

**MAHATMA GANDHI MEMORIAL COLLEGE UDUPI-576102**

Accredited by NAAC with “A+” Grade (CGPA 3.36)

**PROJECT REPORT ON**

**“Online Bookstore-BookQuest Plaza”**

**DEVELOPED BY**

**Avinya Reg. No.:** U05MG21S0257

**Dhwani D Karkera Reg. No.:** U05MG21S012133

**Harshini V Amin Reg. No.:** U05MG21S0046

**Under The Guidance Of**

Prof. Rekha Naveenchandra

Department of Computer Science

Submitted to the Mangalore University in partial fulfilment of the Award Bachelor in Computer Application

**MANGLORE UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE**

**MAHATMA GANDHI MEMORIAL COLLEGE**

**UDUPI-576102**

**2023-2024**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Title** | **Page No** |
| 1 | Introduction | **6** |
| 2 | Synopsis | **7-8** |
| 3 | Importance of project | **9** |
| 4 | Methodology of project | **10** |
| 5 | Objectives of project | **11** |
| 6 | Analysis and Interpretation |  |
| 6.1 System Design | **11** |
| 6.1.1 Context Flow Diagram | **11** |
| 6.1.2 Data Flow Diagram | **12** |
| 7 | Table |  |
| 7.1 Database Design | **15** |
| 7.2 Entity Relationship Diagram | **16-18** |
| 8 | Conclusion | **19** |
| 9 | Learning outcomes of the project | **19** |
| 10 | Code | **20-42** |
| 11 | Photography | **43-50** |
| 12 | Test Cases | **51-56** |
| 13 | Bibliography | **57** |

**CERTIFICATE OF COMPLETION**

This is to certify that the project entitled “Online Bookstore” is a web-based project done by Avinya (U05MG21SO257), Dhwani D Karkera (U05MG21S0121) and Harshini V Amin (U05MG21S0046) of final year BCA, under the guidance and supervision of Prof. Rekha Naveenchandra, Lecturer of Department of Computer Science, Mahatma Gandhi Memorial College, Udupi, towards the partial fulfilment of Bachelor’s degree of Computer Application.

**Signature of HOD**  **Signature of Guide**

**Signature of External Examiner**

**DECLARATION BY STUDENTS**

We, the final year students of BCA, hereby declare that the project report entitled “Online Bookstore” is a website-based project carried out by us under the guidance and supervision of Prof. Rekha Naveenchandra, Lecturer of Department of Computer Science, Mahatma Gandhi Memorial College, Udupi.

We assure that this project work embodied the result of our original work and contents of the project have not been submitted to anybody else for the award of degree. This project is purely of academic interest.

We have followed the guidelines provided by the university in writing the report.

Date: June 05,2024

**Signature of the Students**

Avinya

Dhwani D Karkera

Harshini V Amin

**ACKNOWLEDGEMENT**

Behind every achievement, there is a sea of gratitude to those who have activated this project. The magnitude of this project demanded the co-operation, guidance and help of many people. We have been fortunate enough to have this in the entire task of completion of our project on “**Online Bookstore**”.

We would like to thank our principal **Prof. Dr. Lakshmi Narayan Karanth** for giving us opportunity to carry out our project.

We thank **Dr. M. Vishwanath Pai**, a source of inspiration and encouragement, head of the Department of Computer Science, Mahatma Gandhi Memorial College, Udupi for having permitted us to carry out our project work.

We are extremely grateful to express our overwhelming gratitude to our guide **Prof. Rekha Naveenchandra**, Lecturer of Computer Science Department, Mahatma Gandhi Memorial College Udupi for giving us valuable guidance to undertake this project.

Last but not the least, we are indebted to all teaching and non-teaching staff members, Mahatma Gandhi Memorial College Udupi for making this project successful.

Thanking you

Avinya

Dhwani D Karkera

Harshini V Amin

**1.INTRODUCTION:**

BookQuest Plaza is an online bookstore website aimed at providing a convenient platform for book enthusiasts to explore and purchase their favourite titles from a wide range of genres. With a user-friendly interface and robust features, Book Haven aims to enhance the online book shopping experience.

An online bookstore project acts as a central database containing various books in stock along with their title and cost. This project is a website that acts as a central book store. This web project is developed using html\CSS, JavaScript and php as the front end and MySQL as a backend. The MySQL database store various book related details.

A user visiting the website can see a wide range of books arranged in respective categories. The user may select desired book and view its price. The user may even search for specific books on the website. Once the user selects a book, he then has to fill in a form and the book is booked for the user.

**Implementation Overview**

**1. HTML:** Structures the content of the website, including forms, calculators, and trackers.

**2. CSS & Bootstrap:** Provides styling and ensures the website is responsive and visually appealing.

**3. JavaScript:** Adds interactivity, such as real-time calculations in the BMI calculator.

**4. PHP:** Handles server-side operations, such as form submission, data validation, and interaction with the database.

This integration of technologies ensures that BookQuest Plaza is robust, scalable, and easy to use, making it an essential tool for anyone looking to read new books.

Bottom of Form

**2. Synopsis**

**Introduction:**

BookQuest Plaza is an online bookstore website aimed at providing a convenient platform for book enthusiasts to explore and purchase their favourite titles from a wide range of genres. With a user-friendly interface and robust features, Book Haven aims to enhance the online book shopping experience.

**Title:** BookQuest Plaza - Online Bookstore Website

**Objectives:**

* To create a seamless and intuitive online platform for browsing and purchasing books.
* To provide users with personalized recommendations based on their preferences.
* To streamline the ordering and delivery process for efficient customer service.
* To establish a community-driven platform for book reviews and discussions.

**Scope:**

The scope of the project includes developing a comprehensive website with features such as user registration, book categorization, search functionality, secure payment gateways, order tracking, and user interaction functionalities like ratings and reviews.

**Project Category:**

Web Application

**Tools / Platform:**

**Tools:** HTML/CSS, JavaScript, PHP, MySQL

**Platform:** Web Browser

**Hardware, and Software Requirements:**

**Hardware:** Standard web hosting server

**Software:** Web browser (Chrome, Firefox, etc.)

**Modules and Their Descriptions:**

**User Management:** Allows users to register, log in, and manage their profiles.

**Product Catalog:** Displays books categorized by genre, author, and popularity.

**Search Functionality:** Enables users to search for specific titles or authors.

**Shopping Cart:** Allows users to add, remove, and manage items in their cart.

**Payment Gateway:** Integrates secure payment options for seamless transactions.

**Order Management:** Enables users to track their orders and view order history.

**Review and Rating System:** Allows users to leave reviews and ratings for books.

**Admin Panel:** Provides administrators with tools to manage users, products, and orders.

**Project Type:**

This is a user-defined project developed to cater to the needs of book lovers. Book Haven is not associated with a specific industry or client.

**Limitations of the Project:**

* Limited availability of book titles compared to physical bookstores.
* Dependence on third-party payment gateways for secure transactions.
* Potential security vulnerabilities if not properly maintained.
* Future Scope and Further Enhancement:
* Integration of social media features for sharing book recommendations.
* Implementation of advanced recommendation algorithms based on user behavior.
* Expansion of product offerings to include e-books and audiobooks.
* Integration of a loyalty program to reward frequent customers.
* Mobile application development for increased accessibility.

**3. IMPORTANCE OF BOOKQUEST PLAZA**

The scope of the project includes developing a comprehensive website with features such as user registration, book categorization, search functionality, secure payment gateways, order tracking, and user interaction.

The scope of an online bookstore encompasses several aspects:

* Product Catalog: An online bookstore offers a wide range of books, including fiction, non-fiction, academic texts, self-help, and more. The scope extends to various genres and subjects.
* User Experience: Creating an intuitive and user-friendly website is crucial. This involves designing an easy-to-navigate interface, implementing search filters, and ensuring smooth checkout processes.
* Inventory Management: Managing inventory efficiently is essential. This includes tracking stock levels, updating availability status, and handling out-of-stock items.
* Payment Gateway Integration: Online bookstores need secure payment gateways to process transactions. Integrating reliable payment methods ensures smooth financial transactions.
* Shipping and Delivery: The scope includes setting up shipping options, calculating shipping costs, and ensuring timely delivery to customers.
* Reviews and Recommendations: Providing a platform for customer reviews and personalized book recommendations enhances the user experience.
* Marketing and Promotion: Online bookstores need marketing strategies to attract customers. This includes social media campaigns, email newsletters, and promotions.
* Security and Privacy: Ensuring data security, protecting customer information, and complying with privacy regulations are part of the scope.
* Mobile Responsiveness: With mobile usage on the rise, optimizing the website for mobile devices is essential.
* Analytics and Insights: Tracking website performance, user behavior, and sales metrics falls within the scope.

The scope can vary based on the bookstore’s specific goals, target audience, and business model.

**4. METHODOLOGY OF BOOKQUEST PLAZA**

When studying the methodology for an online bookstore, there are several approaches you can explore. Here are some key steps:

1. Research and Planning:

o Object-Oriented Approach: Consider using an Object-Oriented approach. This involves defining requirements, analysing them, and validating functionalities.

o Research Design: Understand the existing system and identify its problems.

o Population Study: Determine the target audience for your online bookstore.

o Data Collection: Collect data through methods like oral interviews or surveys2.

2. System Design:

* Use Case Diagram: Create a use case diagram to visualize system interactions.
* Class Diagram: Design the classes and relationships within the system.
* Sequence Diagram: Illustrate the flow of events during user interactions.
* Collaboration Diagram: Show how objects collaborate in the system.
* Activity Diagram: Represent the workflow of processes.
* Requirements List: Document the system requirements.

3. Implementation and Testing:

* Develop the online bookstore system based on the design.
* Test the system thoroughly to ensure it meets the requirements.

4. Customer Experience Enhancement:

* Focus on delivering exceptional customer service to build trust and loyalty.
* Optimize the website for mobile devices to accommodate users on smartphones and tablets.
* Implement personalized recommendations, reviews, and social proof elements to enhance the shopping experience and facilitate decision-making.

**5. OBJECTIVES:**

1. To create a seamless and intuitive online platform for browsing and purchasing books.
2. To provide users with personalized recommendations based on their preferences.
3. To streamline the ordering and delivery process for efficient customer service.
4. To establish a community-driven platform for book reviews and discussions.

**6. ANALYSIS AND INTERPRETATION**

* 1. **SYSTEM DESIGN:**

**Introduction:**

System design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

**6.1.1. Context Flow Diagram (CFD):**

* CFD are used to describe the detailed logic of a business process or a business rule.
* The Context Flow Diagram is a one in which the entire system is treated as a single process.
* It lists all the major inputs and outputs for the system.
* It will show the flow of control through the different program.
* The CFD shows the external entity acting in the software.
* The process is shown here in CFD as a single process or a bubble,

User

Bookshop

Automation

Admin

Database

Register,login,search,cart,order,payment

Confirmation

Manage users

Update,delete,edit

**6.1.2. Data Flow Diagram (DFD):**

The DFD is graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The DFD may be used to represent a system or a software at any level of abstraction DFD’s may be partitioned into level’s that represent increasing information flow and functional details. The DFD’s provides a mechanism for functional modelling as well as information flow modelling.

|  |  |
| --- | --- |
| **Notations** | **Description** |
|  | **Process:-**  A circle represents a process or transformation that is applied to the data or control and change it on same ways. |
|  | **Source or Sink:-**  A rectangle represents an external entity that is a system, element or another system that produces information for transformation by software or receives information produced by the software. |
|  | **Data Flow:-**  An arrow represents one or more data items or data objects. |
|  | **File or Database:-**  The open box represents data store, stored information that is used by the software. |

Admin

Login

Manage

users

Manage

Books

Manage

orders

Admin

Login

Book

Order

User

Login

Login

Register

User

Profile

Search

Add to cart

Order

Checkout

Book

Cart

Order

Order

Request for login

Check for login

 Response

 Reply

 User Info

User

info

 Check for login

  Response

  Reply

 Book Info

 Details

   Response

 Reply

    Response

 Order Info

  Reply

 Details

 Success

  Success

 Item Info

 Item details

 Order Details

 Placed Order

 Item List

 Register Success

 Request for Info

 Response

  Request for Info

  Response

 User Info

 Fetch Info

 Successful

 Successful

 Success

 Cart Details

 Add to cart

 Response

 Request for register

  Response

 Login

 Customer login check

If reply not found

Store Info

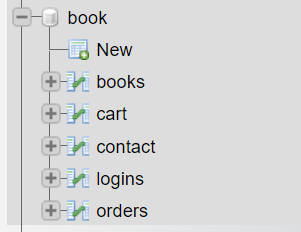
Update Info

Response

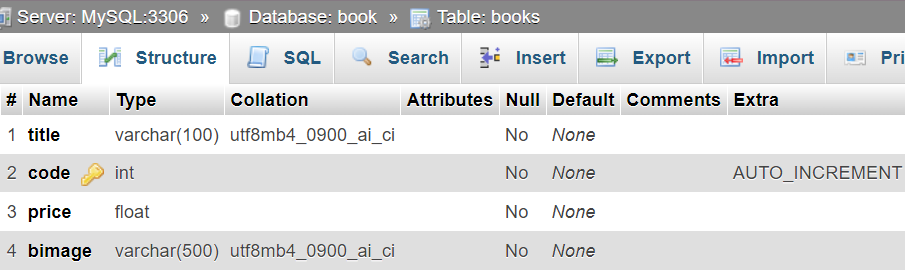
Item Details

**7.Data Design**

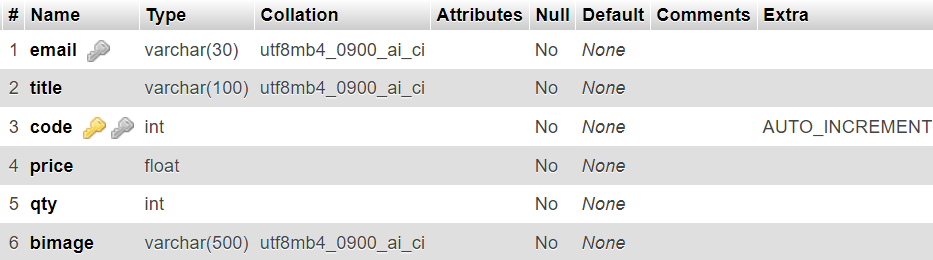
**7.1 Database Design**

Tables In book Database 

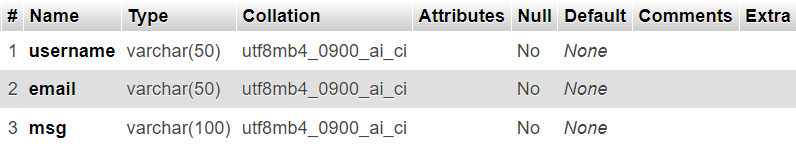
books



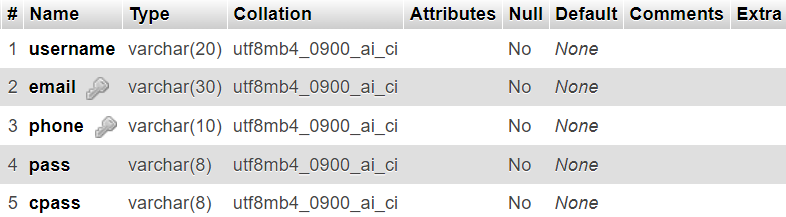
cart



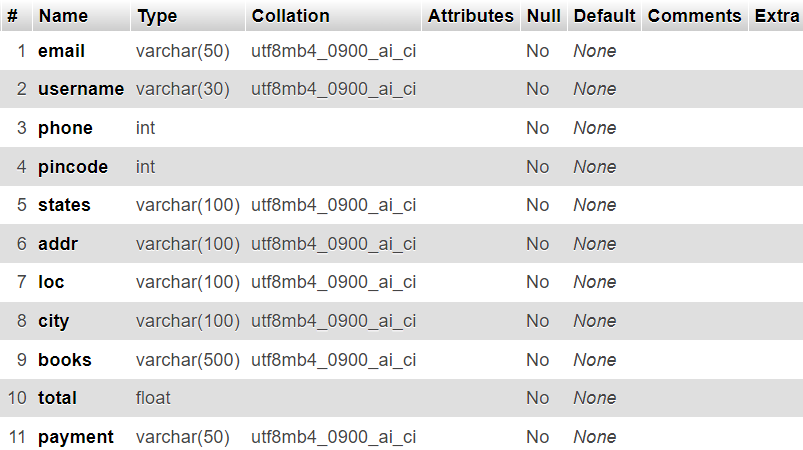
contact



Logins



order



**7.2. Entity Relationship Diagram [ERD]:**

The Entity Relationship Diagram ERD is popular high level conceptual model. This model and its variation are frequently used for the conceptual design of database application and many database design tools employ its concepts. We describe the basic structuring concepts and constraints of ER model and discuss their use in the design of conceptual schemes for database application. We also present the diagrammatic notation known as ER-Diagram.

The main focus of ER-modeling is data items in the system and relationship between them its main aim is to create an Er-Model for the data and user perspective. The sentence can be used during the development of the database and there are methods that use Er-Model to design the database for different database modules are frequently representing as Er diagram through the model can also be represented in mathematical forms.

An entity types defines the collection of entries that have same attributes. Each entity type in database is described by its name and attributes. The collection of all entries of a particular entity type in the database at any point in the time is called as entity set. An entity describes the schema or intention for the set of entries that share same structure. The collection of entities of particular entity type is grouped into an entity set, which is also called extensions of entity type. An important constraint on the entities of an entity type is the key constraints on the attributes.

An Entity is usually having an attribute whose values are distinct for each individual entity in entity set. Such attribute is called key attribute and its value can be used to identify each entity uniquely.

**Summary of Notations of ERD: -**

Entity types are shown in rectangles boxes. Relationship-types are shown in diamond shape boxes attached to the participating entity types with straight lines. Attributes are shown in ovals, straight line to its entity type or relationship type attaches each attribute.

Component attributes of a composite attributes are attached to a oval representing the composite attributes, multi values attribute are shown in the double ovals. Key attributes have their names underlined. Derived attributes are shown in the dotted oval.

**Basic of Entity Relationship Notation:-**

|  |  |
| --- | --- |
| **Symbol** | **Mean** |
|  | Entity |
|  | Weak Entity |
|  | Relationship |
|  | Identifying Relationship |
|  | Attribute |
|  | Key Attribute |
|  | Multivalued Attribute |
|  | Composite Attribute |
| R  1 N  E2  E1 | Cardinality ratio 1:N for E1:E2 in R |
| (Min,Max)  E  R | Structured constraint (Min, Max) on participation of E in R |

**Cart**

**BookQuest Plaza**

**Books**

has

manage

**Payment**

**User**

**Orders**

**Contact**

make

e

contains

make

gives

Add to cart

**Admin**

**CONCLUSION**

In conclusion, the development of an online bookstore website offers significant advantages in today's digital age. By providing a convenient platform for users to browse, purchase, and interact with a vast array of books, such a website can cater to the evolving needs and preferences of modern readers. Through seamless navigation, personalized recommendations, secure transactions, and efficient customer support, the online bookstore can enhance the overall shopping experience and foster customer loyalty. Furthermore, leveraging data analytics and digital marketing strategies can help drive traffic, increase sales, and expand the bookstore's reach to a global audience. Overall, investing in an online bookstore website presents a compelling opportunity for growth, innovation, and success in the competitive landscape of the book industry.

**LEARNING OUTCOMES OF THE PROJECT**

Online bookstore provides valuable learning outcomes across various domains:

1. **Technical Proficiency**: Students gain hands-on experience in web development, database management, and e-commerce functionalities. They learn to design user-friendly interfaces, implement secure payment gateways, and optimize website performance.
2. **Digital Marketing**: Understanding online consumer behavior, students learn to devise effective digital marketing strategies. They explore search engine optimization (SEO), social media marketing, email campaigns, and content creation to attract and retain customers.
3. **Customer Experience Optimization**: Through user testing and feedback analysis, students refine the website's usability and functionality. They learn to prioritize features that enhance the browsing, shopping, and support experiences, ultimately improving customer satisfaction and retention.
4. **Data Analysis and Insights**: Students utilize analytics tools to track website traffic, user interactions, and sales data. They learn to interpret these metrics to identify trends, make data-driven decisions, and optimize the bookstore's performance and offerings.
5. **Business Management Skills**: Operating an online bookstore involves inventory management, pricing strategies, and vendor relationships. Students develop skills in supply chain management, financial planning, and business development, gaining a holistic understanding of running an e-commerce venture.

**Coding**

Login.php

<?php

session\_start();

$conn = new mysqli("localhost", "root", "", "book");

if ($conn->connect\_error) {

die("Connection Failed:" . $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$email = $\_POST['email'];

$pass = $\_POST['pass'];

if (empty($email) || empty($pass)) {

echo ("<script language='javascript'>

window.alert('You did not complete all the required fields')

window.location.href='login.php'

</script>");

exit();

}

if ($email === 'ad@gmail.com' && $pass === 'ad') {

$\_SESSION['log'] = "yes";

$\_SESSION['email'] = $email;

header("location:adminlog.php");

exit();

}

$sql = "SELECT username FROM logins WHERE email='$email' AND pass='$pass'";

$result = mysqli\_query($conn, $sql);

if (mysqli\_num\_rows($result) > 0) {

$row = mysqli\_fetch\_assoc($result);

$username = $row['username'];

$\_SESSION['log'] = "yes";

$\_SESSION['email'] = $email;

$\_SESSION['username'] = $username;

echo ("<script language='javascript'>

window.alert('Logged in successfully')

window.location.href='user.php'

</script>");

exit();

} else {

echo ("<script language='javascript'>

window.alert('Wrong email or password')

window.location.href='login.php'

</script>");

exit();

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Log In</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--BOOTSTRAP-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous" referrerpolicy="no-referrer">

<!--REMIXICONS-->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/remixicon/3.5.0/remixicon.css">

<!--SWIPER CSS-->

<link rel="stylesheet" href="css/swiper-bundle.min.css">

<!--CSS-->

<link rel="stylesheet" href="css/styles.css">

</head>

<body>

<section class="login grid" id="login-content">

<form action="" method="POST" class="login\_\_form grid">

<h3 class="login\_\_title">Log In</h3>

<div class="login\_\_group grid">

<div>

<label for="email" class="login\_\_label">Email</label>

<input type="email" id="email" name="email" class="login\_\_input" placeholder="Enter your email-id" required>

</div>

<div>

<label for="pass" class="login\_\_label">Password</label>

<input type="password" id="pass" name="pass" class="login\_\_input" placeholder="Enter your password" required>

</div>

</div>

<div>

<span class="login\_\_signup">

Don't have an account? <a href="signup.php">Sign up</a>

</span>

<a href="forgot.php" class="login\_\_forgot">You forgot your password</a><br>

<button type="submit" class="login\_\_button button">Log In</button>

</div>

</form>

</section>

</body>

</html>

Signup.php

<?php

$conn = new mysqli("localhost", "root", "", "book");

if ($conn->connect\_error) {

die("Connection Failed:" . $conn->connect\_error);

}

if($\_SERVER["REQUEST\_METHOD"]=="POST"){

$username = $\_POST['username'];

$email =$\_POST['email'];

$phone=$\_POST['phone'];

$pass = $\_POST['pass'];

$cpass =$\_POST['cpass'];

$query="SELECT \* FROM logins WHERE email='$email'";

$query1="SELECT \*FROM logins WHERE phone='$phone'";

$result=$conn->query($query);

$result1=$conn->query($query1);

if($result->num\_rows>0){

echo "<script>

alert('Email already exist');

window.location.href='signup.php';

</script>";

}elseif($result1->num\_rows>0){

echo "<script>

alert('Phone Number already exist');

window.location.href='signup.php';

</script>";

}

elseif($pass == $cpass){

$sql="INSERT INTO logins(username,email,phone,pass,cpass) VALUES('$username','$email','$phone','$pass','$cpass')";

if($conn->query($sql)===TRUE){

echo "<script>

alert('Registration Successfull');

window.location.href='login.php';

</script>";

}

else{

echo "<script>

alert('Invalid credentials');

window.location.href='signup.php';

</script>";

}

}

else{

echo "<script>

alert('Password does not match');

window.location.href='signup.php';

</script>";

}

if($phonenumber>10){

echo"<script>

alert('Phonenumber should contain only 10 digits');

</script>";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sign Up</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--BOOTSTRAP-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous" referrerpolicy="n0-referrer">

<!--REMIXICONS-->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/remixicon/3.5.0/remixicon.css">

<!--SWIPER CSS-->

<link rel="stylesheet" href="css/swiper-bundle.min.css">

<!--CSS-->

<link rel="stylesheet" href="css/styles.css">

</head>

<body>

<section class="signup grid" id="signup-content">

<form action="" method="post" class="signup\_\_form grid">

<h3 class="signup\_\_title">Sign Up</h3>

<div class="signup\_\_group grid">

<div>

<label for="username" class="signup\_\_label">Name</label>

<input type="text" id="username" name="username" class="signup\_\_input" placeholder="Enter username" pattern="[a-zA-Z ]{1,}" title="only alphabets allowed" required>

</div>

<div>

<label for="email" class="signup\_\_label">Email</label>

<input type="email" id="email" name="email" class="signup\_\_input" placeholder="Enter email-id" required>

</div>

<div>

<label for="phone" class="signup\_\_label">Phone Number</label>

<input type="tel" pattern="[0-9]{10}" title="Phonenumber should contain only 10 digits" id="phone" name="phone" class="signup\_\_input" placeholder="Enter phone number" required>

</div>

<div>

<label for="pass" class="signup\_\_label">Password</label>

<input type="password" id="pass" name="pass" class="signup\_\_input" placeholder="Enter Password" required>

</div>

<div>

<label for="cpass" class="signup\_\_label">Confirm Password</label>

<input type="password" id="cpass" name="cpass" class="signup\_\_input" placeholder="Enter Password" required>

</div>

</div>

<div>

<span class="signup\_\_login">

Already have an account ? <a href="login.php">Log In</a>

</span>

<button type="submit" class="signup\_\_button button">Sign Up</button>

</div>

</form>

</section>

</body>

</html>

Adminlog.php

<?php

session\_start();

if (!isset($\_SESSION['log']) || $\_SESSION['log'] !== "yes" || $\_SESSION['email'] !== 'ad@gmail.com') {

header("location: login.php");

exit();

}

// Database connection

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "book";

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$success = false;

$message = "";

$view\_books = [];

$view\_users = [];

$view\_orders = [];

$view\_messages = [];

$current\_section = isset($\_POST['current\_section']) ? $\_POST['current\_section'] : 'manage-books';

if (isset($\_POST['addbook'])) {

// Fetching values from add book form

$book\_title = $\_POST['add-book-title'];

$book\_code = $\_POST['add-book-code'];

$book\_price = $\_POST['add-book-price'];

$image = $\_POST['image'];

$sql\_insert\_data = "INSERT INTO books(title, code, price, bimage) VALUES ('$book\_title', '$book\_code', '$book\_price', '$image')";

if ($conn->query($sql\_insert\_data) === TRUE) {

$success = true;

$message = "Data inserted successfully";

} else {

$message = "Error inserting data: " . $conn->error;

}

}

if (isset($\_POST['removebook'])) {

// Fetching values from remove book form

$book\_code = $\_POST['remove-book-code'];

$sql\_remove\_data = "DELETE FROM books WHERE code='$book\_code'";

if ($conn->query($sql\_remove\_data) === TRUE) {

$success = true;

$message = "Data removed successfully";

} else {

$message = "Error removing data: " . $conn->error;

}

}

if (isset($\_POST['updatebook'])) {

// Fetching values from update book form

$book\_code = $\_POST['upd-book-code'];

$book\_title = $\_POST['upd-book-title'];

$book\_price = $\_POST['upd-book-price'];

$sql\_update\_data = "UPDATE books SET title='$book\_title', price='$book\_price' WHERE code='$book\_code'";

if ($conn->query($sql\_update\_data) === TRUE) {

$success = true;

$message = "Data updated successfully";

} else {

$message = "Error updating data: " . $conn->error;

}

}

if (isset($\_POST['viewbooks'])) {

$sql\_view\_data = "SELECT \* FROM books";

$result = $conn->query($sql\_view\_data);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$view\_books[] = $row;

}

} else {

$message = "No data found";

}

}

if (isset($\_POST['viewusers'])) {

$sql\_view\_users = "SELECT \* FROM logins";

$result = $conn->query($sql\_view\_users);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$view\_users[] = $row;

}

} else {

$message = "No user data found";

}

}

if (isset($\_POST['vieworders'])) {

$sql\_view\_orders = "SELECT \* FROM orders";

$result = $conn->query($sql\_view\_orders);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$view\_orders[] = $row;

}

} else {

$message = "No order data found";

}

}

if (isset($\_POST['viewmessage'])) {

$sql\_view\_messages = "SELECT \* FROM contact";

$result = $conn->query($sql\_view\_messages);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$view\_messages[] = $row;

}

} else {

$message = "No messages found";

}

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bookstore Admin Panel</title>

<link rel="stylesheet" href="STYLENEW.css">

</head>

<body>

<header>

<div class="header-content">

<h1>Admin Panel</h1>

<form method="post" action="logout.php">

<p> <button type="submit" class="logout-button">Log Out</button></p>

</form>

</div>

</header>

<div class="container">

<nav class="sidebar">

<ul>

<li><a href="javascript:void(0)" onclick="showSection('manage-books')">Manage Books</a></li>

<li><a href="javascript:void(0)" onclick="showSection('manage-users')">Manage Users</a></li>

<li><a href="javascript:void(0)" onclick="showSection('manage-orders')">Manage Orders</a></li>

<li><a href="javascript:void(0)" onclick="showSection('message')">Message</a></li>

</ul>

</nav>

<main class="content">

<section id="manage-books" class="admin-section">

<h2>Manage Books</h2>

<div class="book-buttons">

<button onclick="showForm('add-book-form')">Add Book</button>

<button onclick="showForm('remove-book-form')">Remove Book</button>

<button onclick="showForm('update-book-form')">Update Book Info</button>

<button onclick="showForm('view-books-form')">View Books</button>

</div>

<form id="add-book-form" class="book-form" method="post" action="" style="display: none;">

<h3>Add Book</h3>

<label for="add-book-title">Book Title:</label>

<input type="text" id="add-book-title" name="add-book-title">

<label for="add-book-code">Book Code:</label>

<input type="text" id="add-book-code" name="add-book-code">

<label for="add-book-price">Book Price:</label>

<input type="text" id="add-book-price" name="add-book-price">

<label for="image">Book Image:</label>

<input type="file" accept="image/\*" id="image" name="image"><br><br>

<input type="hidden" name="current\_section" value="manage-books">

<button type="submit" name="addbook">Add Book</button>

</form>

<form id="remove-book-form" class="book-form" method="post" action="" style="display: none;">

<h3>Remove Book</h3>

<label for="remove-book-code">Book Code:</label>

<input type="text" id="remove-book-code" name="remove-book-code">

<input type="hidden" name="current\_section" value="manage-books">

<button type="submit" name="removebook">Remove Book</button>

</form>

<form id="update-book-form" class="book-form" method="post" action="" style="display: none;">

<h3>Update Book Info</h3>

<label for="upd-book-title">New Book Title:</label>

<input type="text" id="upd-book-title" name="upd-book-title">

<label for="upd-book-price">New Book Price:</label>

<input type="text" id="upd-book-price" name="upd-book-price">

<label for="upd-book-code">Book Code:</label>

<input type="text" id="upd-book-code" name="upd-book-code">

<input type="hidden" name="current\_section" value="manage-books">

<button type="submit" name="updatebook">Update Book</button>

</form>

<form id="view-books-form" class="book-form" method="post" action="" style="display: none;">

<h3>Viewing The Books In Stock</h3>

<input type="hidden" name="current\_section" value="manage-books">

<button type="submit" name="viewbooks">View Books</button>

</form>

<?php if (!empty($view\_books)) { ?>

<div class="book-view" id="view-books">

<?php foreach ($view\_books as $book) { ?>

<div class="viewbks\_container">

<img src="img/<?php echo $book['bimage']; ?>" alt="" class="bimage">

<div class="title"><?php echo $book['title']; ?></div>

<div class="code">Code: <?php echo $book['code']; ?></div>

<div class="price">₹<?php echo $book['price']; ?></div>

</div>

<?php } ?>

</div>

<?php } ?>

</section>

<section id="manage-users" class="admin-section" style="display: none;">

<h2>Manage Users</h2>

<form id="view-users-form" method="post" action="">

<h3>View Users</h3>

<input type="hidden" name="current\_section" value="manage-users">

<button type="submit" name="viewusers">View Users</button>

</form>

<?php if (!empty($view\_users)) { ?>

<div id="view-users" class="user-view">

<h3>Users List</h3>

<table>

<thead>

<tr>

<th>Username</th>

<th>Email</th>

<th>Phone Number</th>

</tr>

</thead>

<tbody>

<?php foreach ($view\_users as $user) { ?>

<tr>

<td><?php echo $user['username']; ?></td>

<td><?php echo $user['email']; ?></td>

<td><?php echo $user['phone']; ?></td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

<?php } ?>

</section>

<section id="manage-orders" class="admin-section" style="display: none;">

<h2>Manage Orders</h2>

<form id="view-orders-form" method="post" action="">

<h3>View Orders</h3>

<input type="hidden" name="current\_section" value="manage-orders">

<button type="submit" name="vieworders">View Orders</button>

</form>

<?php if (!empty($view\_orders)) { ?>

<div id="view-orders" class="order-view">

<h3>Orders List</h3>

<table>

<thead>

<tr>

<th>Customer Name</th>

<th>Phone</th>

<th>Pincode</th>

<th>State</th>

<th>Address</th>

<th>Locality</th>

<th>City</th>

</tr>

</thead>

<tbody>

<?php foreach ($view\_orders as $order) { ?>

<tr>

<td><?php echo $order['username']; ?></td>

<td><?php echo $order['phone']; ?></td>

<td><?php echo $order['pincode']; ?></td>

<td><?php echo $order['state']; ?></td>

<td><?php echo $order['addr']; ?></td>

<td><?php echo $order['loc']; ?></td>

<td><?php echo $order['city']; ?></td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

<?php } ?>

</section>

<section id="message" class="admin-section" style="display: none;">

<h2>Message</h2>

<form id="view-message-form" method="post" action="">

<input type="hidden" name="current\_section" value="message">

<button type="submit" name="viewmessage">View Messages</button>

</form>

<?php if (!empty($view\_messages)) { ?>

<div id="view-messages" class="message-view">

<h3>Messages List</h3>

<table>

<thead>

<tr>

<th>Username</th>

<th>Email id</th>

<th>Message</th>

</tr>

</thead>

<tbody>

<?php foreach ($view\_messages as $contact) { ?>

<tr>

<td><?php echo $contact['username']; ?></td>

<td><?php echo $contact['email']; ?></td>

<td><?php echo $contact['msg']; ?></td>

<!-- Add more columns as needed -->

</tr>

<?php } ?>

</tbody>

</table>

</div>

<?php } else { echo $message; } ?>

</section>

</main>

</div>

<script>

function showSection(sectionId) {

const sections = document.querySelectorAll('.admin-section');

sections.forEach(section => section.style.display = 'none');

document.getElementById(sectionId).style.display = 'block';

}

function showForm(formId) {

const forms = document.querySelectorAll('.book-form');

forms.forEach(form => form.style.display = 'none');

document.getElementById(formId).style.display = 'block';

}

document.addEventListener('DOMContentLoaded', () => {

const currentSection = "<?php echo $current\_section; ?>";

showSection(currentSection);

});

window.onload = function() {

const success = "<?php echo $success ? '1' : '0'; ?>";

const message = "<?php echo $message; ?>";

if (success === "1") {

alert(message);

} else if (message !== "") {

alert(message);

}

};

function goBack(sectionId) {

const viewBooksSection = document.getElementById('view-books');

if (viewBooksSection) {

viewBooksSection.style.display = 'none';

}

showSection(sectionId);

}

</script>

</body>

</html>

Cart.php

<?php

session\_start();

$conn = new mysqli('localhost', 'root', '', 'book');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$email = $\_SESSION['email'];

if(!isset($\_SESSION['email']) || empty($\_SESSION['email'])) {

header('location:login.php');

exit();

}

if(isset($\_POST['update'])){

$bkcode = $\_POST['bkcode'];

$bkqty = $\_POST['bkqty'];

$sql ="UPDATE cart SET qty = '$bkqty' WHERE code = '$bkcode'" or die('query failed');

if($conn->query($sql)===TRUE){

echo ("<script language='javascript'>

window.alert('Cart Quantity Updated!')

window.location.href='cart.php'

</script>");

exit();

}

}

if(isset($\_GET['delete'])){

$delete\_code = $\_GET['delete'];

$sql ="DELETE FROM cart WHERE code = '$delete\_code'" or die('query failed');

if($conn->query($sql)===TRUE){

echo ("<script language='javascript'>

window.alert('Book Removed From Cart!')

window.location.href='cart.php'

</script>");

exit();

}

}

if(isset($\_GET['delete\_all'])){

$sql = "DELETE FROM cart WHERE email = '$email'" or die('query failed');

if($conn->query($sql)===TRUE){

echo ("<script language='javascript'>

window.alert('All Books Removed From Cart!')

window.location.href='cart.php'

</script>");

exit();

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--BOOTSTRAP-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous" referrerpolicy="no-referrer">

<!--REMIXICONS-->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/remixicon/3.5.0/remixicon.css">

<!--SWIPER CSS-->

<link rel="stylesheet" href="css/swiper-bundle.min.css">

<!--CSS-->

<link rel="stylesheet" href="css/styles.css">

<title>Cart Page</title>

</head>

<body>

<!--HEADER-->

<?php include 'header.php'; ?>

<div class="cart\_\_header">

<h3 class="cart\_\_title">Your Cart</h3><br>

</div>

<div class="cart">

<div class="cart\_\_container">

<?php

$total\_amt = 0;

$sql = "SELECT \* FROM cart WHERE email = '$email'";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

?>

<div class="bks">

<form action="" method="post" class="box">

<a href="cart.php?delete=<?php echo $row['code']; ?>" class="close-icon" onclick="return confirm('Do You want to Delete This Book From Cart?');">

<i class="ri-close-line"></i>

</a>

<?php if (strpos($row['bimage'], 'img/') === 0): ?>

<img src="<?php echo $row['bimage']; ?>" alt="Book Image">

<?php else: ?>

<img src="img/<?php echo $row['bimage']; ?>" alt="Book Image">

<?php endif; ?>

<div class="bktitle"><?php echo htmlspecialchars($row['title']); ?></div>

<div class="bkprice">₹<?php echo htmlspecialchars($row['price']); ?></div>

<input type="hidden" name="bkcode" value="<?php echo htmlspecialchars($row['code']); ?>">

<div class="qtyupdate">

<input type="number" class="bkqty" name="bkqty" min="1" max="5" value="<?php echo htmlspecialchars($row['qty']); ?>">

<button type="submit" class="update\_\_button button" name="update">Update</button>

</div>

</form>

<div class="total">Price : <span>₹<?php echo $total = ($row['qty'] \* $row['price']); ?></span> </div>

</div>

<?php

$total\_amt += $total;

}

} else {

echo '<p class="empty">Your cart is empty!</p>';

}

?>

</div>

<div style="margin-top: 2rem; text-align:center;">

<a href="cart.php?delete\_all" class="delete\_button button <?php echo ($total\_amt > 1)?'':'disabled'; ?>" onclick="return confirm('delete all from cart?');">Delete All Books In Cart</a>

</div>

<div class="cart\_total">

<p>Total Amount : <span>₹<?php echo $total\_amt; ?></span></p>

<div class="cont\_container">

<a href="search.php" class="continue\_button button">Continue Browsing</a>

<a href="checkout.php" class="checkout\_button button <?php echo ($total\_amt > 1)?'':'disabled'; ?>">Checkout</a>

</div>

</div>

</div>

<?php

$conn->close();

?>

<!--FOOTER-->

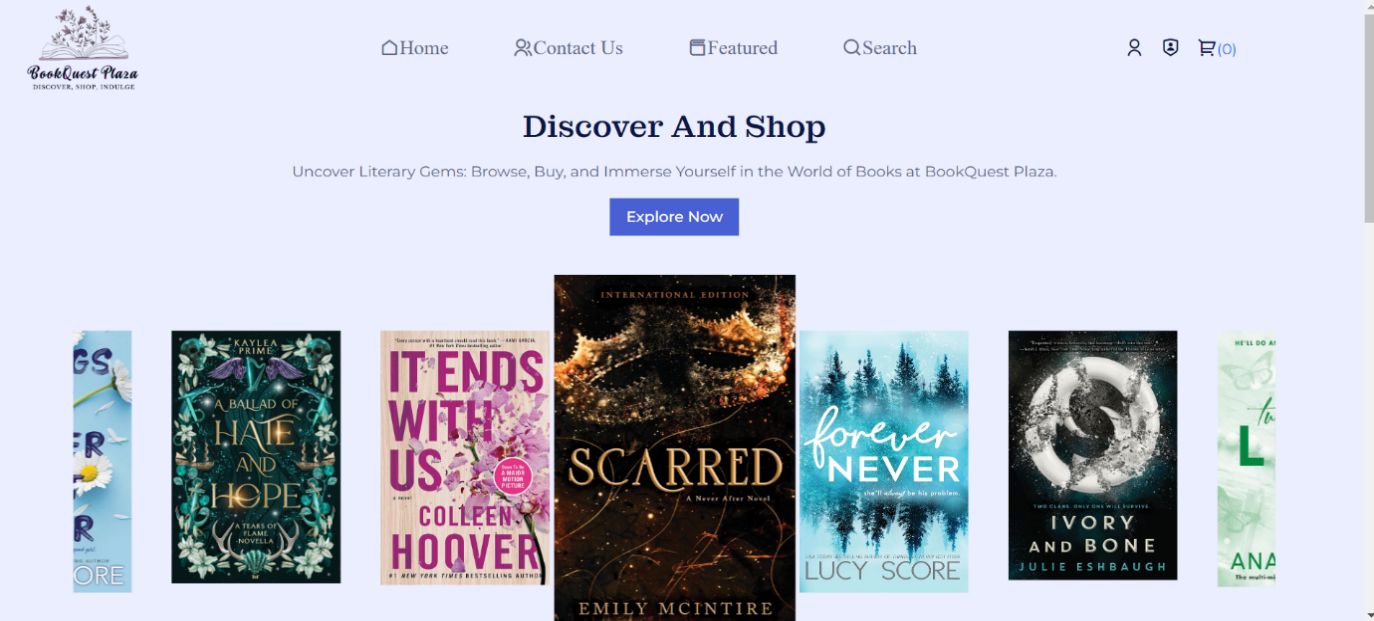
<?php include 'footer.php'; ?>

</body>

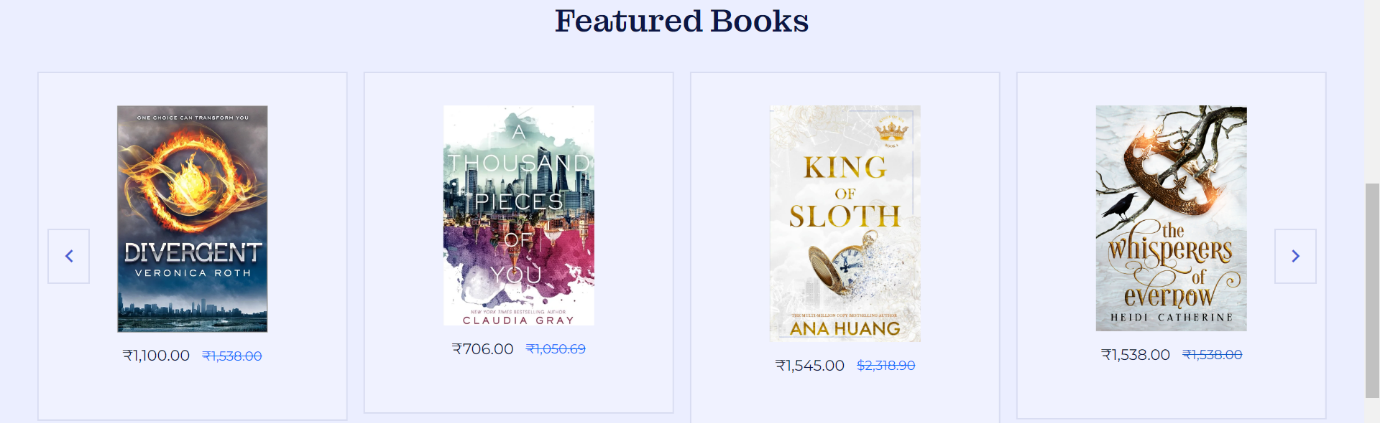
</html>

**PHOTOGRAPHY**

Home Page



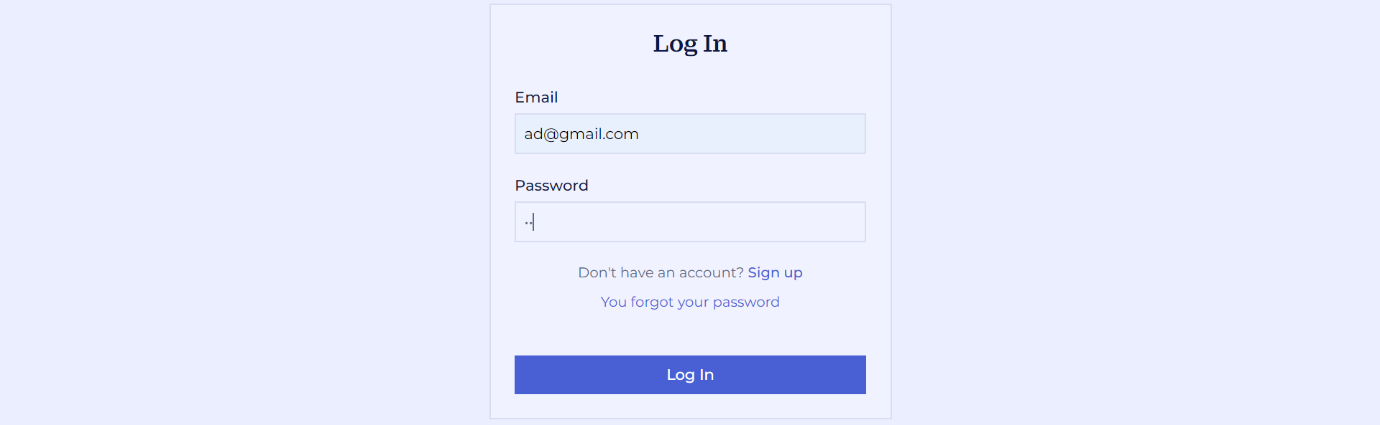


****

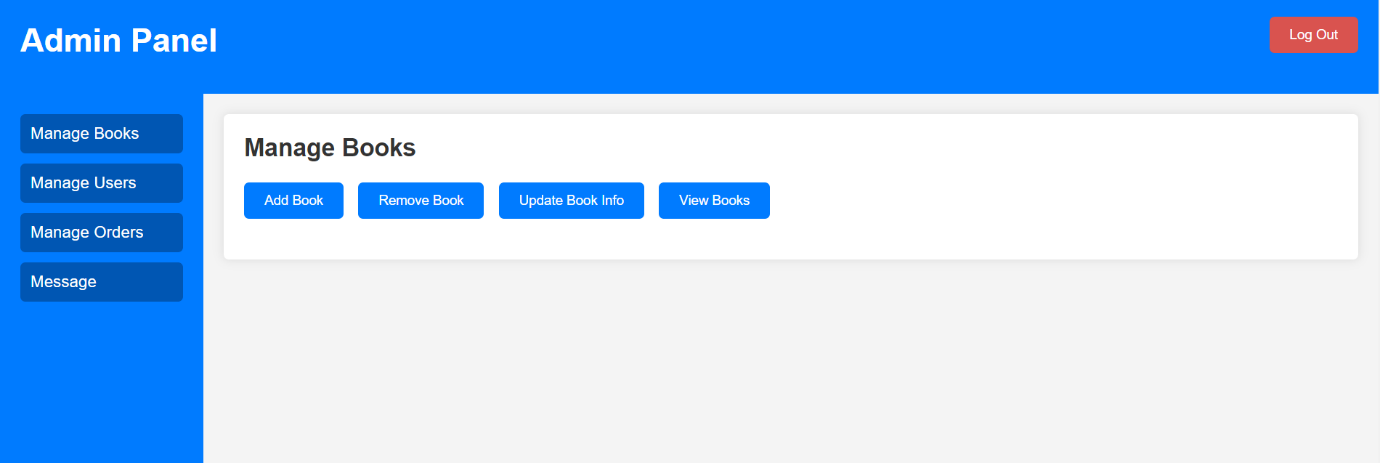
****

**Admin Side**

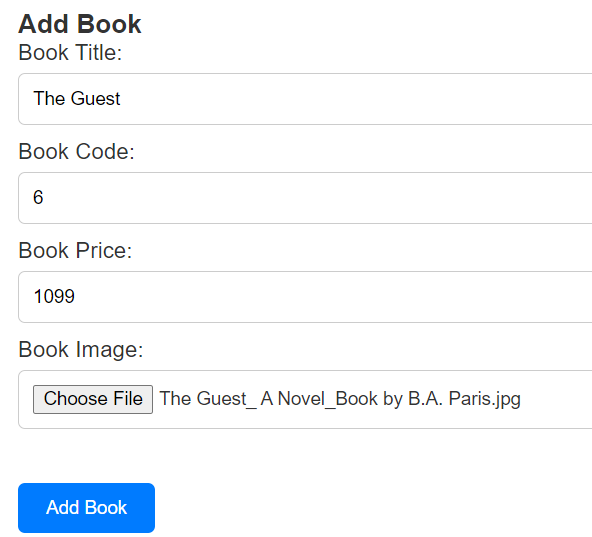
Login Page



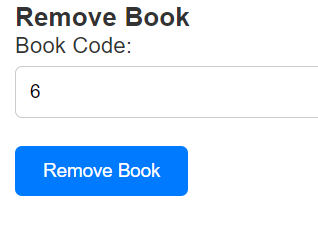
Admin Panel



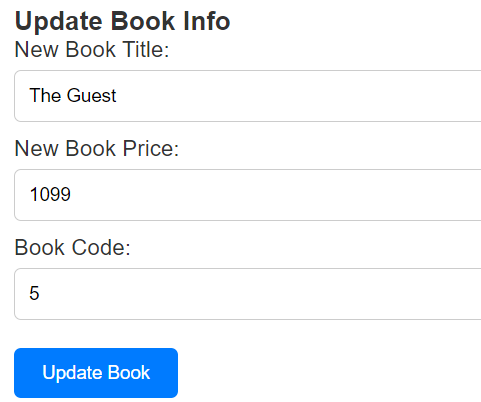
Add Books



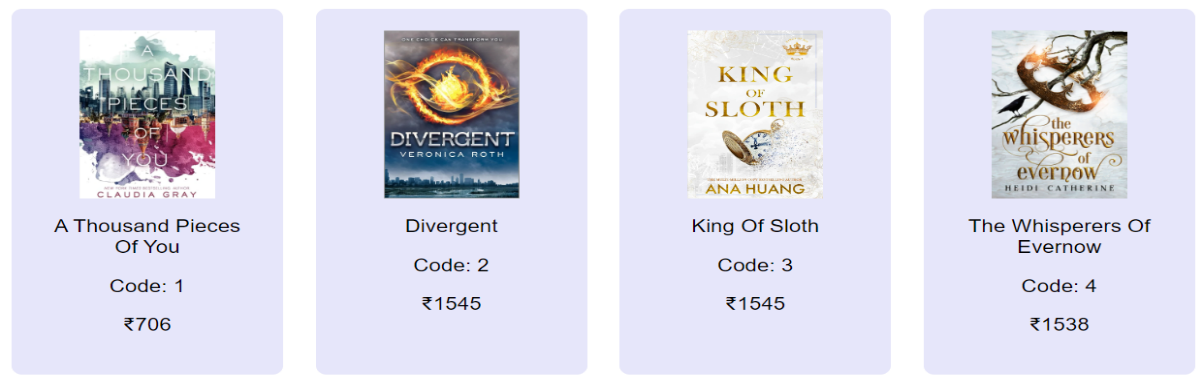
Remove Books



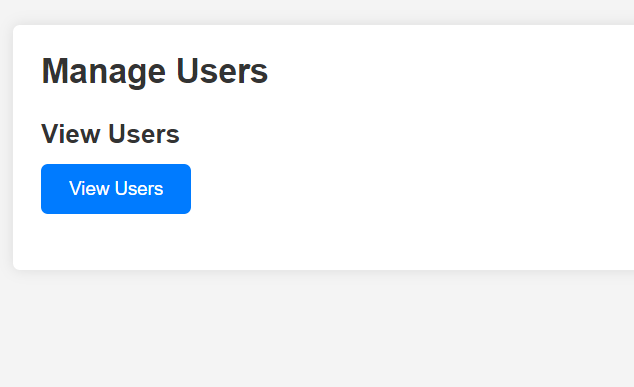
Update Book Info



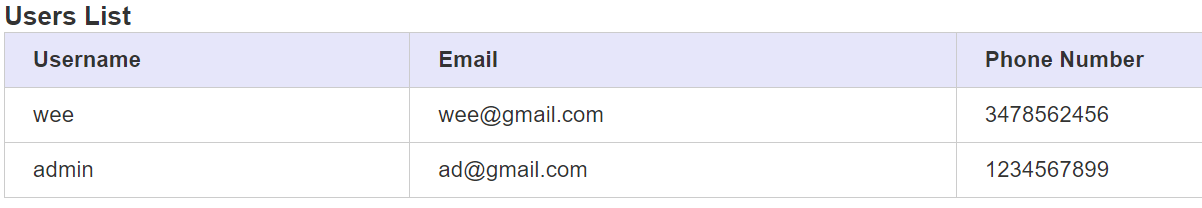
View Books



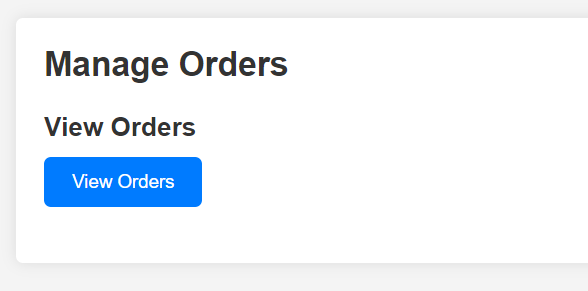
Manage Users



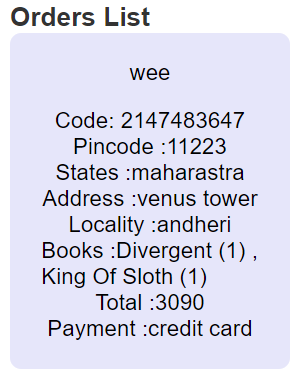
View Users



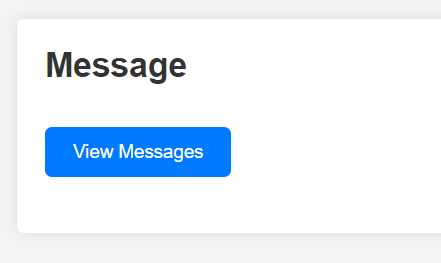
Manage Orders



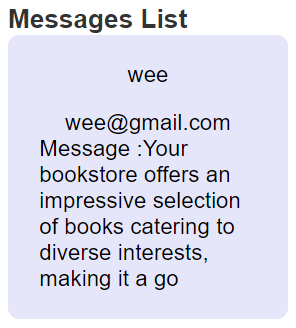
View Orders



Message

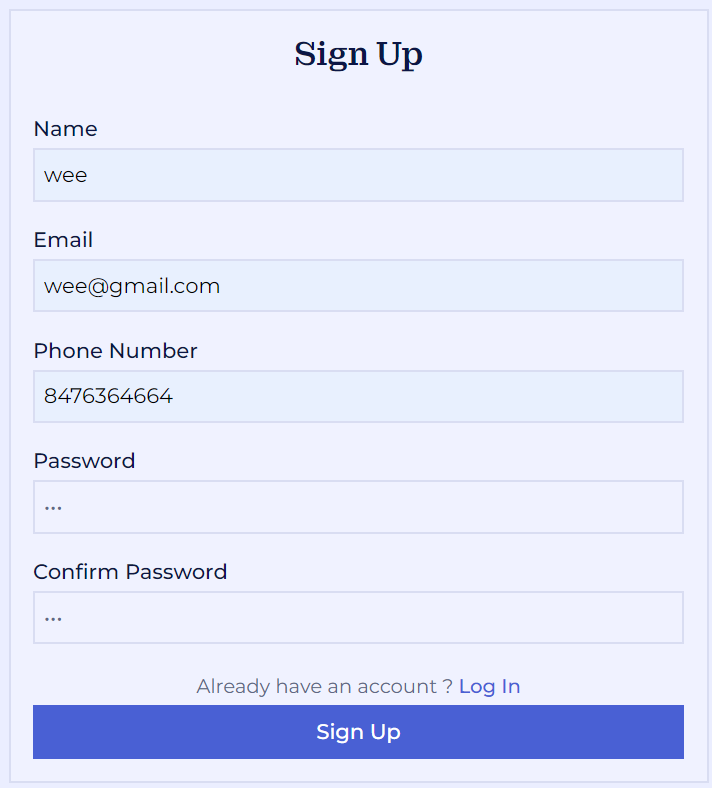


View Message

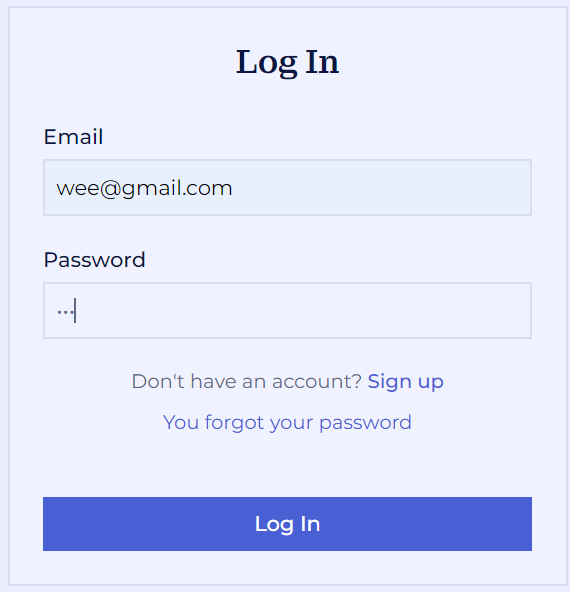


**Customer View**

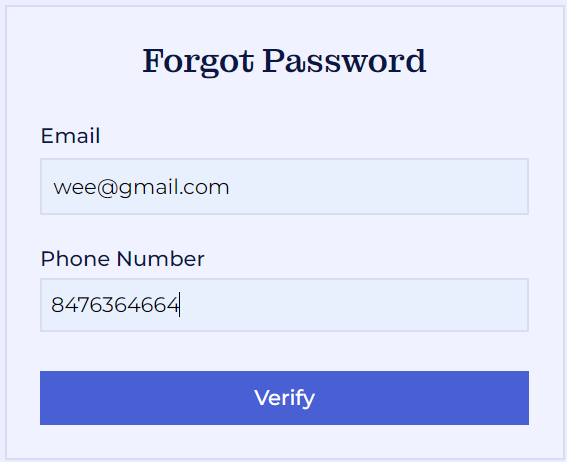
Customer Registration



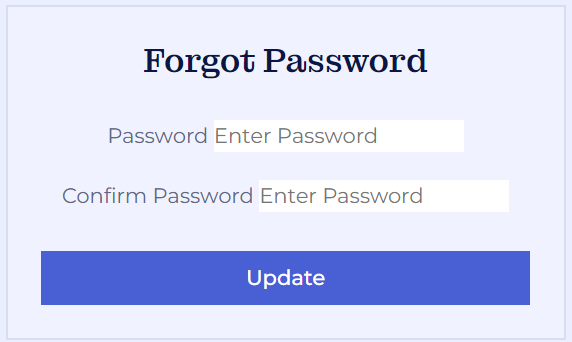
Customer Login



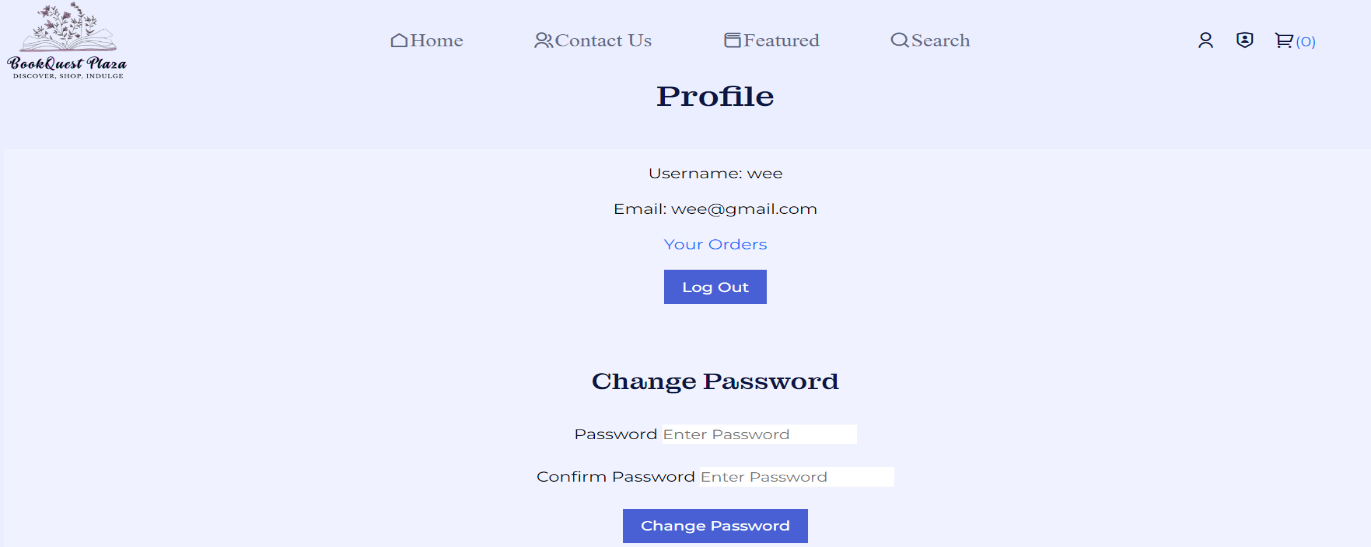
Forget Password



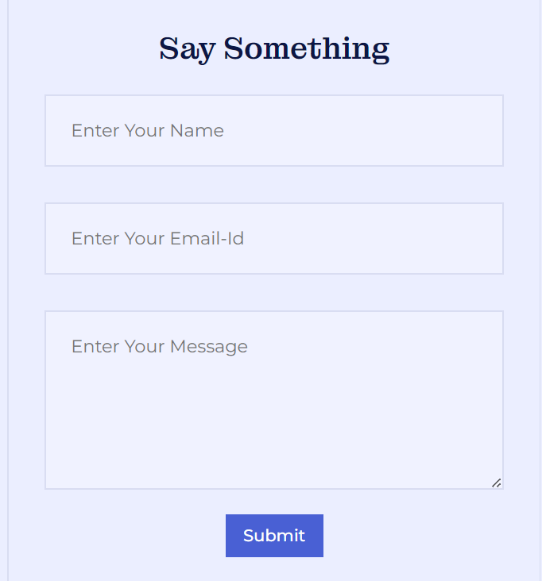
Change Password



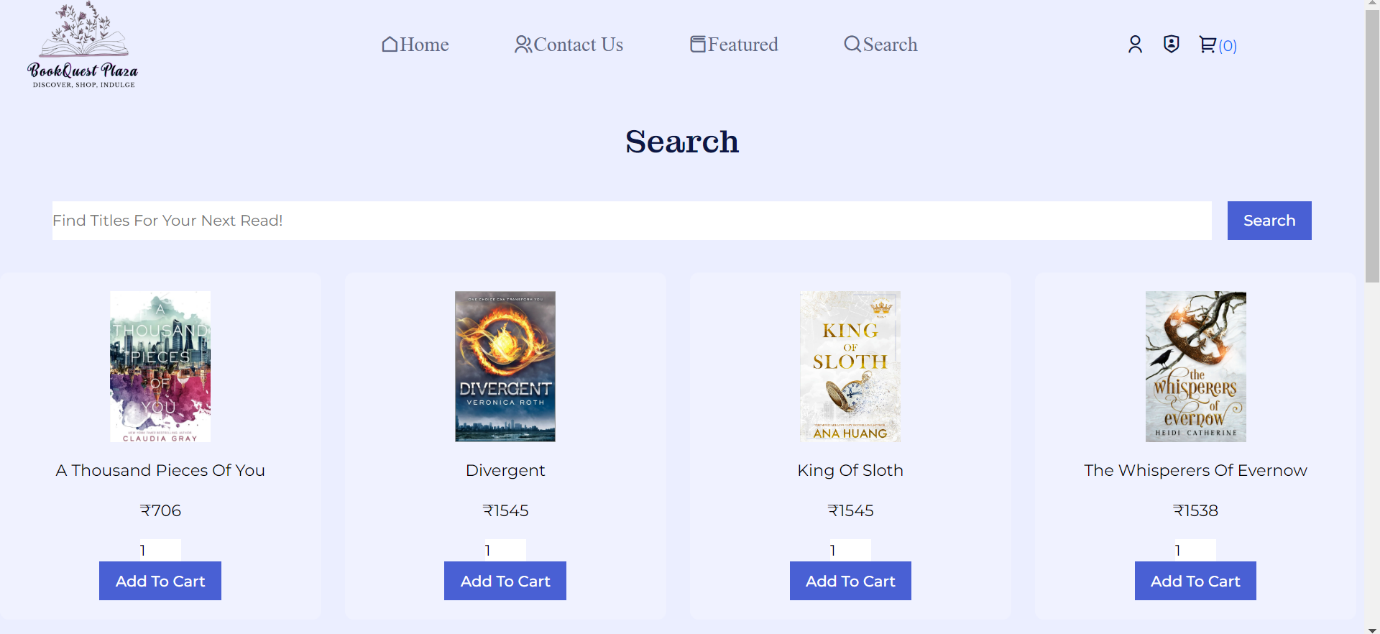
User Profile



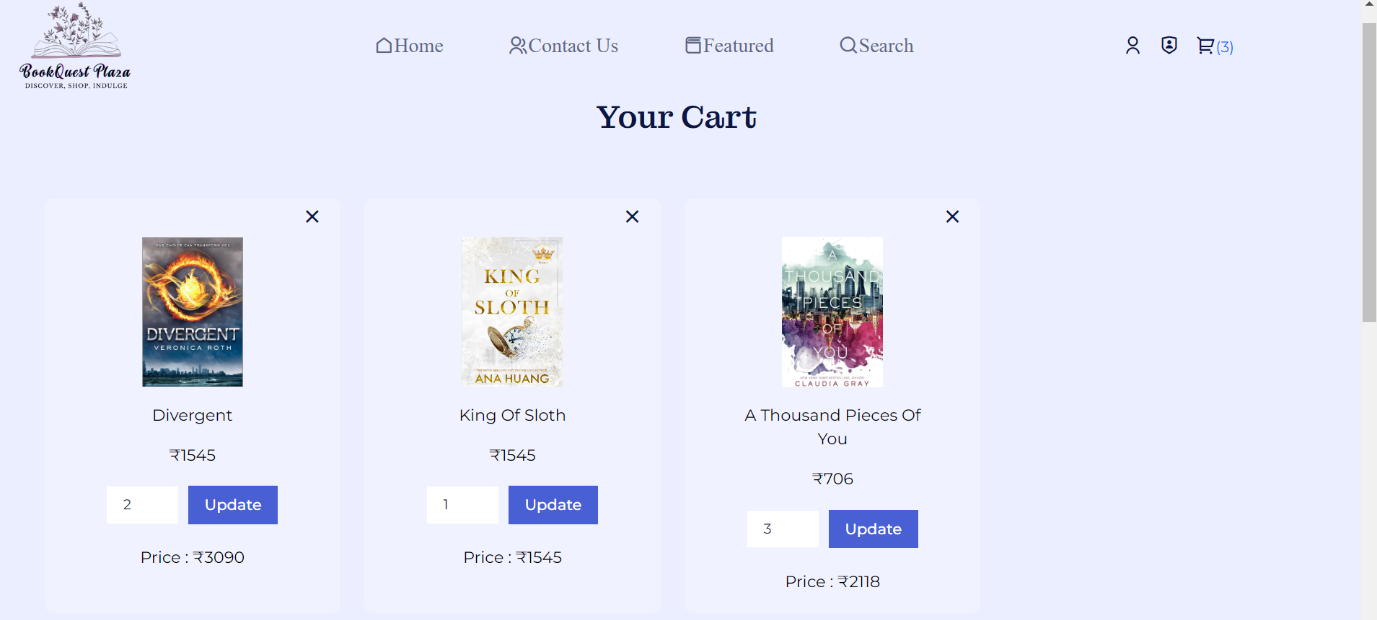
Get In Touch

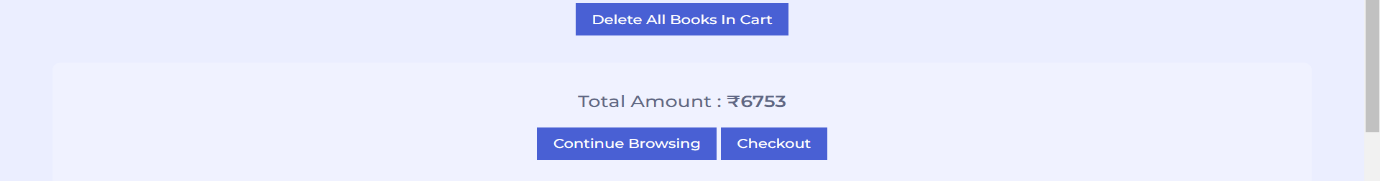


Search Page

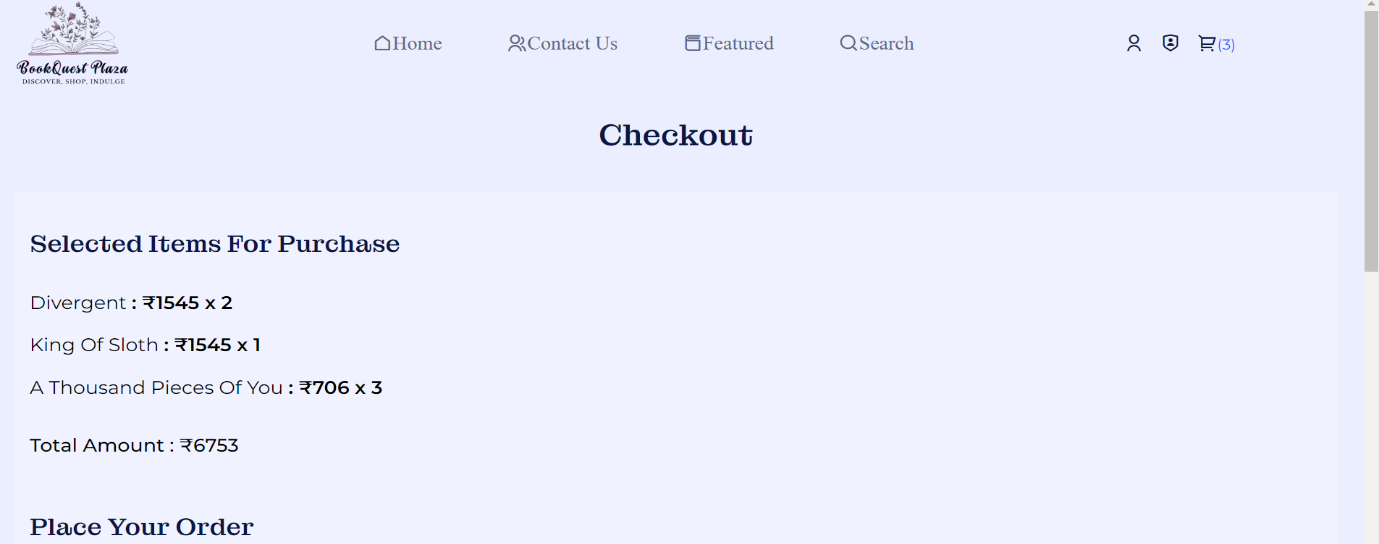


Cart Page

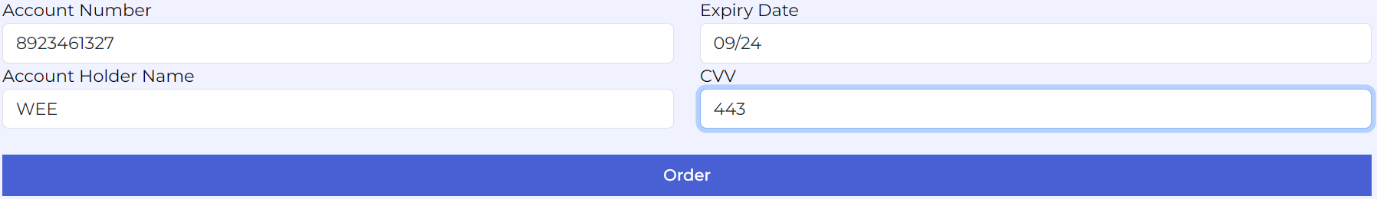




Checkout







Order Page



**TESTING AND IMPLEMENTATION**

Testing and implementation is the process, which tells the reality efficiency and the flexibility of the system design. Reliability means how much the user is expecting from the system. Flexibility tells how much the user is comfortable and hopes additional facilities with the system

**Test case : 1**

Objectives: Test for username and password

Test Data: Valid- One of the valid login name and password to enter the program

Invalid- Use of invalid login name and password to enter the program

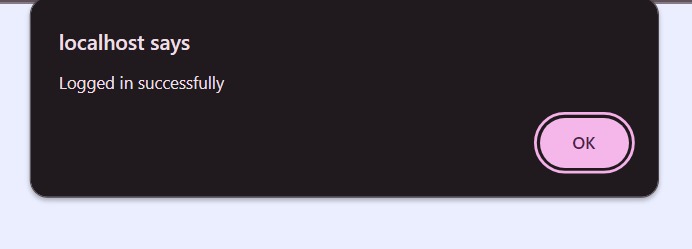
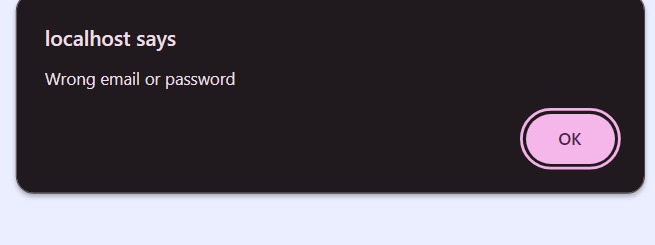
Output: Valid: Enter into the System normally

Invalid: Show the error message

Result: Valid: The user was allowed to enter the program

Invalid: The user is prompted with an error message and restricted to enter the program

**Conclusion**: Both the valid and invalid results are tested. Output matches with the required result hence the test case is successful



**Test case:2**

Objectives: Test for changing password

Test Data: Valid: valid password to be entered

Invalid: Password is blank and invalid password

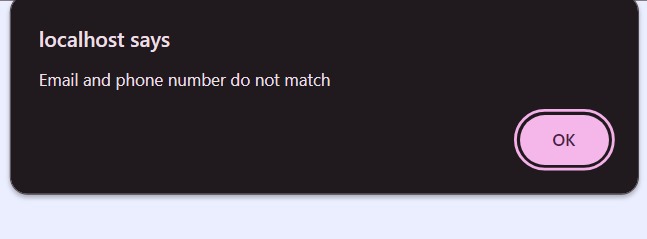
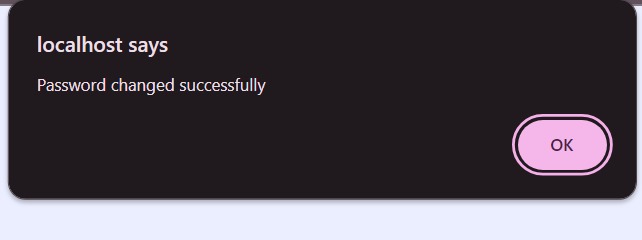
Output: Valid: Allows changing the password

Invalid: The user is prompt with an error message

Result: Valid: Password will be changed

Invalid: The record is not updated to password

**Conclusion:** Both the valid and invalid results are tested. Output tally with the required result hence the test is successfulUser password is change by admin.



**Test case: 3**

Objectives: Test for entering the name

Test Data: Valid: Only letters are allowed

Invalid: Characters other than letters

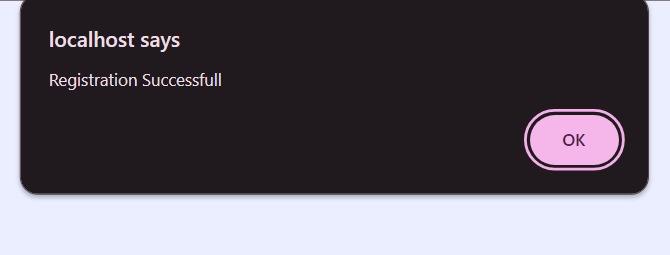
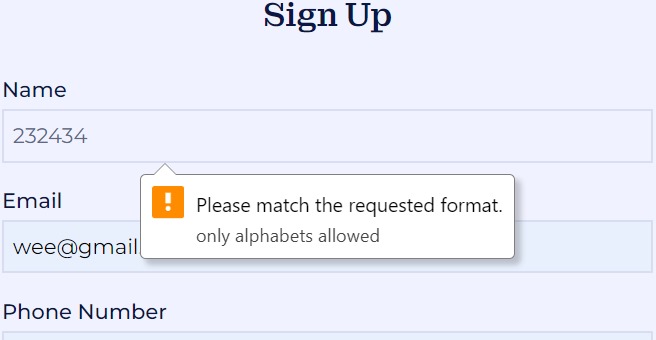
Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

**Conclusion:** Both the valid and invalid results are tested. Output tally with the require results hence the test is successful.



**Test case: 3**

Objectives: Test for entering mobile number

Test Data: Valid: Only numbers are allowed and must contain only 10 characters.

Invalid: Characters other than numbers.

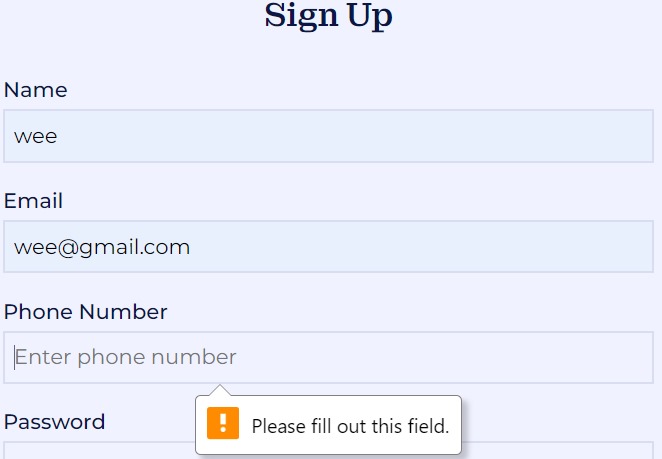
Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

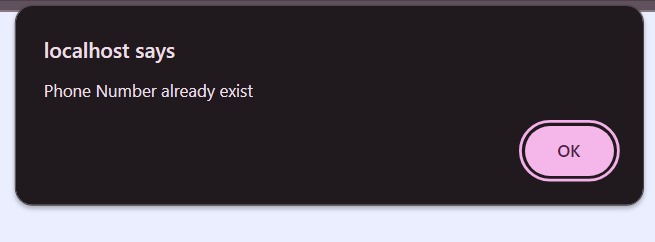
Result: Valid: Record will be saved

Invalid: The record will not saved

**Conclusion:** Both the valid and invalid results are tested. Output tally with the require results hence the test is successful







**Test case:5**

Objectives: Test for valid password

Test Data: Password should match with the entered confirm password.

Invalid- Entered password different from confirm password.

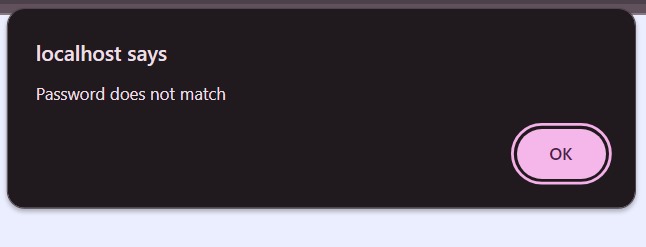
Output: Valid: logging in succesfully

Invalid: Show the error message

Result: Valid: The user was allowed to continue

Invalid: The user is prompted with an error message and restricted to continue

**Conclusion**: Both the valid and invalid results are tested. Output matches with the required result hence the test case is successful



**Test case: 6**

Objectives: Test for already registered account

Test Data: Valid: user should use new email

Invalid: user has used already registered email

Output: Valid: account should be registered

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

**Conclusion:** Both the valid and invalid results are tested. Output tally with the require results hence the test is successful.



**Test case:7**

Objectives: Test for Payment process

Test Data: Account number: should contain only numbers and should contain 10-16 digits.

Invalid- Use of invalid Account number

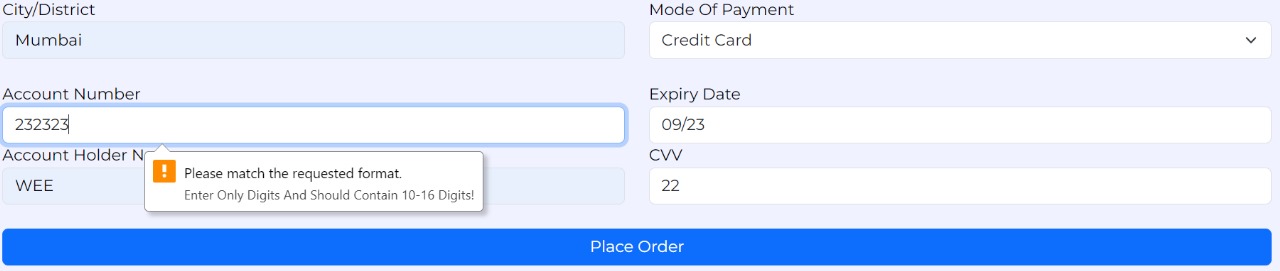
Output: Valid: Continuing for making payment for product

Invalid: Show the error message

Result: Valid: The user was allowed to continue

Invalid: The user is prompted with an error message and restricted to continue

**Conclusion:** Both the valid and invalid results are tested. Output matches with the required result hence the test case is successful.



**Test case:8**

Objectives: Test for Payment process

Test Data: Account Name: should contain alphabets.

Invalid- Use of invalid Account Name.

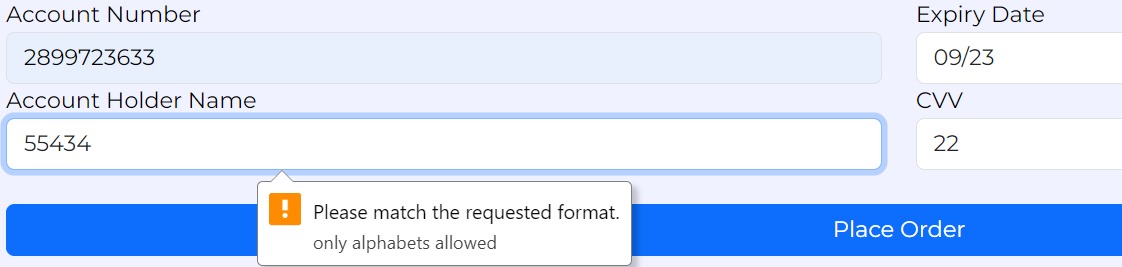
Output: Valid: Continuing for making payment for product

Invalid: Show the error message

Result: Valid: The user was allowed to continue

Invalid: The user is prompted with an error message and restricted to continue

**Conclusion:** Both the valid and invalid results are tested. Output matches with the required result hence the test case is successful.



**Test case:9**

Objectives: Test for Payment process

Test Data: CVV: should contain 3-4 digits

Invalid- Use of invalid CVV number.

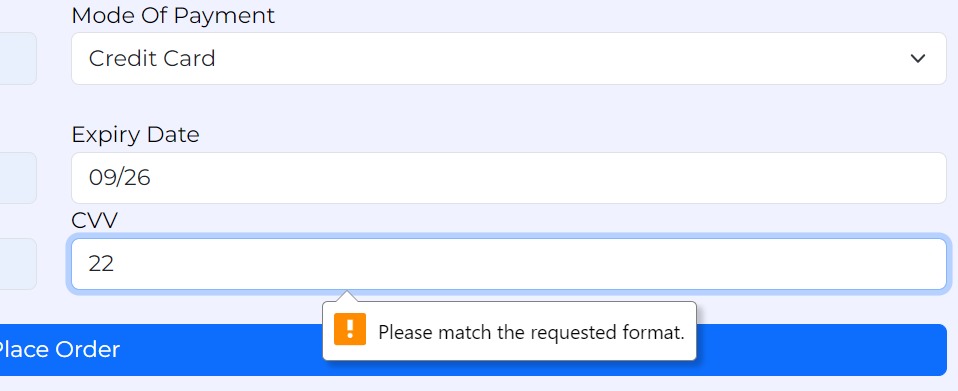
Output: Valid: Continuing for making payment for product

Invalid: Show the error message

Result: Valid: The user was allowed to continue

Invalid: The user is prompted with an error message and restricted to continue

**Conclusion:** Both the valid and invalid results are tested. Output matches with the required result hence the test case is successful



**Test case:10**

Objectives: Test for Payment process

Test Data: Expiry date: the date shouldn't be in the past, should be in future

Invalid- Use of invalid date.

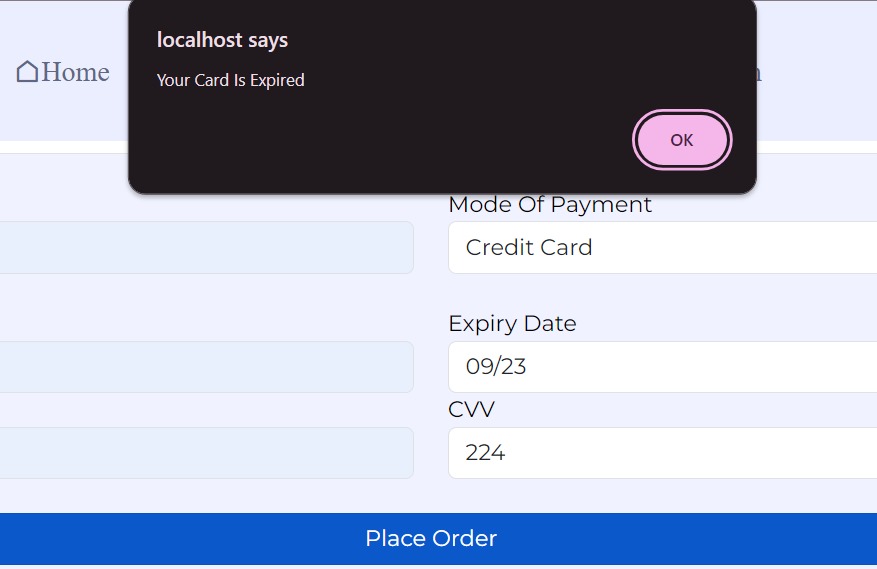
Output: Valid: Continuing for making payment for product

Invalid: Show the error message

Result: Valid: The user was allowed to continue

Invalid: The user is prompted with an error message and restricted to continue

**Conclusion:** Both the valid and invalid results are tested. Output matches with the required result hence the test case is successful



BIBLIOGRAPHY

* [chat gpt - Search (bing.com)](https://www.bing.com/search?q=chat+gpt&form=ANSPH1&refig=41439ce7b76544949a73903197354f7d&pc=HCTS&sp=1&ghc=1&lq=0&qs=HS&pq=chat&sc=10-4&cvid=41439ce7b76544949a73903197354f7d)
* [w3schools - Search (bing.com)](https://www.bing.com/search?q=w3schools&filters=ufn%3a%22W3Schools%22+sid%3a%22895f65dc-20b1-1b52-141a-7e440b8718f0%22&asbe=LS&qs=MB&pq=w3s&sc=10-3&cvid=19B220D327A647F384EABFDAF8EED103&FORM=QBRE&sp=1&ghc=1&lq=0)
* [tutorialspoint - Search (bing.com)](https://www.bing.com/search?pglt=41&q=tutorialspoint&cvid=6a7149b68b3841d5a6d22a9d71bc6291&gs_lcrp=EgZjaHJvbWUqBggAEEUYOzIGCAAQRRg7MgYIARBFGDkyBggCEAAYQDIGCAMQABhAMgYIBBAAGEAyBggFEAAYQDIGCAYQABhAMgYIBxAAGEAyBggIEAUYQNIBCDUwMzBqMGoxqAIAsAIA&FORM=ANNTA1&PC=HCTS)