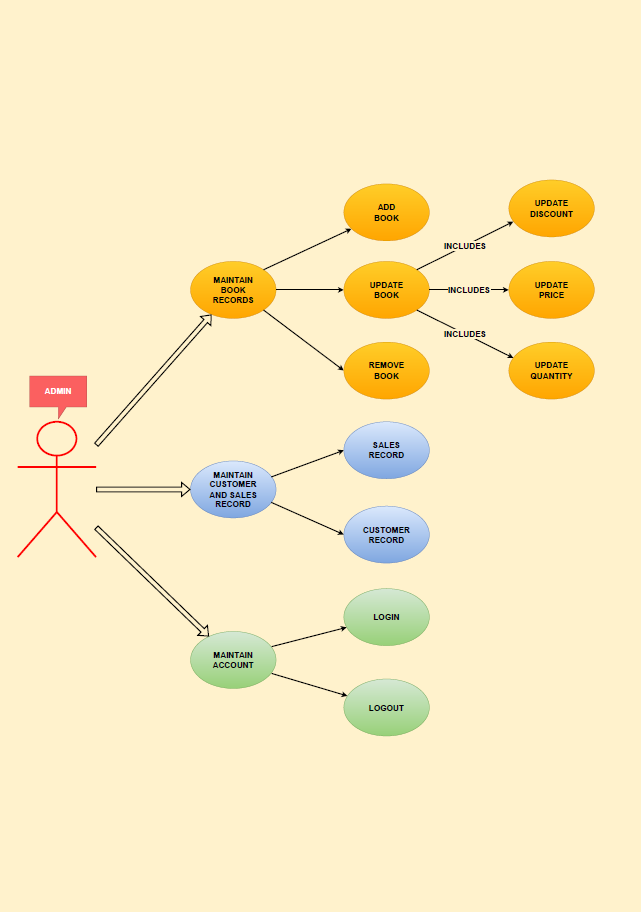
• **Input:** The user will click the shopping cart link at the top of every page.

• **Output:** The user’s shopping cart summary will be displayed in the form of a

tabular format with all the books and their quantity. A total cost of all the items is

also displayed at the bottom.

## **2.2 Admin Account**

****

**Use Cases:**

**1) Login**

• **Purpose**: If the Administrator wants to get access to all the functionalities of

Online Book Store he should login using his username and password.

• **Actor:** Administrator

• **Input:** The Administrator will enter his username and password.

• **Output:** If it is a successful login the Administrator will be directed to his menu

page. Else if the Administrator enters invalid information he will be asked to

check the entered information.

**2) Add or Delete Book**

• **Purpose**: If the Administrator wants to add or delete a book then he can insert or

delete a book using his administration rights and the book table will be updated in

the database.

• **Actor:** Administrator

• **Input:** If the Administrator wants to add a book the he should click the insert link

button in the book page and fill the following fields related to the book.

1. Title

2. Author

3. Price

4. Description

If he wants to delete a book he can click the delete button to remove it from the

database.

• **Output:** The updated books list will be displayed in the main home page under

their particular category.

**4) Manage Orders**

• **Purpose**: If the Administrator wants to add or delete an order then he can insert or

delete an order using his administration rights.

• **Actor:** Administrator

• **Input:** If the Administrator wants to add an order the he should click the insert

link button in the orders page else he can delete a particular selected order

**Output:** The updated orders list will be processed to the users.

**7) Logout**

• **Purpose**: If the Administrator wants to end his session and sign out of the website

then he can use the logout option.

• **Actor:** Administrator

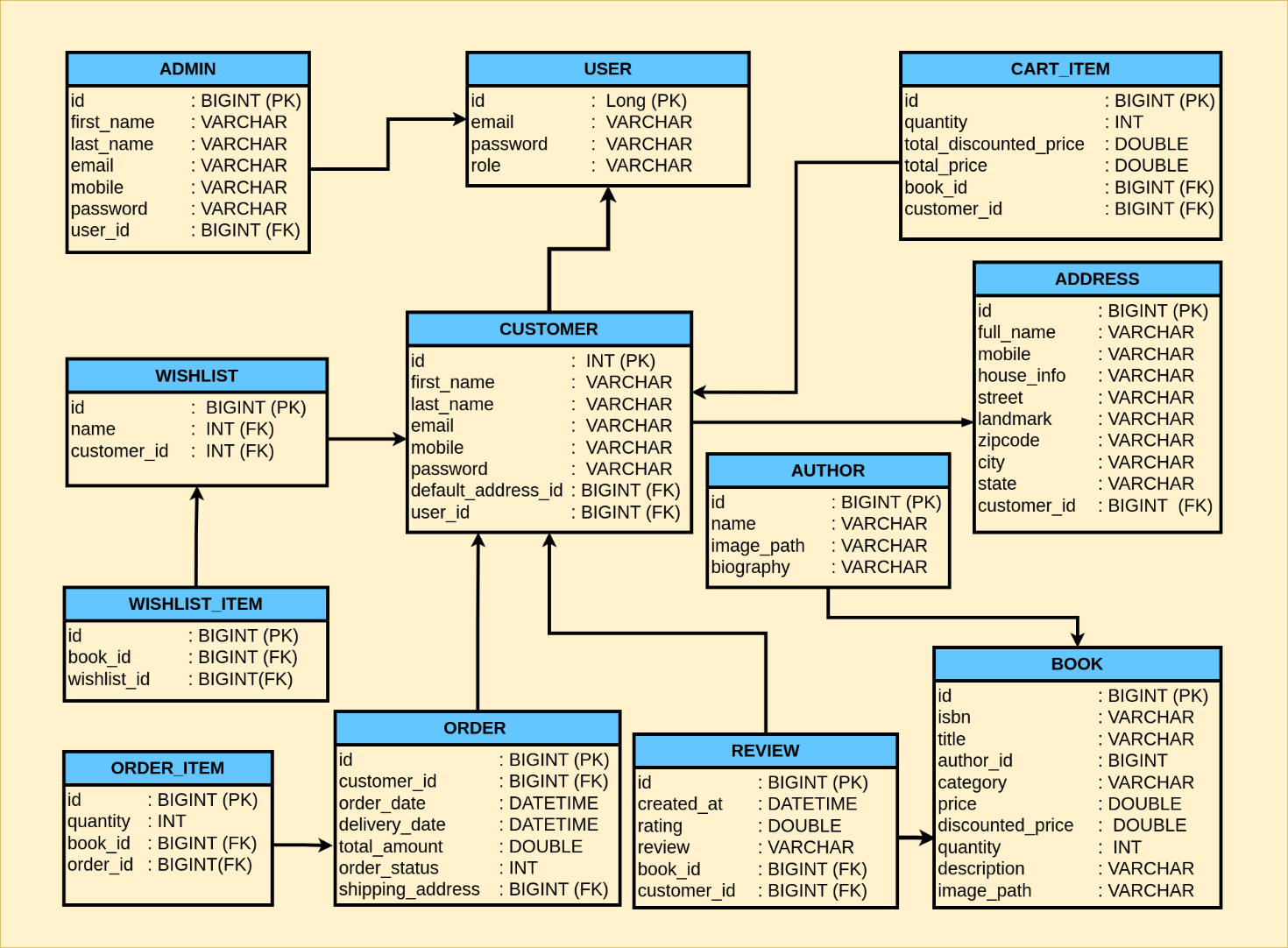
• **Input:** The Administrator will click the logout button.

• **Output:** The Administrator’s account session comes to an end and he should

login again if he wants to enter into the website.

# 3. DESIGN

## **3.1 Class Design**

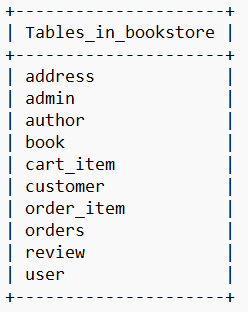


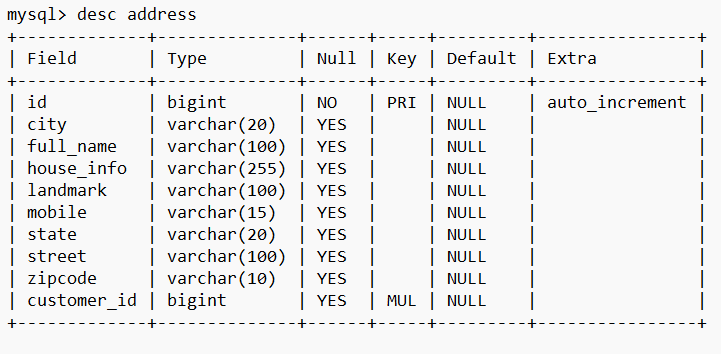
## **3.2 Database Design**

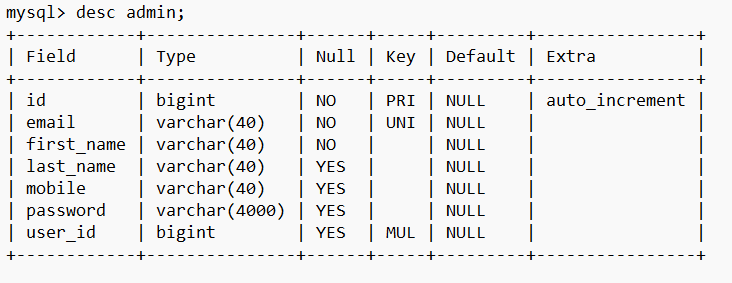
One of the most important components of any data-driven application, such as a Web-Based System, is the database. As a result, appropriate approaches are used to ensure the database's integrity. MySQL is used as the backend database in this project. MySQL is a database management system that is free and open source. The following are some of MySQL's features:

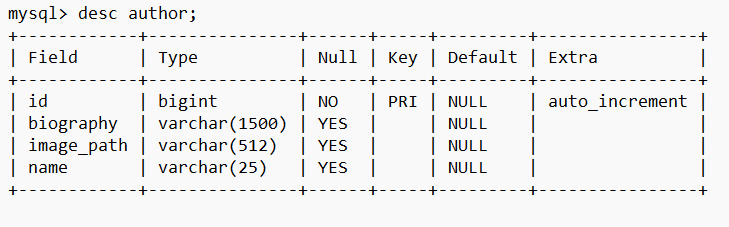
* MySQL is a database management system that uses a relational model. Rather than one big table, a relational database stores data in multiple tables. These tables can be linked together to make it easier to access and manage data.
* MySQL is a free and open-source database management system. Anyone can use and modify the database software to meet their specific needs. It's quick, dependable, and simple to use to enhance the level of performance.
* MySQL is a database engine that runs in several threads. A multithreaded application accomplishes multiple tasks at once, as if multiple instances of the application were executing at the same time.

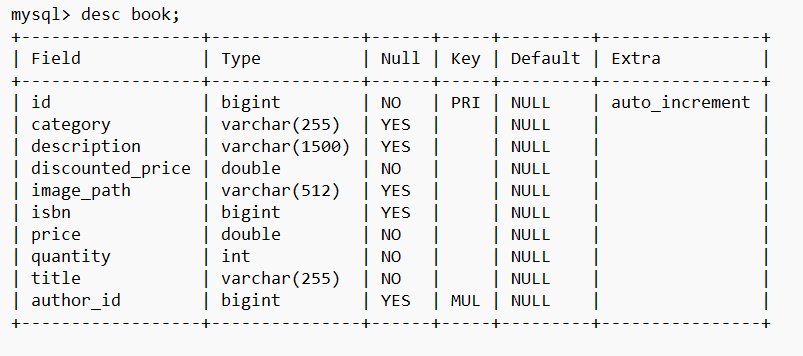
* MySQL provides a lot of advantages because it is multithreaded. Each incoming connection is managed by a single thread, with an additional thread that is always running to manage the connections. Multiple clients can perform read operations at the same time, but only one client can access the data being modified while writing. Despite the fact that the threads share the same process space, they operate separately, allowing multiprocessor machines to spread the thread across many CPUs as long as the host operating system supports multiple CPUs. Multithreading is a critical component for MySQL's performance goals. It is the foundation upon which MySQL is based.
* An ODBC driver is used to link the MySQL database to Java. Open Database Connectivity (ODBC) is a frequently used database access Application Programming Interface (API). The ODBC driver is a library that implements the ODBC API's functions. It handles ODBC function calls, sends SQL requests to the MySQL server, and then returns the results to the application. If necessary, the driver alters an application's request so that it is compatible with MySQL's syntax.

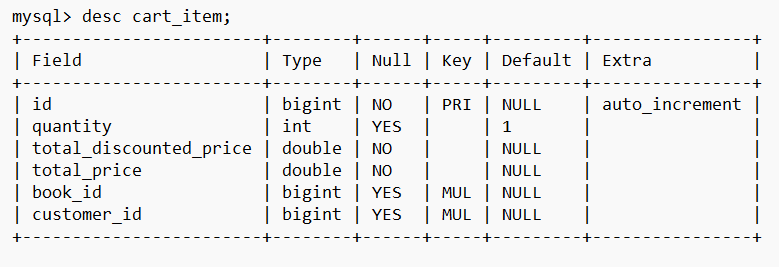




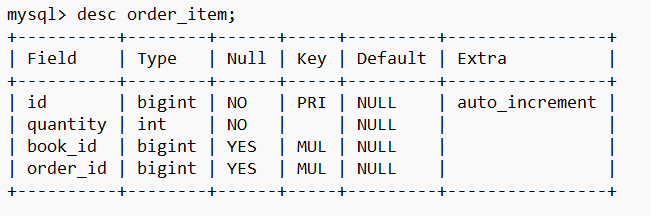


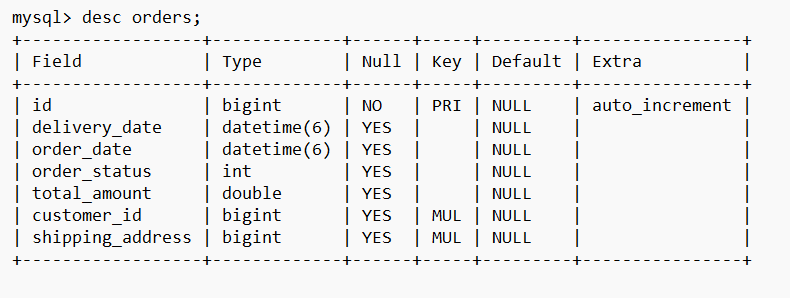


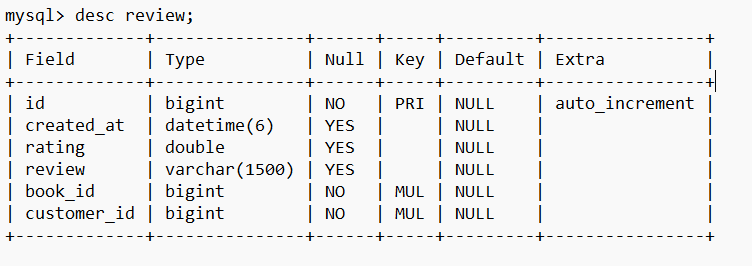


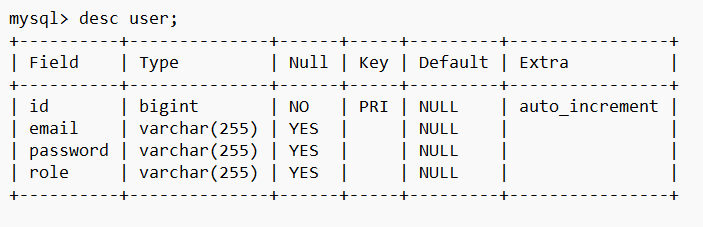


# 









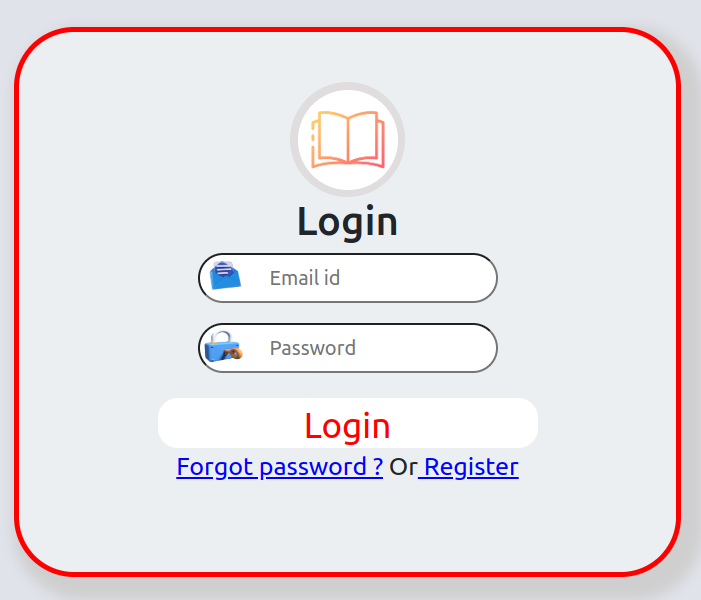
## **3.3 User Interface Design**

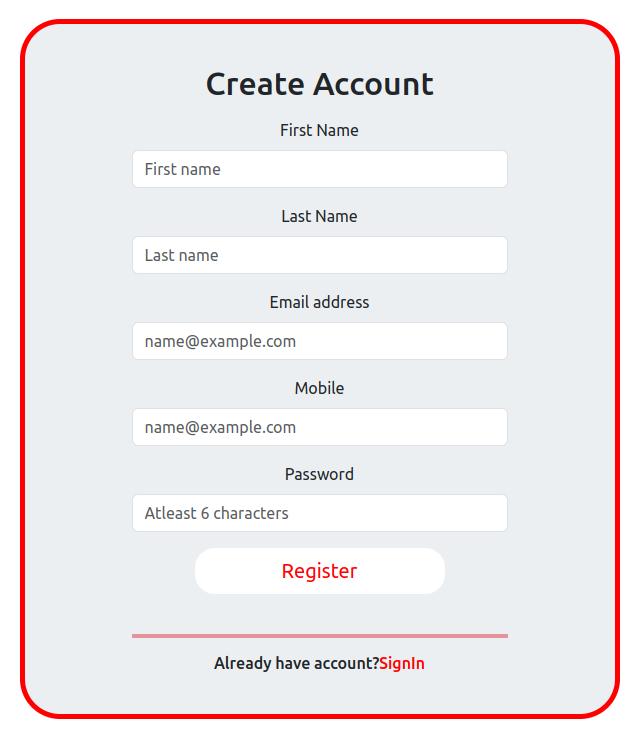
One of the most important factors in determining an application's user friendliness is its user interface. Because it is the component with which the user interacts. The following are some of the guidelines that we followed:

* Colors, text styles, component structure, and functionality, such as navigations, should all be consistent across all interfaces.
* All interface effects and dynamic changes should have a clear meaning for the user, such as links with a red pointer cursor for deletion or deletion warning, and so on.
* It should be simple for users to understand. Interfaces should be as basic as possible, with helpful features such as tool tips, popup messages, and notifications recommended.
* Error and confirmation messages, for example, should be consistent, straightforward, and free of technical terms.
* For interfaces such as forms, the data input process should be improved and as many errors as possible detected. It would be helpful to provide default values and watermarks, as well as to use lists and option buttons instead of text boxes for selecting data rather than typing.

### **3.3.1 Register and Login Interface**

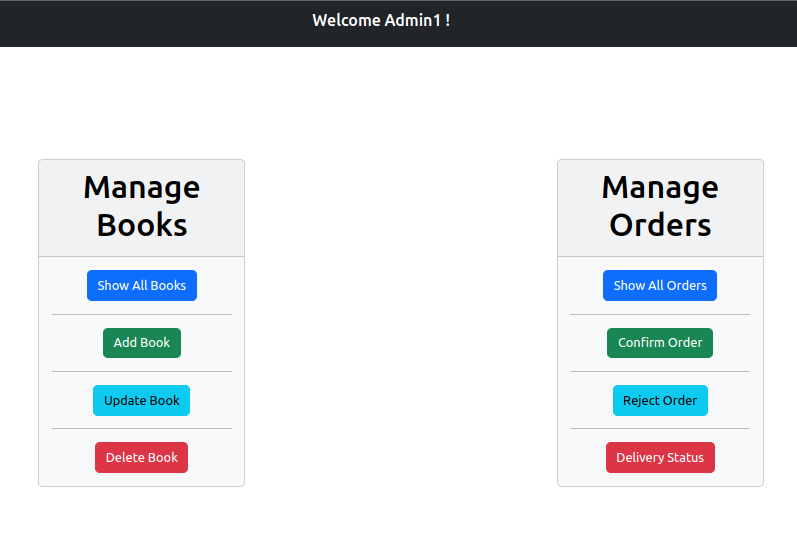
The system registers and login page, which is part of the developed system, is the primary interface for logging into the system. Both pages are the initial interface a user encounters in any computerized system. The main register and login interface for Web-Based Book Store System is shown in Figures

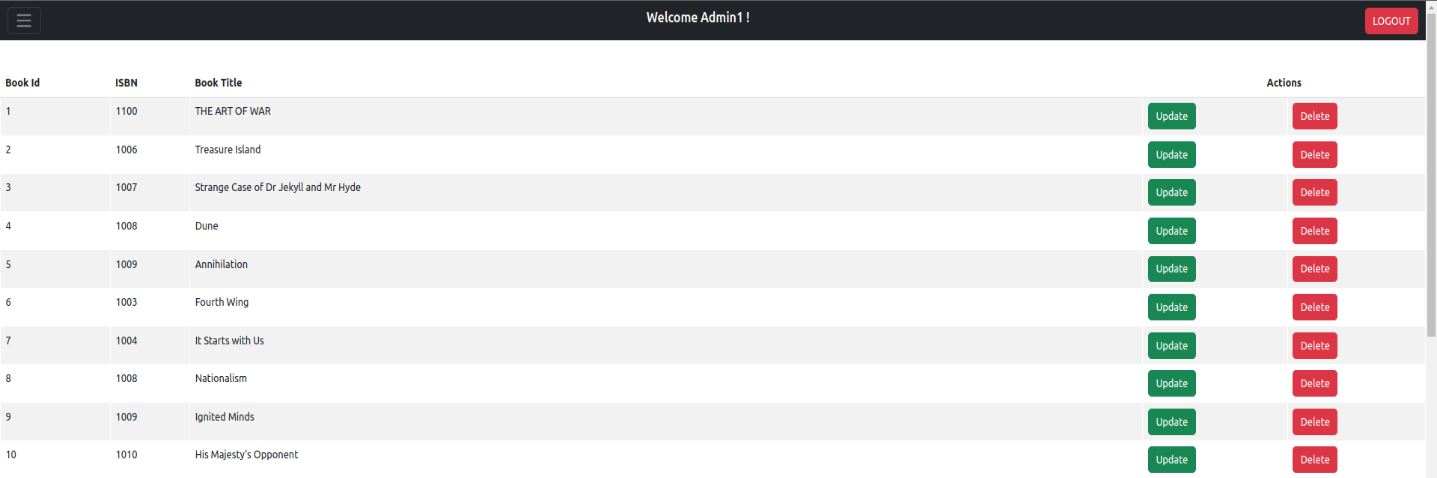


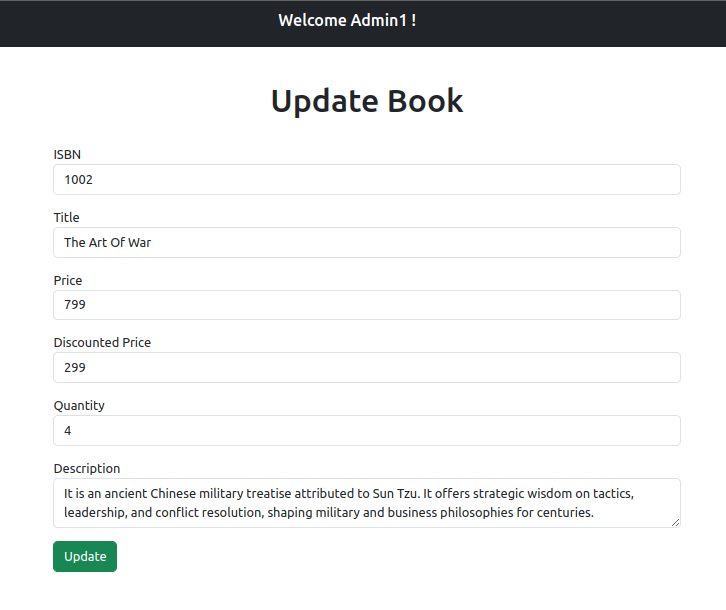


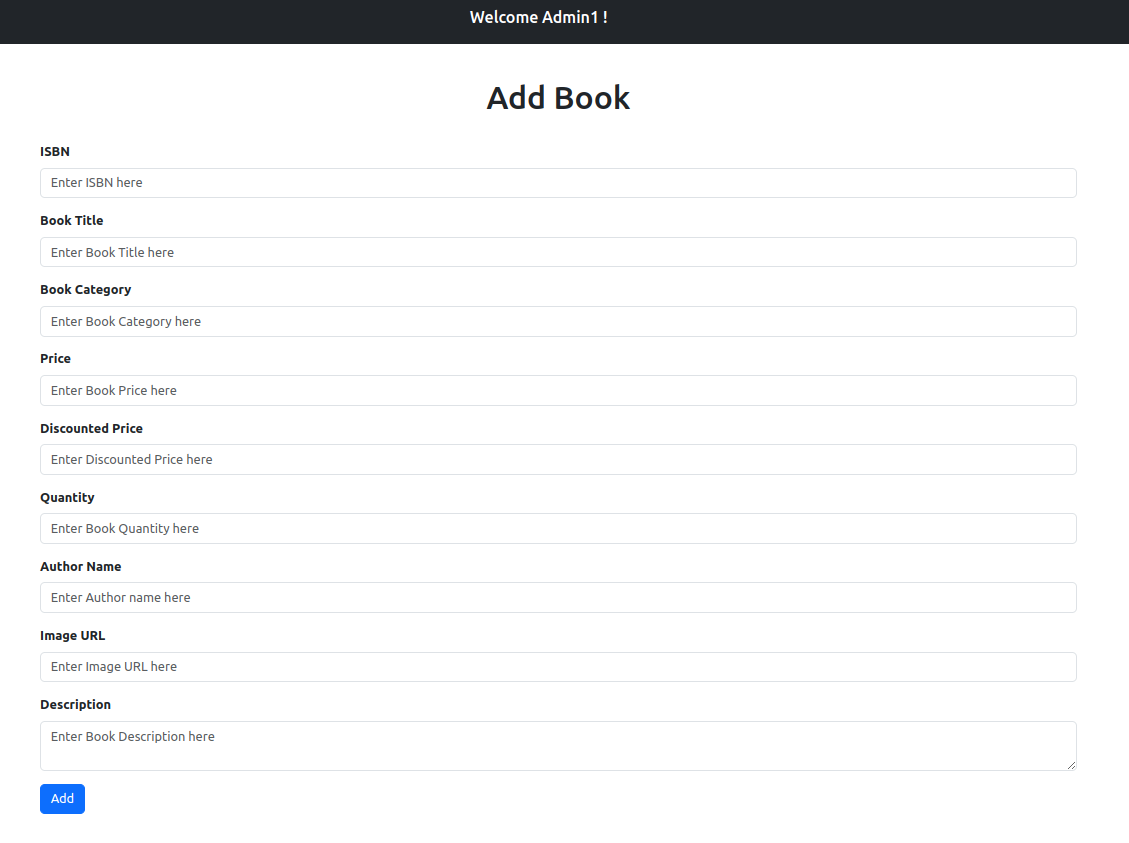
### **3.3.2 Admin Home Page**

Figure shows the home page of the admin's account. The final output was developed based on the research into other similar web-based system interfaces.



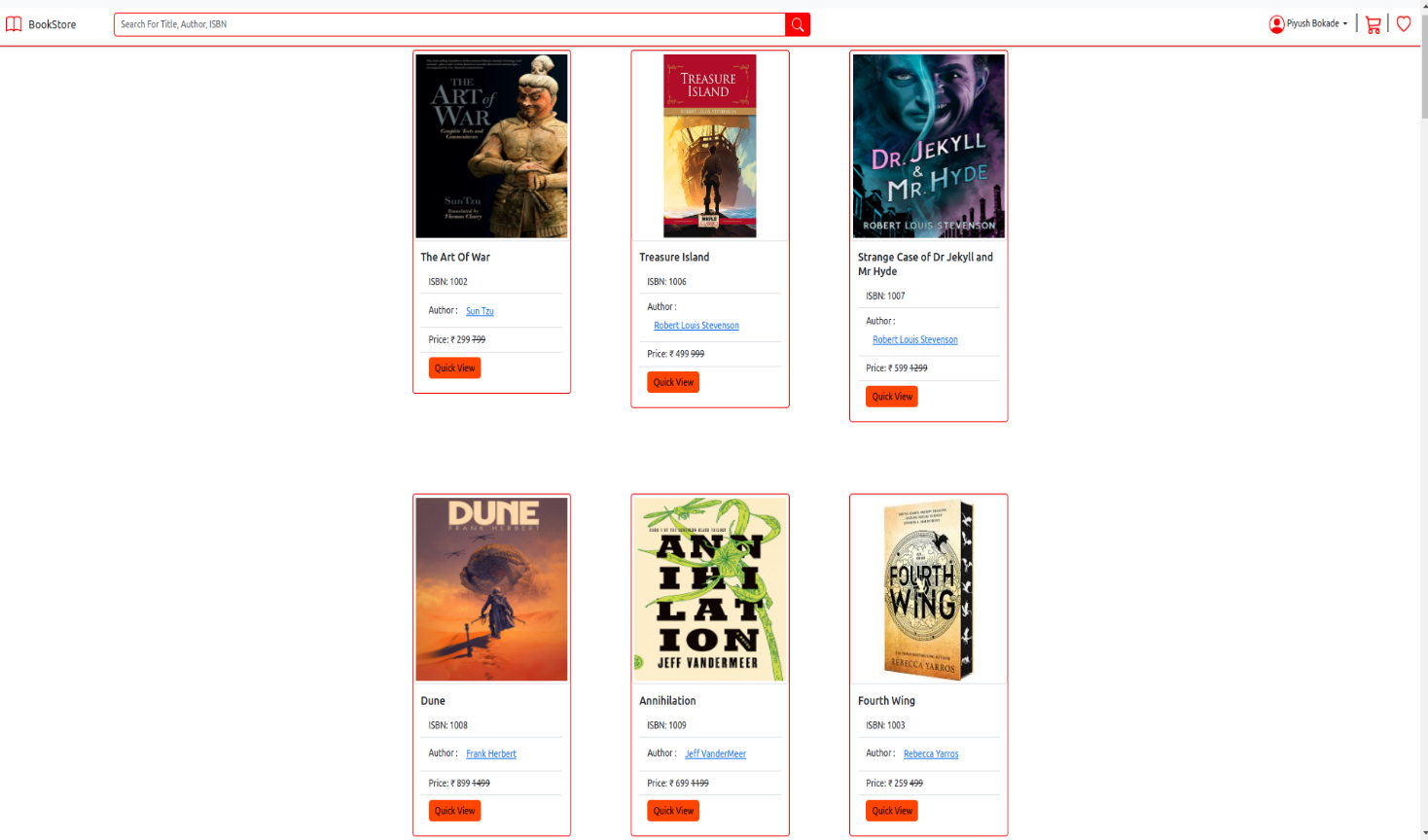


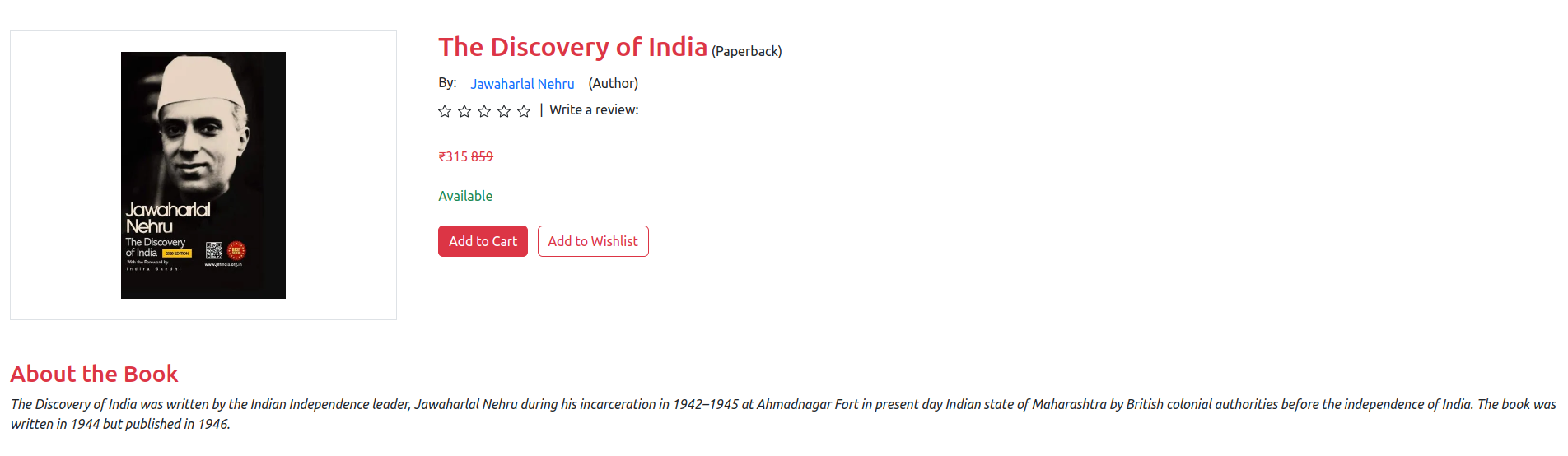




### **3.3.3 Customer Home Page**

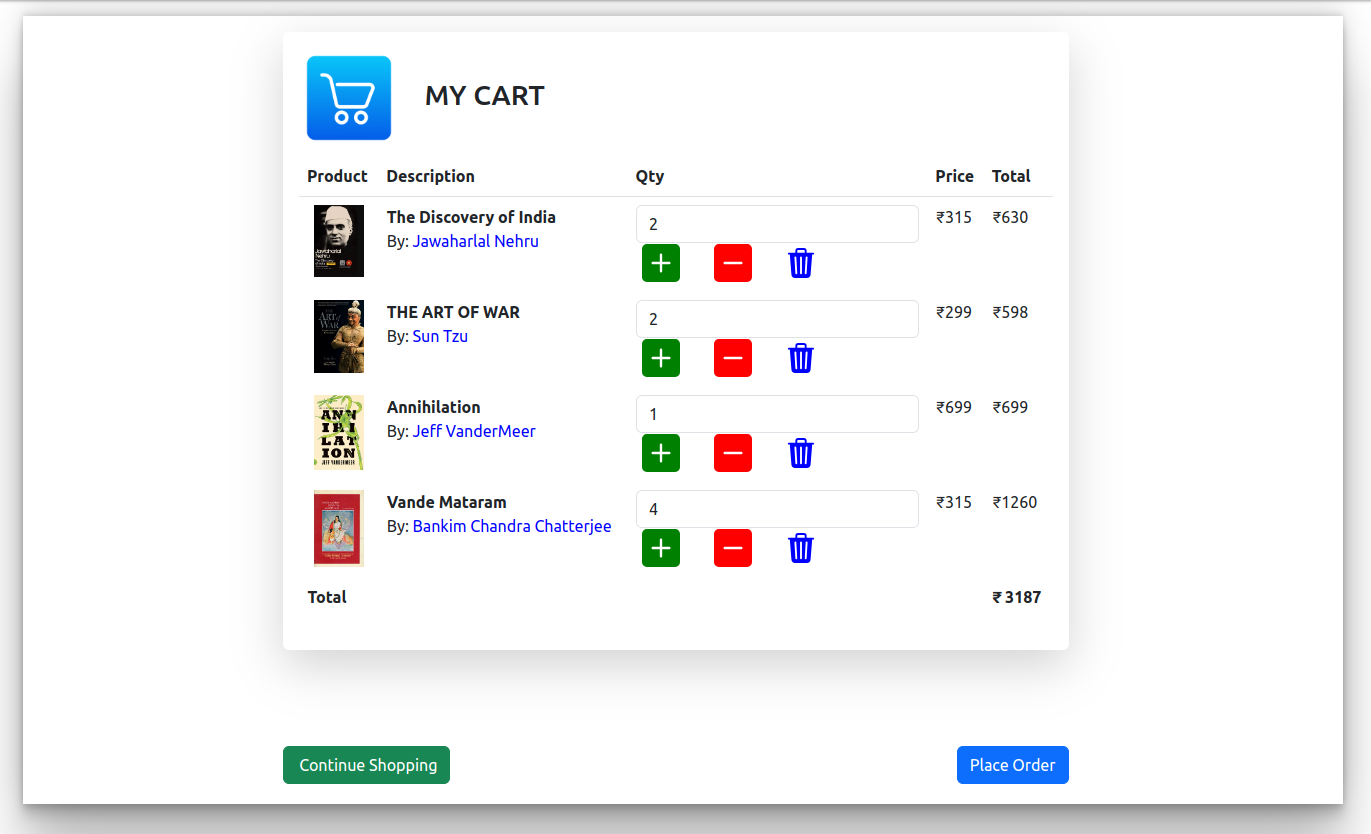
Since first impressions can influence how many people perceive your company, the web homepage is typically the first opportunity to hook a potential customer. The homepage of the website should be well-designed because it is the anchor that ties the rest of the website together.

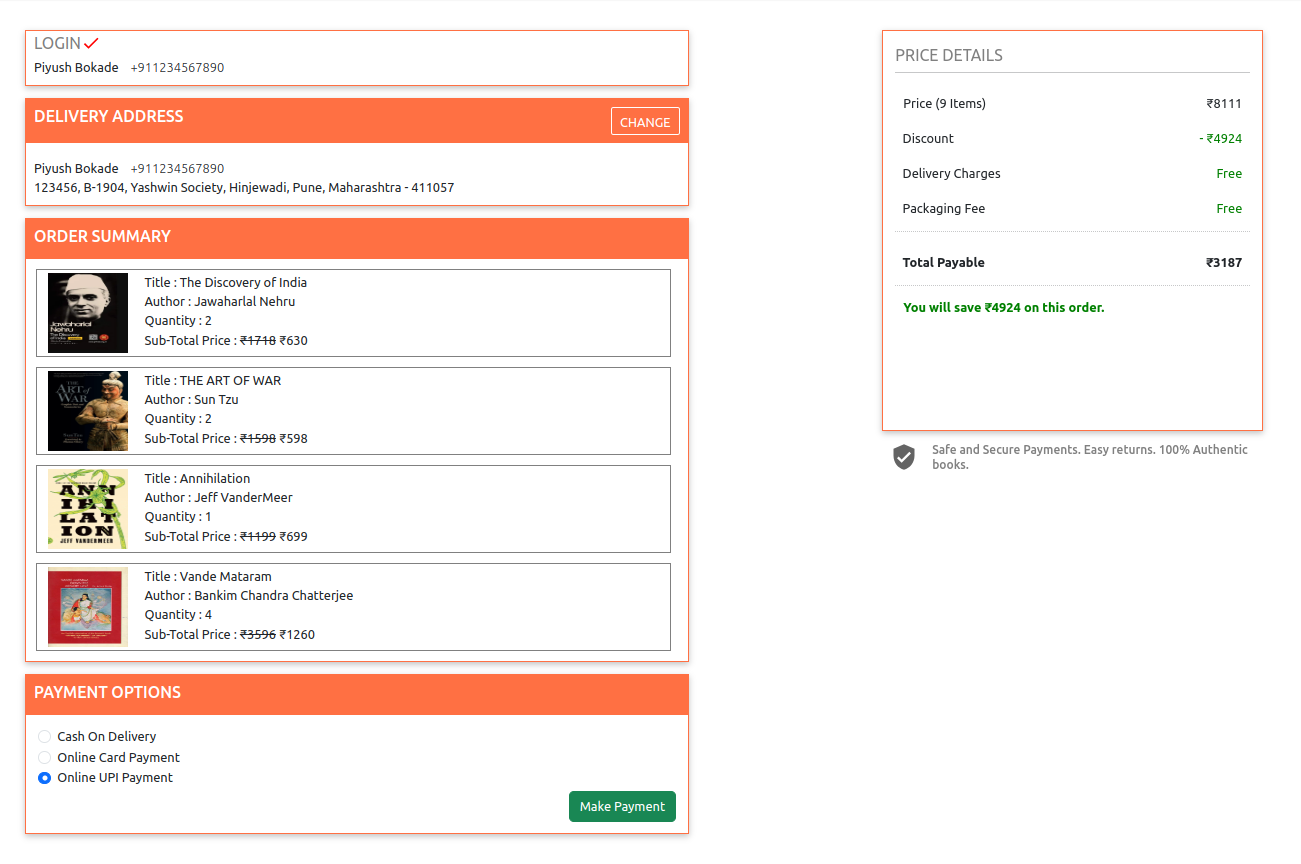


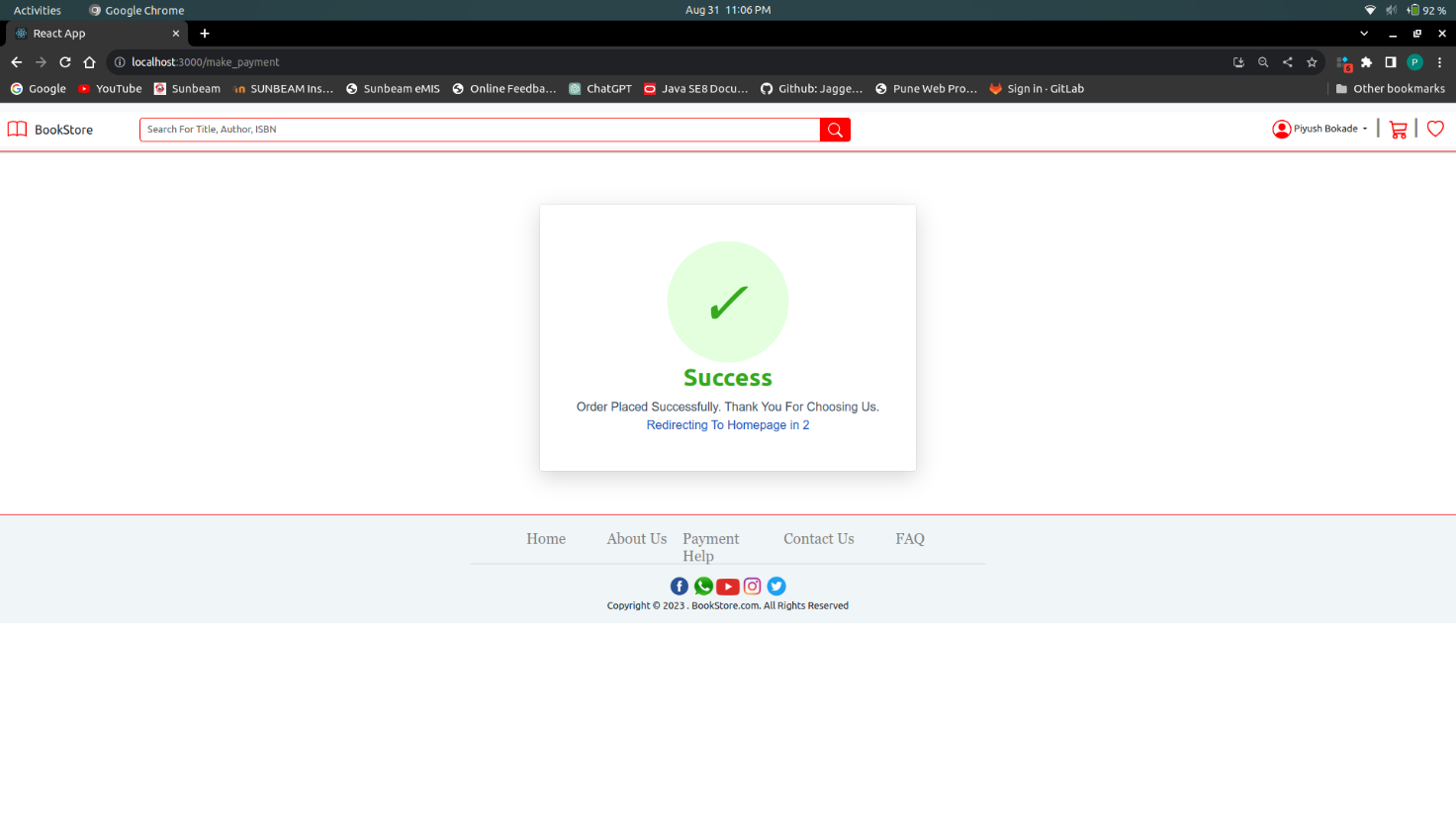


### **3.3.4 Shopping Cart Page**

The aim of this program is to give users with an online store where they would purchase books from the comfort of their own homes. For this purpose, a shopping cart is implemented. The customer can choose the books they want, add them to their shopping cart, and pay for them with a debit or credit card. The user's order will be shipped based on the delivery method selected at the time of purchase. Figure 10 illustrate the main shopping cart page.







# 4. CODING STANDARDS IMPLEMENTED

### Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | AdminController, BookController,  UserController | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type prefixes like ‘C’ for class. |
| Method | Camel | getDetails, updateCart | Methods should use **verbs** or verb phrases. |
| Parameter | Camel | firstName, bookId | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios. |
| Exception Class | Pascal with "Exception" suffix | ResourceNotFoundException |  |

**5. TEST REPORT**

The implementation phase entails the development of an executable program based on the design created during the design phase. Selecting programming languages, additional tools and technologies like as frameworks, selecting hardware platforms, and coding the system are some of the main activities carried out during this phase.

The system is evaluated against multiple factors such as functional and non-functional requirements in order to guarantee that it is working properly and meets all of its specified standards during the evaluation phase of the software development process. This chapter discusses the web-based system testing methodologies, test plan, and test cases, as well as test data and outputs and acceptance testing.

This chapter covers the application's implementation environment, the framework, the development tools used, the application's structure, and an explanation of the key code segments.

**5.2 Development Tools and Technologies**

* STS for backend coding and Visual Studio Code for frontend coding.
* Apache Tomcat for local host and server connectivity with MySQL database.

The Web-Based Book Store System was built using the following technologies:

* The web pages were developed using Visual Studio Code. The new features that were added in the new version are developed to increase the semantics of web pages. This is the most basic web-related language, and it assists in maintaining a clear and conscious system structure.
* The pages were styled with CSS3. It makes it possible to create a unified design for the entire system in an easy-to-manage manner.
* When developing the system, Spring Boot was employed as the server-side object-oriented programming language.
* The application's database was implemented using MySQL.
* The client-side scripting was implemented with JSX. Especially for the form validation process on the client side.
* Bootstrap – It’s a free framework for developing websites and online apps. It includes design templates that are based on HTML and CSS. With a single code base, Bootstrap easily and efficiently scales the application. Bootstrap speeds up and simplifies frontend web development.

**5.3 System Testing**

The planning of a testing phase is critical for both the development and completion of a system. The test plan should be able to test the overall system's functionality. By extensively testing a system, it is possible to identify and fix errors that occur as a result of the system. Multiple test scenarios were used to evaluate the implemented system. The test plan has continued to test the system units since the development began. Upon completing the system, it was thoroughly evaluated to determine whether it could execute as planned. As a result, this testing stage assisted in the early detection of errors.

Following the system unit testing, integration testing was conducted, which allowed for the detection of errors. To evaluate the functionality of the fully developed web-based application, system testing was performed as the final stage.

**5.4 Testing Methods**

In software engineering, a number of software testing methodologies and styles have been introduced to test various features of various systems. The major testing methodologies utilized widely in the software testing process are Black Box testing and White Box testing. In Black Box testing, the function is evaluated by comparing the output to the input without taking into account the inner structure of the function, whereas in White Box testing, each of the function's logical routes is evaluated while taking into account the inner structure of the function.

Various styles of testing were utilized at various stages of the web-based system's development. During the design phase, unit testing was performed on each individual function of each module to check that they were correct. Following the completion of unit testing, integration testing was performed to check that the functionality of the modules when they interacted with one another was correct. Following the conclusion of the integration testing, system testing was performed on the entire system to guarantee its reliability.

**5.4.1 Stock Module**

|  |  |  |  |
| --- | --- | --- | --- |
| The following table list a relevant test case for the Stock Module. Test No. | Test Description | Testing Procedures | Expected Result |
| 1. | Add new book | Go to Products, add details and click ‘Add Book’ button. | Book is added and confirmed. |
| 2. | Edit book | Click the ‘Update’ button in front of the particular book record, edit relevant fields and confirm. | Update the details and confirm. |
| 3. | Delete book | Click the ‘Delete’ button in front of the particular book record and confirm. | Book is deleted from the record. |

**5.4.2 Web Module**

|  |  |  |  |
| --- | --- | --- | --- |
| The following table lists the most relevant test cases for the Web Module. Test No. | Test Description | Testing Procedures | Expected Result |
| 1. | Main menu navigation | Click each menu item of the main menu. | Load the relevant page. |
| 2. | Customer / admin login | Enter id and password of the customer / admin. Click login. | customer / admin is able to login. |
| 3. | Add books to the shopping cart | Clicks ‘Add to Cart’ button of the particular product. | Update the shopping cart. |
| 4. | View shopping cart | Click the shopping cart button. | Display the shopping cart. |
| 5. | Remove books from the shopping cart | Click delete button in front of the particular book. | Remove the particular book from the shopping cart. |
| 6. | Clear shopping cart | Click ‘Delete All’ button of the shopping cart area. | Remove the entire books in the shopping cart. |
| 7. | Place order | Click ‘Proceed to Checkout’ button. | Add the new order entry, redirected to the orders page & display confirmation. |

**6.REFERENCES:**

<http://www.google.com>

[http://](http://www.xml101.com:8081/xml/)www.amazon.com

http://www.flipkart.com

http://www.bookswagon.com

<http://www.w3.org>

<http://www.npmjs.com>

https://getbootstrap.com/