

<b>Student Details ( Student should fill the content)</b>				
Name				
Student ID	KD/BSCSD/18/			
<b>Scheduled unit details</b>				
Unit code	CIS6003			
Unit title	Advanced Programming			
Unit enrolment details	Year	3		
Lecturer	Ms. Ruvani Jayaweera			
Mode of delivery	Full Time			
<b>Assignment Details</b>				
Nature of the Assessment	<b>Course work 100%</b>			
Topic of the Case Study	<b>Online reservation System</b>			
Learning Outcomes covered	<b>1,2,3</b>			
Word count	4000			
Due date / Time	11th September 2024			
Extension granted?	Yes	No	Extension Date	
Is this a resubmission?	Yes	No	Resubmission Date	
<b>Declaration</b>				
I certify that the attached material is my original work. No other person's work or ideas have been used without acknowledgement. Except where I have clearly stated that I have used some of this material elsewhere, I have not presented it for examination/assessment in any other course or unit at this or any other institution				
Name/Signature			Date	11th September 2024
<b>Submission</b>				
Return to:				
<b>Result</b>				
Marks by 1 <sup>st</sup> Assessor		Name & Signature of the 1 <sup>st</sup> Assessor		<b>Agreed Mark</b>
Marks by IV:		Name & Signature of the IV		
<b>For Office use only (hard copy assignments)</b>				
Receipt date		Received by		

<b>STUDENT NAME:</b>	<b>STUDENT NUMBER:</b>	
<b>Module Number &amp; Title:</b> CIS6003 - Advanced Programming	<b>Semester:</b> 1st	
<b>Assignment Type &amp; Title:</b> Online reservation System <b>For student use:</b> <i>Critical feedback on the individual progression towards achieving the assignment outcomes</i>		
<b>For the Assessors' feedback</b> Indicate the Task number strength and Weaknesses and the marks for each task		
<b>Task/ Questi on No</b>	<b>Strengths (1<sup>st</sup> Assessor)</b>	<b>Weaknesses (1<sup>st</sup> Assessor)</b>
	<b>Strengths (2<sup>nd</sup> Assessor)</b>	<b>Weaknesses (2<sup>nd</sup> Assessor)</b>

<b>Marks</b>				
<b>Task /Question No</b>	<b>Marks by 1<sup>st</sup> Assessor</b>	<b>Marks by 2<sup>nd</sup> Assessor</b>	<b>Marks by IV (if any)</b>	<b>IV comments (If Any)</b>
<b>Total Marks</b>				
<b>Name and the Signature of the 1st Assessor</b>				<b>Date:</b>
<b>Name &amp; Signature of the 2nd Assessor :</b>				<b>Date :</b>
<b>Name &amp; Signature of the IV: (If any)</b>				<b>Date :</b>
<b>Areas for Improvement(1st Assessor)</b>			<b>Areas for Improvement(2<sup>nd</sup> Assessor)</b>	

# Plagiarism Report

---

**ORIGINALITY REPORT**

---

<b>2%</b> SIMILARITY INDEX	<b>2%</b> INTERNET SOURCES	<b>0%</b> PUBLICATIONS	<b>2%</b> STUDENT PAPERS
-------------------------------	-------------------------------	---------------------------	-----------------------------

---

**PRIMARY SOURCES**

---

- |          |   |               |
|----------|---|---------------|
| <b>1</b> | <b>Submitted to University of Wales Institute, Cardiff</b><br>Student Paper | <b>1%</b>     |
| <b>2</b> | <b>Submitted to Colorado Technical University</b><br>Student Paper          | <b>1%</b>     |
| <b>3</b> | <b>www.almuslimstore.com</b><br>Internet Source                             | <b>&lt;1%</b> |
| <b>4</b> | <b>Submitted to CINEC Campus</b><br>Student Paper                           | <b>&lt;1%</b> |
-

## Table of Contents

Plagiarism Report.....	iv
Table of Figures .....	vi
Acknowledgment.....	viii
Introduction.....	1
<b>Task A .....</b>	2
System Design of ABC Restaurant.....	2
Admin Requirements .....	2
Staff Requirements.....	3
Customer Requirements.....	3
Use Case Diagram.....	4
Class Diagram.....	5
Sequence Diagrams.....	6
<b>Task B .....</b>	9
<b>System Design Patterns.....</b>	9
<b>Overview of the ABC Restaurant Web Application .....</b>	13
<b>Web Application Structure .....</b>	14
Task C .....	36
ABC Restaurant Management System Testing plans .....	36
Task 4.....	52
Conclusion .....	55
References.....	56

## Table of Figures

Figure 1 Class Diagram .....	4
Figure 2 Class Diagram .....	5
Figure 3 Login Sequence Diagram .....	6
Figure 4 Register Sequence Diagram .....	7
Figure 5 Product Add, Update, Delete Sequence Diagram .....	8
Figure 6 Folder Structure .....	10
Figure 7 DB Connection .....	11
Figure 8 Database schema .....	12
Figure 9 Login form and Controller.....	14
Figure 10 Registration form, Controller and welcome email .....	15
Figure 11 Welcome E mail code Structure .....	16
Figure 12 Menu Interface and Navigation bar with customer name .....	18
Figure 13 Show Menu Items and Cart option.....	19
Figure 14About Us Frontend and Backend .....	21
Figure 15 Service Interface and Backend .....	23
Figure 16 Frontend and Email acception of Reservation.....	24
Figure 17 Reservation servlet (add) and Email Backend.....	25
Figure 18 Gallery Frontend and Backend.....	26
Figure 19 Query Servlet and frontend and backend .....	27
Figure 20 admin Login interface and admin servlet .....	28
Figure 21Admin Controller Servlet .....	30
Figure 22 Item List withe search bar .....	31
Figure 23 Reservation report pdf .....	31
Figure 24 Reservation Generate backend code.....	32
Figure 25 View Sales report pdf download button and backend code .....	33
Figure 26 Offer Management.....	34
Figure 27 Order Management.....	34
Figure 28 Gallery Management .....	34
Figure 29 Staff Loggin, Register and Dashboard .....	36
Figure 30 Register Proof .....	37
Figure 31 Invalid email or password proof.....	38
Figure 32 Reservation Success Proof .....	39

Figure 33 Query Success Proof.....	40
Figure 34 Add items to cart proof.....	41
Figure 35 Makes payment successfully proof .....	42
Figure 36 Order Conformation Message .....	43
Figure 37 Admin Search Products .....	44
Figure 38 Product Added Successfully Proof.....	45
Figure 39 Product Update Proof .....	46
Figure 40 Product Delete Proof .....	47
Figure 41 Reservation Accept Proof.....	49
Figure 42 Reply email proof .....	50
Figure 43 Staff Login Proof.....	51

## Acknowledgment

I extend my deepest gratitude to my family members for their unwavering support and encouragement throughout my educational journey. Their guidance and wisdom have undoubtedly been instrumental in shaping my path. I am also indebted to Ms. Ruvani Jayaweera, my lecturer, whose invaluable insights, and teachings on Advanced Programming have been pivotal in successfully completing this assignment. Additionally, I am thankful to my friends for their collaboration and shared educational insights, which have enriched my learning experience.

## Introduction

The goal of the ABC Restaurant project is to create a fully functional web application that supports a chain of restaurants by utilizing a database. Important aspects of the system include role-based access control for administrators and staff, online reservations, product and offer administration, and client registration. The goal is to guarantee effective corporate operations and a flawless customer experience. Administrators, employees, and consumers are among the various user categories that the system is intended to support, each with distinct functionalities and access rights.

Throughout the course of a project, version control with Git and GitHub promotes feature integration, logs changes, and makes collaboration easier. To provide the ABC Restaurant with a reliable and scalable solution, this project places a strong emphasis on important aspects including system design, feature development, testing, and version control. The project exhibits high proficiency in web development and database management by following best practices and integrating sophisticated features like reservation management and product handling.

## Task A

### System Design of ABC Restaurant

The "ABC Restaurant" chain, which operates in several cities across Sri Lanka, is planning to introduce a web-based platform to optimize its operations. The system is designed for three main types of users: Administrators, Restaurant Staff, and Customers. Each user group will have specific roles and access to different functions within the platform. Customers will be able to explore services, book tables, make payments, and ask questions via the website. Restaurant staff will handle reservations and respond to customer inquiries, while Administrators will have oversight of users, services, and reports.

### Purpose:

The primary aim is to develop a user-friendly and interactive system that facilitates smooth service delivery for both the business and its customers. The system will be secure, robust, and scalable to support multiple restaurant branches efficiently.

### Admin Requirements

1. **User Management:** Admins can create, update, and remove user accounts, managing access levels and permissions for restaurant staff and customers.
2. **Food Management:** Admins can add, update, and remove items from the menu, including details like pricing, availability, and descriptions.
3. **Reservation and Order Management:** Admins can monitor, update, or cancel customer reservations and orders, ensuring smooth operations across the system.
4. **Payment Monitoring:** Admins can track all payments made through the system, reviewing transactions and managing refunds if necessary.
5. **Query and Feedback Management:** Admins can review and respond to customer inquiries and feedback submitted through the website.
6. **Report Generation:** Admins can generate reports in PDF format for sales, reservations, and overall system usage to assist with business decision-making.
7. **Access Control:** Admins can manage who has access to different features and functionalities, ensuring that only authorized personnel can perform certain actions.

## Staff Requirements

1. **Order and Reservation Management:** Staff can view, manage, and update customer reservations and orders, ensuring that bookings and orders are processed smoothly. They can adjust seating arrangements, confirm orders, and notify customers about the status of their reservations or orders.
2. **Payment and Order Processing:** Staff are responsible for processing payments from customers, ensuring that orders are correctly entered, and confirming that payments are completed. This includes handling various payment methods and generating receipts for customers.
3. **Query Handling:** Staff can respond to customer queries, whether they relate to reservations, orders, or general questions. This ensures prompt and efficient communication with customers, helping to resolve any issues or provide additional information as needed.

## Customer Requirements

1. **Registration and Login:** Customers can create an account by registering their details and securely log in to the system. This allows them to access personalized services such as order history, saved preferences, and reservation details.
2. **Food Ordering:** Customers can browse the menu, select items, customize their orders (e.g. special requests), and place food orders for dine-in, takeaway, or delivery through the platform.
3. **Order and Reservation:** Customers can book tables at their preferred restaurant location, choose their preferred dining time, and manage their reservations. They can also track the status of their current orders and view past orders.
4. **Payments Processing:** Customers can securely make payments through the system using various payment methods. This includes paying for food orders or reservation deposits.
5. **Query and Feedback Submission:** Customers can easily submit queries, complaints, or feedback about their experience through the platform, helping the restaurant improve its services.

## Use Case Diagram

The use case diagram will represent how different users (Admin, Restaurant Staff, and Customers) interact with the system. Each user has specific roles, and the diagram will demonstrate how they engage with the system's functionalities.

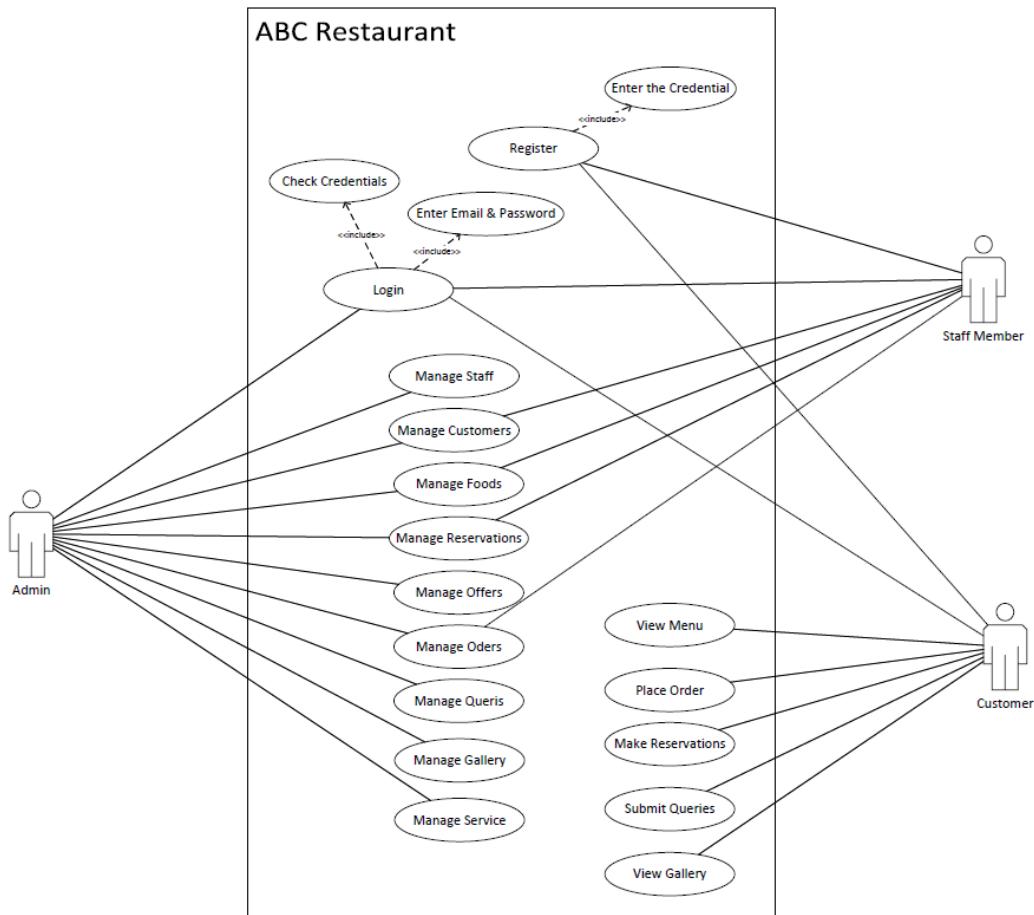


Figure 1 Class Diagram

The diagram also illustrates system interactions such as credential verification during the login process for all user types. Each user type (Admin, Staff Member, and Customer) has specific access rights and roles, indicating a role-based access system.

- Multiplicity: Customers can place multiple orders and make several reservations.
- Aggregation: There is a relationship between a branch and the variety of food items or services it offers.
- Composition: Customers are essential for the existence of orders, and payments are inherently tied to the presence of an order.

## Class Diagram

Users are responsible for managing their account details, including personal information, contact data, and login credentials, ensuring secure and personalized access to the system. The Product component manages detailed information about each item offered by the restaurant, including the product name, price, description, and category, allowing customers to explore the menu more effectively.

The Service and Promotion components play key roles in managing the restaurant's services and promotions, helping administrators design and implement special offers, deals, or new service offerings to attract customers.

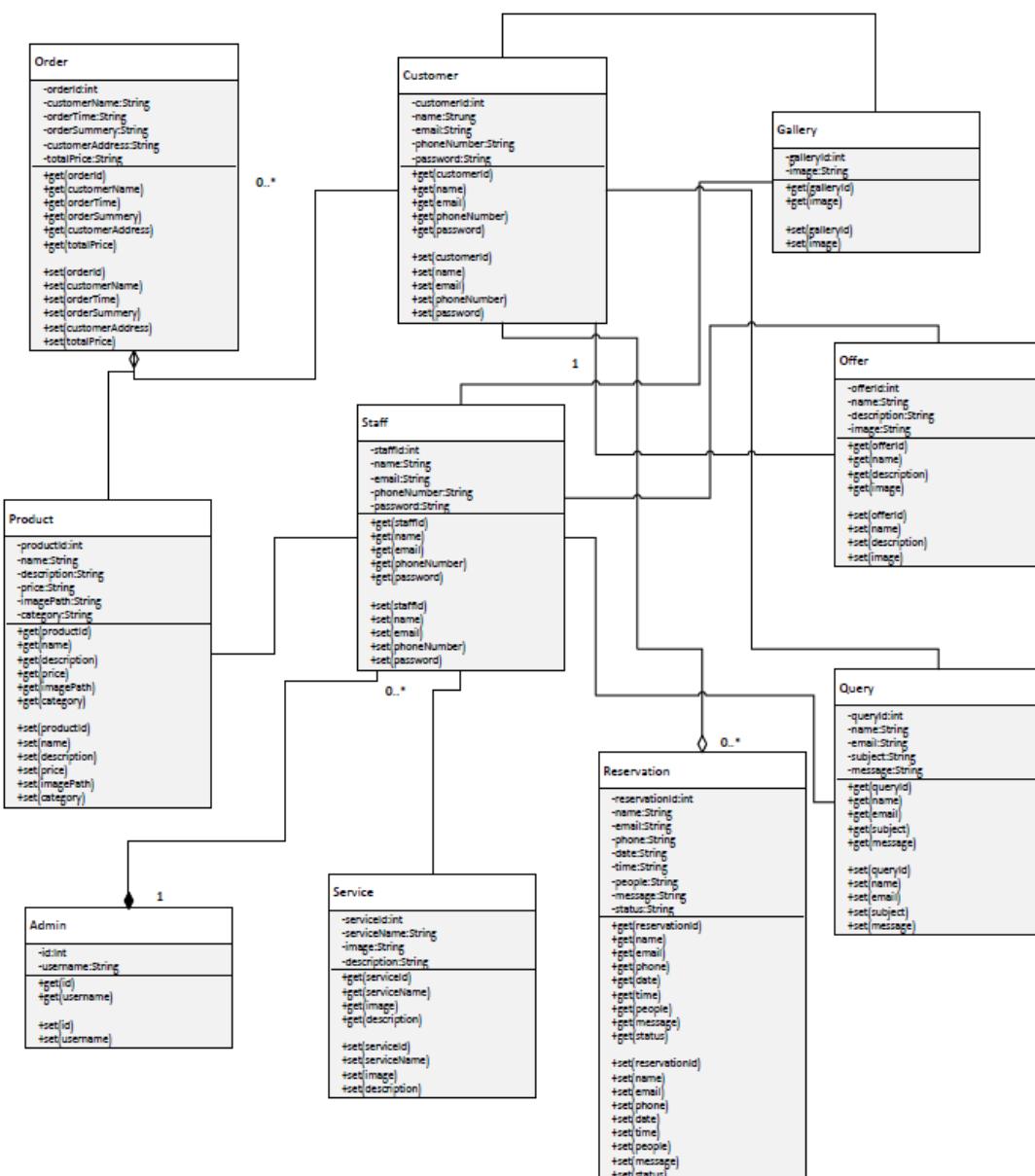


Figure 2 Class Diagram

## Sequence Diagrams

- **Login Sequence**

The user navigates to the login page, enters their credentials, and submits the login form, depending on whether they are a client, staff member, or administrator. The authentication server receives the credentials after that for verification. To confirm the user's login details and ascertain their role, the server sends a query to the database. The user is given access and taken to the relevant dashboard according to their role (Admin, Staff, or Customer) if the credentials are valid. If the user's credentials are invalid, the system will notify them with an error notice and ask them to enter their information again.

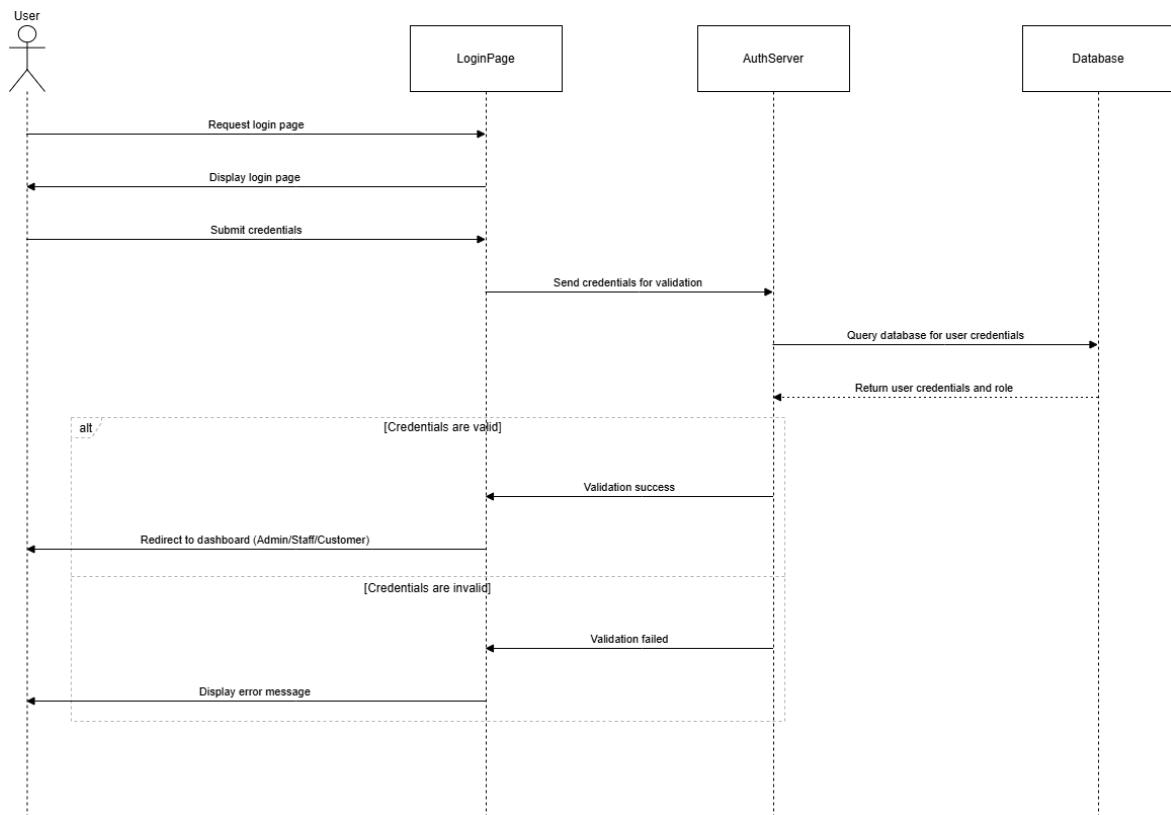


Figure 3 Login Sequence Diagram

- **Register Sequence**

When a user goes to the registration page, they complete out the form by entering their name, email address, username, and password and clicking the submit button. In order to confirm whether the email address or username is already stored in the database, the system transmits this data to the authentication server. The user is notified by the system and the registration process is stopped if it is discovered that the email address or username is already in use. The user is successfully registered, and a new account is generated if the details are genuine and unique. After displaying a confirmation message, the system either takes the user to the login page or, if something goes wrong, displays an error message.

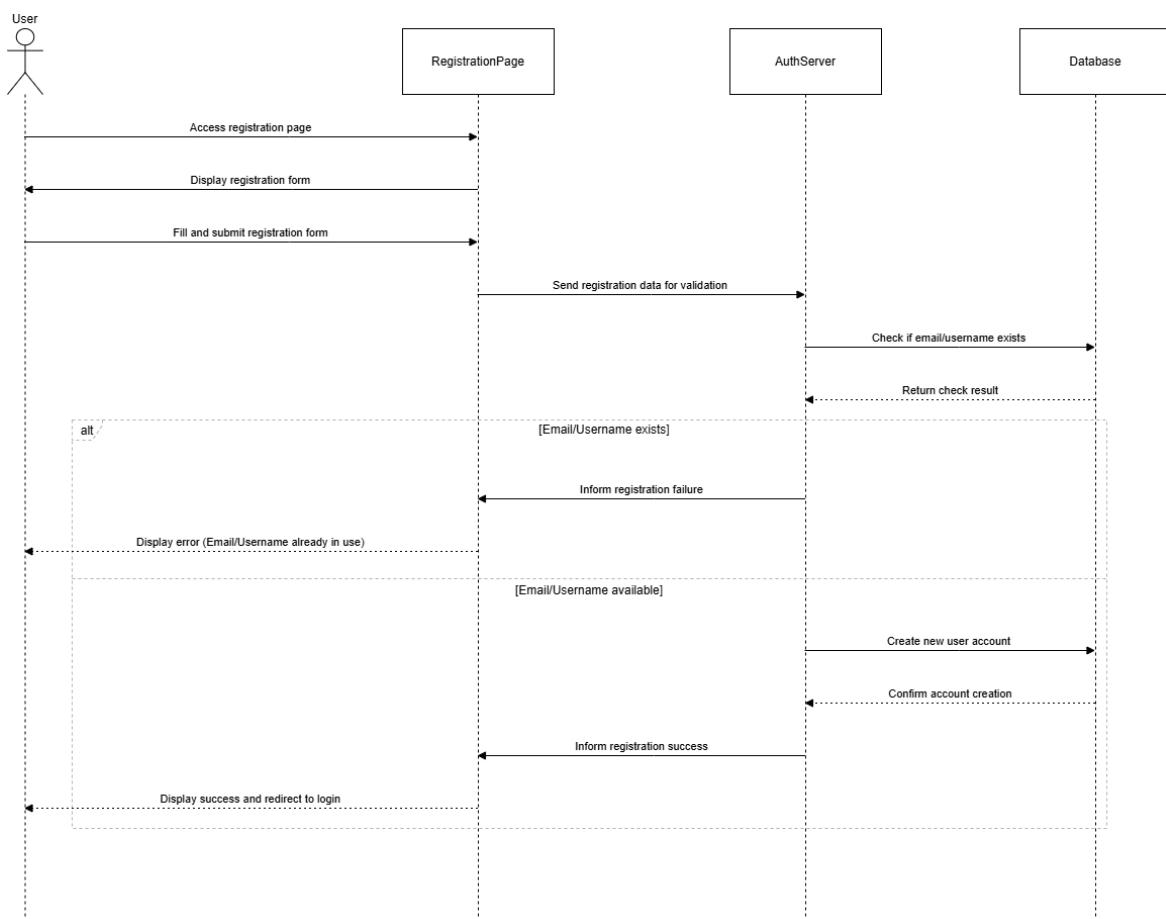


Figure 4 Register Sequence Diagram

- **Product Add, Update, Delete Sequence Diagram**

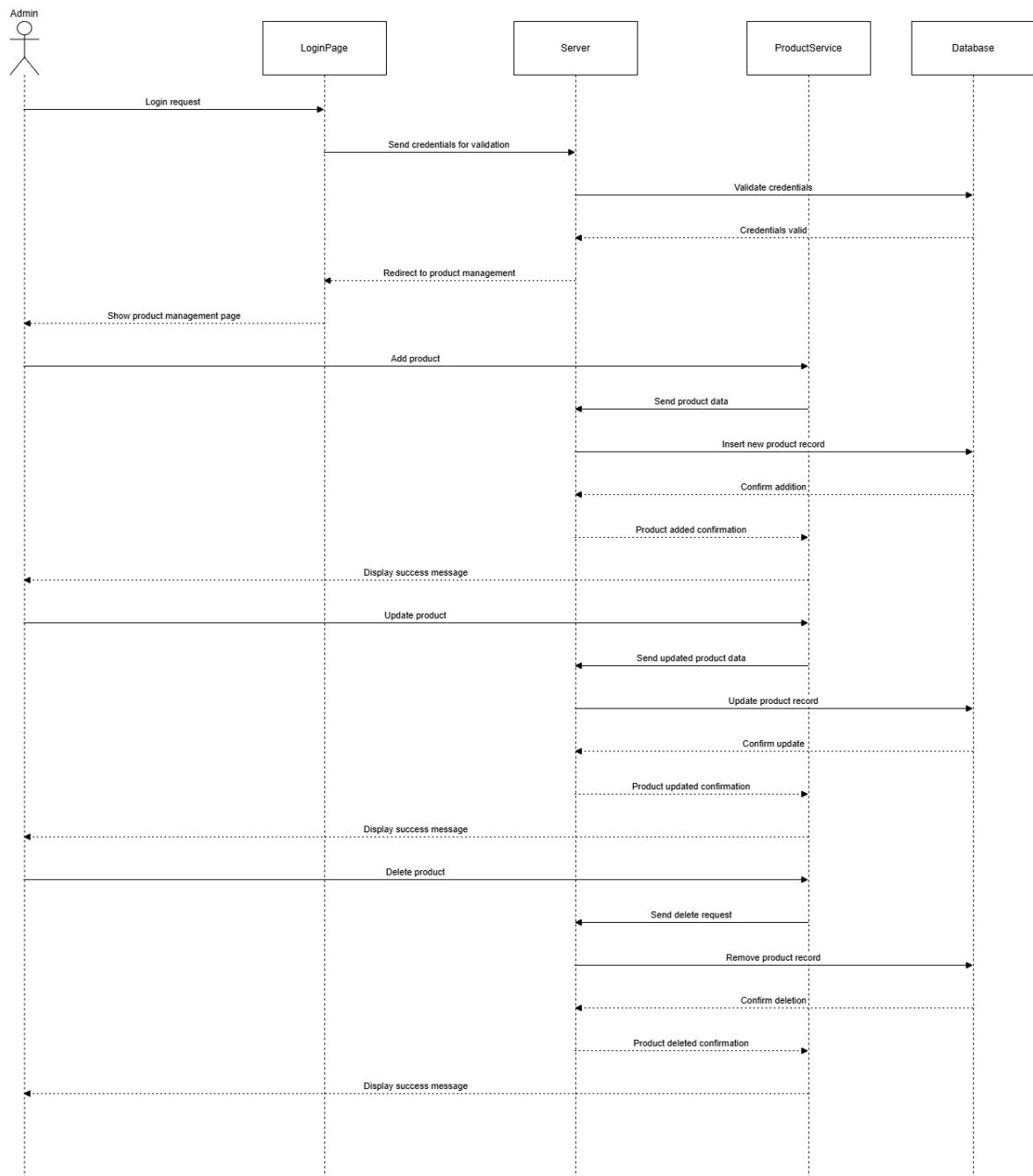


Figure 5 Product Add, Update, Delete Sequence Diagram

This is also the case with other CRUDS

## Task B

### System Design Patterns

Creating the online application for "ABC Restaurant" involves using different design patterns to achieve a system architecture that is not only efficient but also scalable and easy to maintain.

The architecture of the ABC Restaurant application is built upon several essential layers, each designed to handle specific responsibilities to ensure efficient communication and scalability:

- **Model Layer:** This layer consists of the core business entities that the application operates upon, such as Reservation, Customer, and Service. These entities represent real-world objects and contain attributes and behaviors that align with the system's business logic.
- **DAO Layer (Data Access Object):** This layer provides an abstraction over the database operations, enabling efficient interaction with the MySQL database. It serves as a bridge between the application and the database, offering methods for creating, reading, updating, and deleting data. The DAO layer ensures that data access operations, such as fetching reservation records or updating customer details, are performed seamlessly without exposing the internal workings of the database to the higher layers.
- **Controller Layer:** Acting as an intermediary between the user interface and the underlying services, the controller layer is responsible for handling incoming user requests from the front-end (such as JSP pages). It processes these requests and routes them to the appropriate service layer methods.
- **Service Layer:** This layer is tasked with managing the business logic of the application. It handles critical processes such as managing customer reservations, answering inquiries, and providing various services.

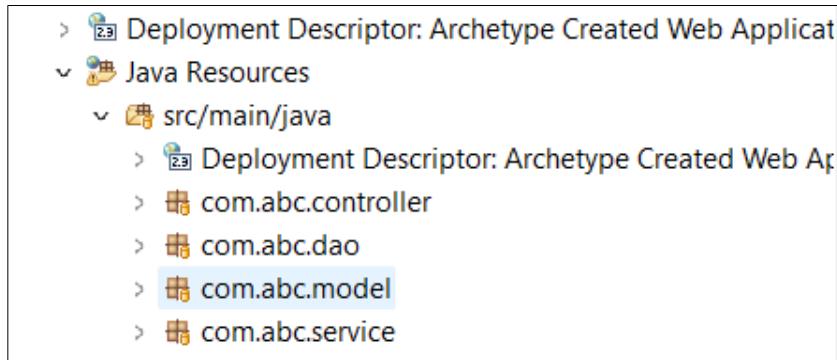


Figure 6 Folder Structure

When developing a complex system like the "ABC Restaurant" web-based application, it's crucial to use design patterns to improve the structure, maintainability, and scalability of the application. Design patterns are proven solutions to common problems that arise during software development.

- **Singleton Pattern**

The Singleton Pattern ensures that a class has only one instance throughout the system and provides a global point of access to that instance. This pattern is useful when you want to manage shared resources like database connections, logging, or configuration settings. (geeks., 2024)

**Application in the ABC Restaurant System:** The Singleton Pattern can be applied to manage the database connection for the application. Since every part of the system (reservations, queries, payments) will need to interact with the database, ensuring that only one instance of the database connection exists at any given time helps in managing resources efficiently. This ensures consistent access to the database without creating multiple, unnecessary connections.

```

public class DBConnection {

    private static final String DB_URL = "jdbc:mysql://localhost:3306/product_abc";
    private static final String DB_USER = "root";
    private static final String DB_PASSWORD = "";

    private static DBConnection instance;
    private Connection connection;

    private DBConnection() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            this.connection = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }

    public static DBConnection getInstance() {
        if (instance == null) {
            synchronized (DBConnection.class) {
                if (instance == null) {
                    instance = new DBConnection();
                }
            }
        }
        return instance;
    }
}

```

Figure 7 DB Connection

- **Factory Pattern**

The Factory Pattern enables the creation of families of related objects without revealing the precise class to the client. It is used to construct objects without disclosing the instantiation logic to the client. This is especially helpful when dealing with different object types that have identical properties but different implementations within the system. (Aibin, 2024)

**Application in the ABC Restaurant System:** When interacting with the various user types (Admin, Restaurant Staff, Customers), the Factory Pattern can be used. The Factory Pattern enables the system to dynamically produce the appropriate type of user based on the role, as opposed to hardcoding how these user objects are created. Because the creation mechanism is still centralized and controllable, this increases the system's flexibility in the event that other user kinds are introduced later.

## Database Structure

The database schema has been thoughtfully structured to meet the restaurant's operational requirements, ensuring data integrity and reducing redundancy. Below is an overview of the key components of the database

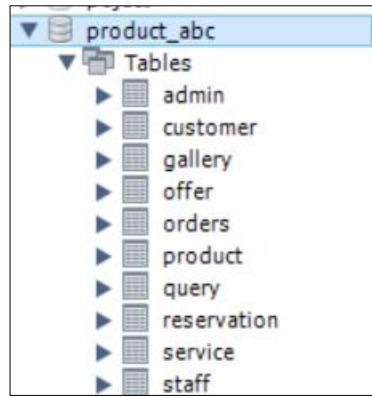


Figure 8 Database schema

## Overview of the ABC Restaurant Web Application

The ABC Restaurant Management System is an advanced, web-based platform crafted to enhance and automate the routine operations of a restaurant. Powered by core Java for backend processes, JSP for dynamic content generation, and MySQL for database management, this system effectively handles tasks like customer reservations, order management, and staff coordination. The front-end interface designed using HTML5, CSS, Bootstrap, and JavaScript, ensures a responsive and intuitive user experience across various devices.

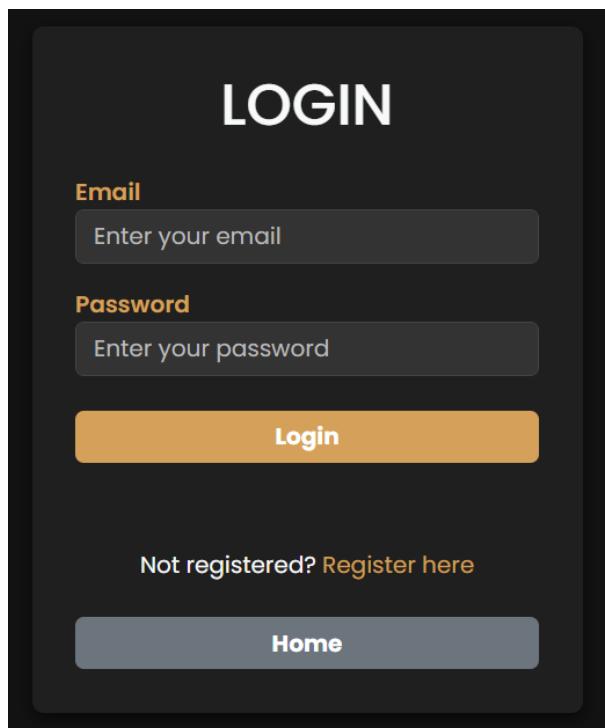
The system includes features like real-time email notifications, providing customers with immediate updates on their orders and reservations. Additionally, the platform offers role-based access control, where administrators oversee the restaurant's full operations, staff members handle customer orders and inquiries, and customers can make reservations, view menus, submit queries, and place their orders.

The system generates comprehensive reports, providing administrators with important insights into daily sales performance, customer reviews, and reservation patterns, aiding in making informed, data-driven decisions. This architecture enables seamless updates and expansions without affecting core operations, making it an ideal choice for modern restaurant management systems.

## Web Application Structure

- **Customer Login**

The customer login interface for the ABC Restaurant Management System allows customers to securely access their accounts. After entering valid login credentials, the system verifies the customer's details and directs them to their personalized dashboard. This dashboard provides access to features such as managing reservations, checking orders, and submitting queries, ensuring a smooth and centralized user experience tailored specifically for customer interactions.



```
private void loginCustomer(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String email = request.getParameter("email");
    String password = request.getParameter("password");

    Customer customer = customerService.loginCustomer(email, password);
    if (customer != null) {
        request.getSession().setAttribute("customer", customer);
        response.sendRedirect("index");
    } else {
        request.setAttribute("error", "Invalid email or password");
        request.getRequestDispatcher("WEB-INF/view/loginCustomer.jsp").forward(request, response);
    }
}
```

Figure 9 Login form and Controller

## Register

- **Customer Register**

The form includes input fields for Name, Email, Phone Number, and Password, followed by a "Register" button to submit the information. The form is clean, user-friendly, and aims to make the registration process straightforward and efficient for customers.

After the customer registers on the website, an automated welcome email is sent to their email address. This email confirms their successful registration and provides a warm greeting, thanking them for joining ABC Restaurant

The figure consists of two side-by-side screenshots. On the left is a dark-themed registration form titled 'REGISTER'. It has four input fields: 'Name' (placeholder 'Enter your name'), 'Email' (placeholder 'Enter your email'), 'Phone Number' (placeholder 'Enter your phone number'), and 'Password' (placeholder 'Enter your password'). Below these is an orange 'Register' button. At the bottom is a link 'Already a Member? Login here'. On the right is an email inbox showing a welcome message from 'abcrestaurantsl@outlook.com' to the recipient. The email subject is 'Welcome to ABC Restaurant!' and the body contains a greeting, a thank you message, and a note about being excited to have the user onboard. It is signed off by 'The ABC Team'.

```

private void registerCustomer(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String phoneNumber = request.getParameter("phoneNumber");
    String password = request.getParameter("password");

    Customer customer = new Customer();
    customer.setName(name);
    customer.setEmail(email);
    customer.setPhoneNumber(phoneNumber);
    customer.setPassword(password);

    customerService.registerCustomer(customer);

    EmailUtil.sendWelcomeEmail(email, name);
    response.sendRedirect("customer?action=login");
}
  
```

Figure 10 Registration form, Controller and welcome email

```

public static void sendWelcomeEmail(String recipientEmail, String recipientName) {
    try {
        Session session = getSession();

        Message message = new MimeMessage(session);
        message.setFrom(new InternetAddress(SENDER_EMAIL));
        message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(recipientEmail));
        message.setSubject("Welcome to ABC Restaurant!");

        String emailContent = "<h1>Welcome to ABC Restaurant</h1>" +
            + "<p>Dear " + recipientName + ",</p>" +
            + "<p>Thank you for registering with ABC Restaurant.</p>" +
            + "<p>We are excited to have you onboard!</p>" +
            + "<br><p>The ABC Team</p>";

        message.setContent(emailContent, "text/html");

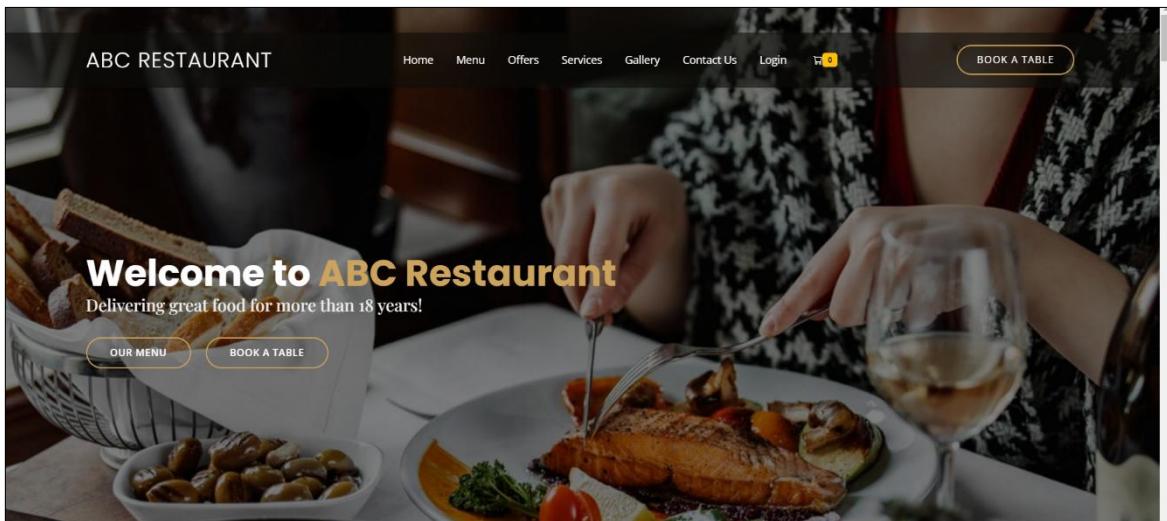
        Transport.send(message);
        System.out.println("Welcome email sent to " + recipientEmail);
    } catch (MessagingException e) {
        e.printStackTrace();
    }
}

```

Figure 11 Welcome E mail code Structure

- **Customer Main Interface**

The navigation bar at the top offer's links to sections like Home, Menu, Offers, Services, Gallery, and Contact Us. There are also prominent buttons for booking a table and viewing the menu, inviting users to interact with the site.



```

<header id="header" class="fixed-top d-flex align-items-cente">
  <div class="container-fluid container-xl d-flex align-items-center justify-content-lg-between">
    <c:choose>
      <c:when test="${sessionScope.customer != null}">
        <h1 class="Logo me-auto me-lg-0"><a href="index">Welcome, ${sessionScope.customer.name}!</a></h1>
      </c:when>
      <c:otherwise>
        <h1 class="Logo me-auto me-lg-0"><a href="index">ABC Restaurant</a></h1>
      </c:otherwise>
    </c:choose>

    <nav id="navbar" class="navbar order-last order-lg-0">
      <ul>
        <c:choose>
          <c:when test="${sessionScope.customer != null}">
            <li class="nav-item">
              <a class="nav-link" href="index">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#menu">Menu</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#specials">Offers</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#events">Services</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#gallery">Gallery</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#contact">Contact Us</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="customer?action=logout">Logout</a>
            </li>
          </c:when>
          <c:otherwise>
        </c:choose>
      </ul>
    </nav>
  </div>
</header>

```

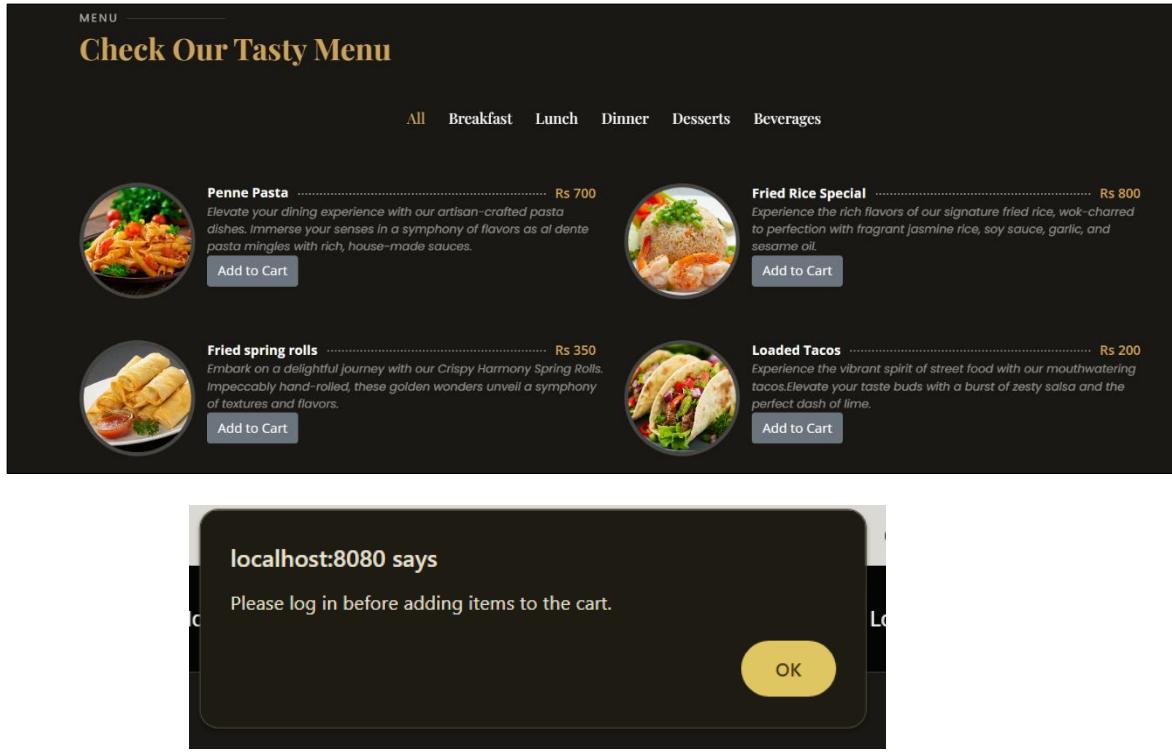
- **Menu Section**

The "Menu" section of the ABC Restaurant website allows customers to browse various food items, categorized by meal types such as breakfast, lunch, dinner, desserts, and beverages. Each menu item is displayed with an image, description, price, and an "Add to Cart" button.

### User Interaction and Behavior:

**Before Login:** If a customer is not logged in, they can only view the menu items, but cannot add them to the cart. The "Add to Cart" buttons are shown in a grey color, indicating they are disabled.

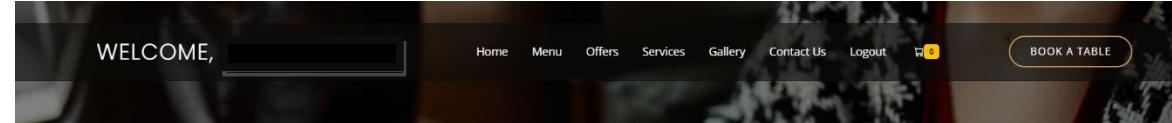
**After Login:** Once a customer logs in, the "Add to Cart" buttons become active and change to a golden color. This signifies that the customer can now interact with the menu and add items to their cart for ordering.



The screenshot shows the restaurant's menu page. At the top, there's a navigation bar with links for Home, Menu, Offers, Services, Gallery, Contact Us, Logout, and a Book A Table button. Below the navigation is a header "Check Our Tasty Menu". Underneath are category filters: All, Breakfast, Lunch, Dinner, Desserts, and Beverages. The main content area displays four menu items with their descriptions and "Add to Cart" buttons:

- Penne Pasta** ..... Rs 700  
Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.  
**Add to Cart**
- Fried Rice Special** ..... Rs 800  
Experience the rich flavors of our signature fried rice, wok-charred to perfection with fragrant jasmine rice, soy sauce, garlic, and sesame oil.  
**Add to Cart**
- Fried spring rolls** ..... Rs 350  
Embark on a delightful journey with our Crispy Harmony Spring Rolls. Impeccably hand-rolled, these golden wonders unveil a symphony of textures and flavors.  
**Add to Cart**
- Loaded Tacos** ..... Rs 200  
Experience the vibrant spirit of street food with our mouthwatering tacos. Elevate your taste buds with a burst of zesty salsa and the perfect dash of lime.  
**Add to Cart**

A modal window titled "localhost:8080 says" is overlaid on the page, displaying the message "Please log in before adding items to the cart." with an "OK" button.



The second screenshot shows the same menu interface, but the top navigation bar now includes a "WELCOME, [REDACTED]" message, indicating a logged-in user. The rest of the page content is identical to the first screenshot.

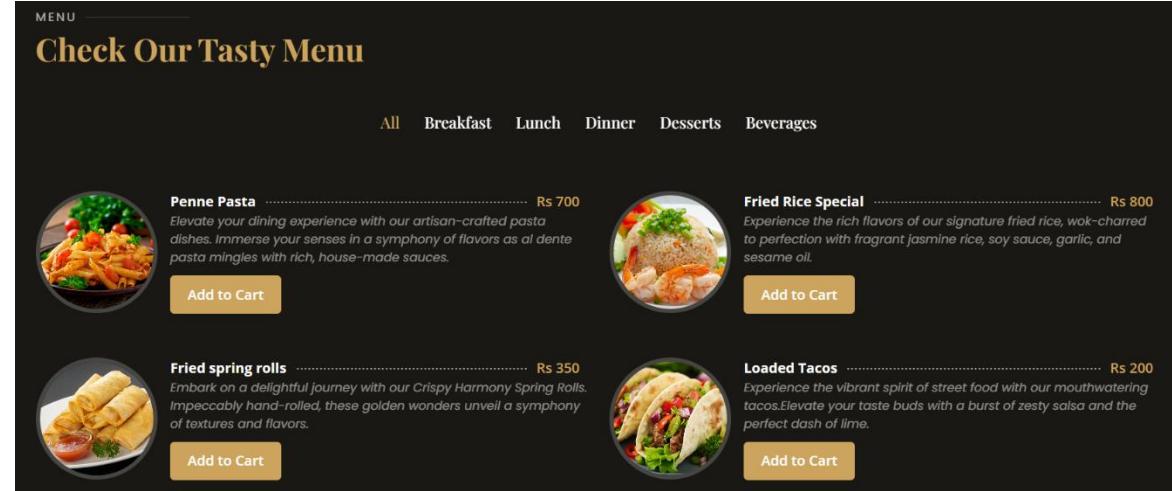


Figure 12 Menu Interface and Navigation bar with customer name

```

<section id="menu" class="menu section-bg">
<div class="container" data-aos="fade-up">

<div class="section-title">
  <h2>Menu</h2>
  <p>Check Our Tasty Menu</p>
</div>

<div class="row" data-aos="fade-up" data-aos-delay="100">
  <div class="col-lg-12 d-flex justify-content-center">
    <ul id="menu-filters">
      <li data-filter="#" class="filter-active">All</li>
      <li data-filter=".filter-Breakfast">Breakfast</li>
      <li data-filter=".filter-Lunch">Lunch</li>
      <li data-filter=".filter-Dinner">Dinner</li>
      <li data-filter=".filter-Dessert">Desserts</li>
      <li data-filter=".filter-Beverages">Beverages</li>
    </ul>
  </div>
</div>

<div class="row menu-container" data-aos="fade-up" data-aos-delay="200">
  <c:forEach var="product" items="${products}">
    <div class="col-lg-6 menu-item filter-${product.category}">
      
      <div class="menu-content">
        <a href="#">${product.name}</a><span>Rs ${product.price}</span>
      </div>
      <div class="menu-ingredients">
        ${product.description}
      </div>
    </div>
  </c:forEach>
</div>

```

```

<!-- Check if customer is logged in -->
<c:choose>
  <c:when test="${sessionScope.customer != null}">
    <!-- Add to Cart Form -->
    <form action="cart" method="post">
      <input type="hidden" name="productId" value="${product.productId}">
      <input type="hidden" name="quantity" value="1" /> <!-- Default quantity to 1 -->
      <input type="hidden" name="action" value="add">
      <button type="submit" class="add-to-cart-btn">Add to Cart</button>
    </form>
  </c:when>
  <c:otherwise>
    <!-- Show an alert if not logged in -->
    <button class="btn btn-secondary" onclick="alertLoginMessage()">Add to Cart</button>
  </c:otherwise>
</c:choose>

```

Figure 13 Show Menu Items and Cart option

In this food ordering system, customers can add meals to their cart by selecting the desired quantity. After reviewing their cart, they can enter their delivery address and proceed to checkout. Once the order is placed, a confirmation email is automatically sent to the customer's email address with the order details, including items, quantity, and total price.

Additionally, customers have the option to download their bill as a PDF after the order confirmation. This system makes the ordering process simple and efficient by automating order confirmation and providing downloadable receipts for convenience.

**Your Cart**

Product	Price	Quantity	Total	Actions
Penne Pasta	Rs 700	1	Rs 700.0	<button>Update</button> <button>Remove</button>
Fried Rice Special	Rs 800	1	Rs 800.0	<button>Update</button> <button>Remove</button>
<b>Total:</b>			<b>Rs 1500.0</b>	

Customer Name:

Address:

Card Number:

Expiry Month:  Expiry Year:

CVV:

**Proceed to Checkout** **Home**

**localhost:8080 says**  
Payment form submitted successfully!

**OK**

**Order Confirmation**  
Thank you for your order,   
Your order has been successfully placed.

**Return to Home** **Print Your Bill**

**Order Summary**

Customer Name:   
Order Time: Sun Sep 08 03:27:03 IST 2024  
Customer Address:

<b>Order Summary</b>	[Penne Pasta - Quantity: 1 - Total Price: Rs 700.0, Fried Rice Special - Quantity: 1 - Total Price: Rs 800.0]
<b>Total Price</b>	Rs 1500.0

**Thanks for your order!**

**Thank you for your order from ABC Restaurant!** [Inbox x]

**a** abcrestaurantsl@outlook.com to me ▾ 03:27 (1 minute ago)

**Thank You for Your Order!**

Dear

Thank you for placing an order with ABC Restaurant. We are thrilled to serve you!

**Order Summary:**

Items:

[Penne Pasta - Quantity: 1 - Total Price: Rs 700.0, Fried Rice Special - Quantity: 1 - Total Price: Rs 800.0]

**Total Price:** Rs 1500.0

**Delivery Address:**

If you have any questions about your order, feel free to reach out to us.  
We appreciate your business and hope to see you again soon!

Best regards,  
ABC Restaurant

## About Us

**About Us**

Welcome to ABC Restaurant, where culinary passion meets warm hospitality. Nestled in the heart of Kandy, we are dedicated to offering our guests an unforgettable dining experience that blends the rich flavors of global with the freshest local ingredients.

**Our Story**

Founded in 2024, ABC was born from a passion for exploring the diverse culinary landscapes of the world. Our mission is to bring the best of Italian, Mediterranean, French, Mexican, Japanese, Indian, American, Thai, Fusion, and Farm-to-Table cuisines to your table, all crafted with the freshest ingredients and a deep love for food.

**Our Values**

- ✓ Diversity: Embracing the culinary traditions of cultures from around the globe.
- ✓ Quality: Using only the finest ingredients, sourced locally whenever possible, to ensure each dish is a true representation of its origins..
- ✓ Innovation: Combining traditional techniques with modern creativity to offer unique and unforgettable dining experiences..

```
<!-- ===== About Section ===== -->
<section id="about" class="about">
  <div class="container" data-aos="fade-up">

    <div class="row">
      <div class="col-lg-6 order-6 order-lg-6" data-aos="zoom-in" data-aos-delay="100">
        <div class="about-img">
          
        </div>
      </div>
      <div class="col-lg-6 pt-4 pt-lg-0 order-2 order-lg-1 content">
        <h3>About Us</h3>
        <p class="fst-italic" style="text-align: justify;">
          Welcome to ABC Restaurant, where culinary passion meets warm hospitality. Nestled in the
        </p>
        | 
        <h3>Our Story</h3>
        <p class="justify-text" >Founded in 2024, ABC was born from a passion for exploring the div
        <h3>Our Values</h3>
        <ul>
          <li><i class="bi bi-check-circle" ></i> Diversity: Embracing the culinary traditions of c
          <li><i class="bi bi-check-circle"></i> Quality: Using only the finest ingredients, source
          <li><i class="bi bi-check-circle"></i> Innovation: Combining traditional techniques with i
        </ul>
      </div>
    </div>
  </div>
```

Figure 14About Us Frontend and Backend

## Offers Section

This section displays special deals and promotions. The admin can easily add new offers, while customers can view the latest discounts and exclusive deals available for online orders.

The screenshot shows a mobile application interface for an offers section. At the top, there's a navigation bar with the word 'OFFERS'. Below it is a large, bold heading 'Check Our Special Offers - Today'. Underneath this, there's a sub-section title 'Ultimate Double Decker Burger' followed by a note '- Exclusively Available for Online Orders -'. A detailed description of the burger follows, mentioning its components like juicy beef patties, melted cheddar cheese, lettuce, tomatoes, onions, and pickles, along with a tangy ketchup and mustard blend. To the right of the text is a promotional image of the burger on a wooden board, with text overlaying it: 'HOT & SPECIAL', 'BURGER', '50% SAVE', and 'RS 350 ONLY'. Below the image is a small 'ORDER NOW' button. At the bottom of the slide, there are four small circular dots indicating the number of slides in the carousel.

```
<section id="specials" class="specials">
<div class="container" data-aos="fade-up">
<div class="section-title">
<h2>Offers</h2>
<p>Check Our Special Offers - Today</p>
</div>

<div class="row justify-content-center" data-aos="fade-up" data-aos-delay="100">
<div class="col-lg-9 mt-4 mt-lg-0">
<div class="swiper-container">
<div class="swiper-wrapper">

<c:forEach var="offer" items="${offers}">
<div class="swiper-slide">
<div class="row">
<div class="col-lg-8 details order-2 order-lg-1">
<h3>${offer.name}</h3>
<p class="fst-italic text-justify">- Exclusively Available for Online Orders - </p>
<p>${offer.description}</p>
</div>
<div class="col-lg-4 text-center order-1 order-lg-2">

</div>
</div>
</div>
</c:forEach>
</div>
</div>
```

## Service Section

In this section, customers can explore the various services offered, including custom parties, private gatherings, birthday celebrations, and food delivery. Administrators can manage and update the service offerings as needed to ensure a smooth experience for all users.



```

<div class="section-title">
    <h2>Services</h2>
    <p>Services We Provide to You</p>
</div>

<div class="events-slider swiper-container" data-aos="fade-up" data-aos-delay="100">
    <div class="swiper-wrapper">

        <c:forEach var="service" items="${services}">
            <div class="swiper-slide">
                <div class="row event-item">
                    <div class="col-lg-6">
                        
                    </div>
                    <div class="col-lg-6 pt-4 pt-lg-0 content">
                        <h3>${service.serviceName}</h3>
                        <p class="fst-italic text-justify">
                            ${service.description}
                        </p>
                    </div>
                </div>
            </div>
        </c:forEach>
    </div>
</div>
```

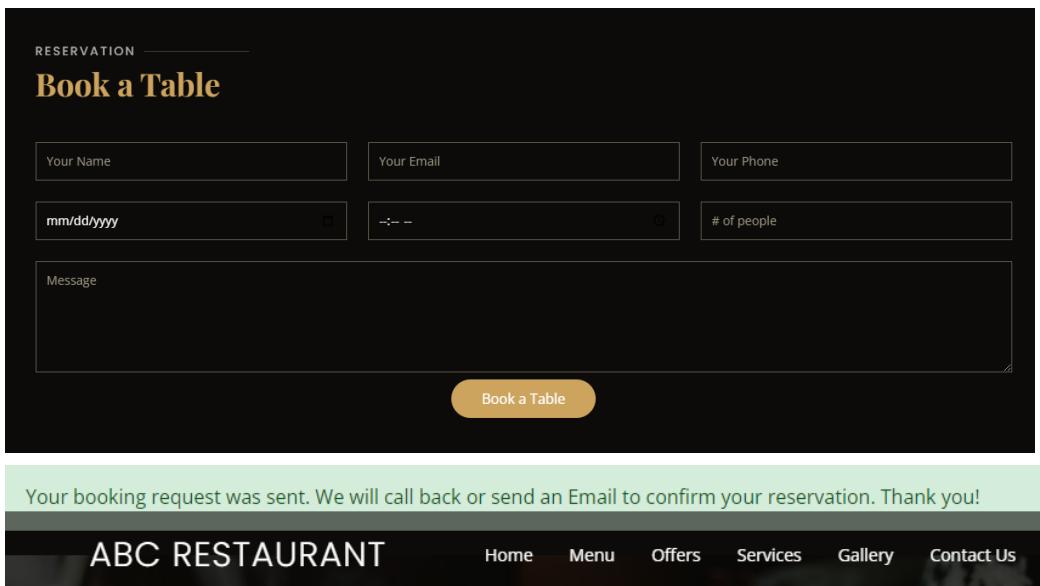
Figure 15 Service Interface and Backend

## Reservation Section

The "Reservation" section allows users to book a table at the restaurant through a simple, user-friendly form. Customers are required to provide essential details, including their name, email, phone number, preferred date, time, number of people, and any special requests they may have. Upon submission, the form is processed by the system.

### User Flow

- **Customer:** Fills out the reservation form with the required details and submits it.
- **Admin:** Reviews the reservation and decides whether to accept or reject it.
- **Notification:** The customer receives an email with the admin's decision, ensuring a seamless communication process



**RESERVATION**

### Book a Table

Your Name

Your Email

Your Phone

mm/dd/yyyy  - - -  # of people

Message

**Book a Table**

Your booking request was sent. We will call back or send an Email to confirm your reservation. Thank you!

ABC RESTAURANT      Home    Menu    Offers    Services    Gallery    Contact Us

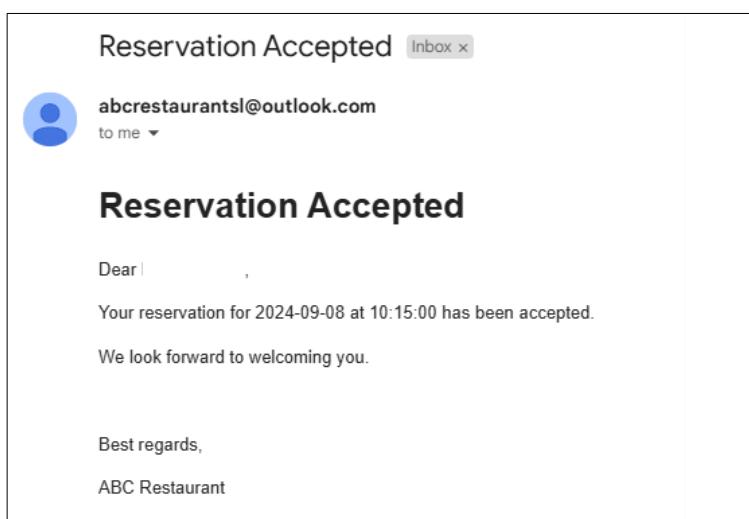


Figure 16 Frontend and Email acceptance of Reservation

```

private void addReservation(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String phone = request.getParameter("phone");
    String date = request.getParameter("date");
    String time = request.getParameter("time");
    int people = Integer.parseInt(request.getParameter("people"));
    String message = request.getParameter("message");
}

public static void sendReservationAcceptedEmail(String recipientEmail, String recipientName, String date, String time)
try {
    Session session = getSession();

    Message message = new MimeMessage(session);
    message.setFrom(new InternetAddress(SENDER_EMAIL));
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(recipientEmail));
    message.setSubject("Reservation Accepted");

    String emailContent = "<h1>Reservation Accepted</h1>
        + "<p>Dear " + recipientName + ",</p>
        + "<p>Your reservation for " + date + " at " + time + " has been accepted.</p>
        + "<p>We look forward to welcoming you.</p>
        + "<br><p>Best regards,</p>
        + "<p>ABC Restaurant</p>";

    message.setContent(emailContent, "text/html");

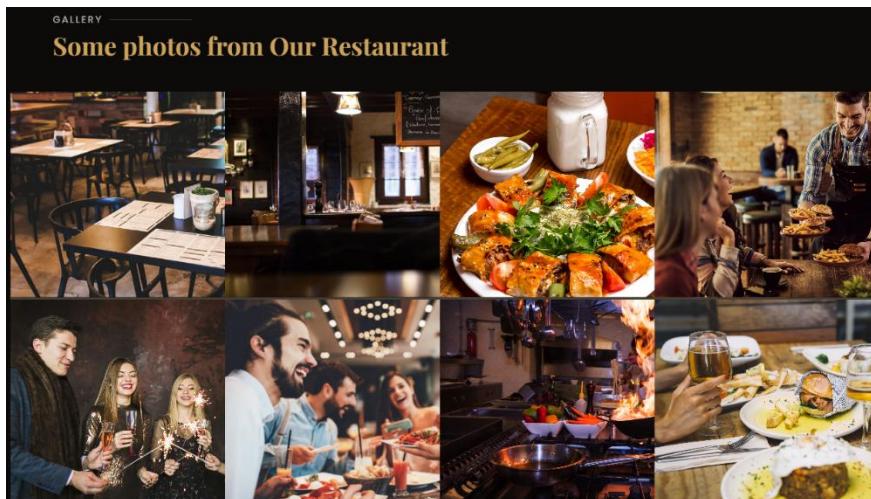
    Transport.send(message);
    System.out.println("Reservation accepted email sent to " + recipientEmail);
} catch (MessagingException e) {
    e.printStackTrace();
}
}

```

Figure 17 Reservation servlet (add) and Email Backend

## Gallery Section

The "Gallery" section offers a visual showcase of the restaurant's atmosphere, dishes, and memorable moments enjoyed by its guests. Customers can browse through these photos to get a feel for the dining experience and the quality of the meals provided. Administrators can add and update photos in the gallery. This allows for continuous updates that showcase the latest dishes, events, and features of the restaurant.



```

<section id="gallery" class="gallery">
    <div class="container" data-aos="fade-up">
        <div class="section-title">
            <h2>Gallery</h2>
            <p>Some photos from Our Restaurant</p>
        </div>
    </div>

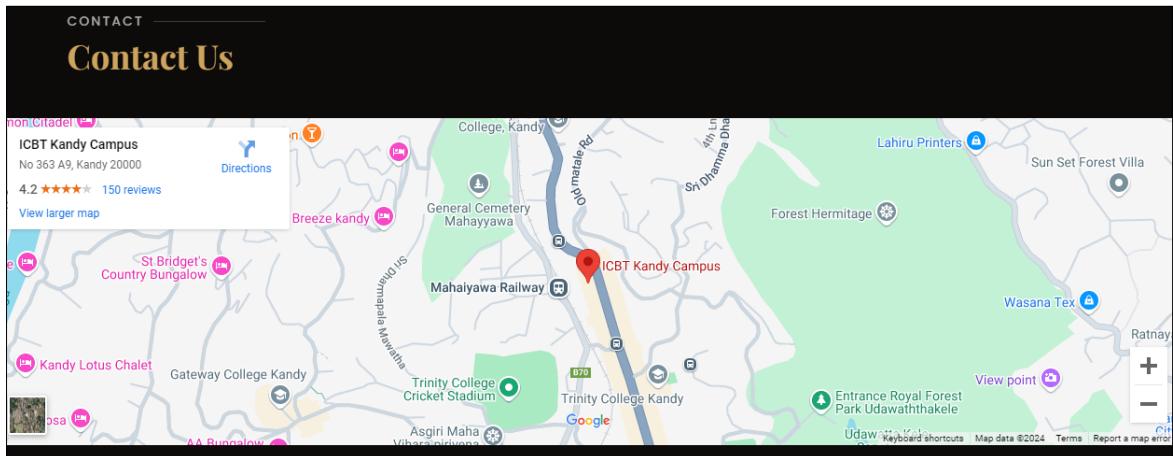
    <div class="container-fluid" data-aos="fade-up" data-aos-delay="100">
        <div class="row g-0">
            <c:forEach var="gallery" items="${galleries}">
                <div class="col-lg-3 col-md-4">
                    <div class="gallery-item">
                        <a href="${gallery.image}" class="gallery-lightbox" data-gall="gallery-item">
                            
                        </a>
                    </div>
                </div>
            </c:forEach>
        </div>
    </div>
</section>

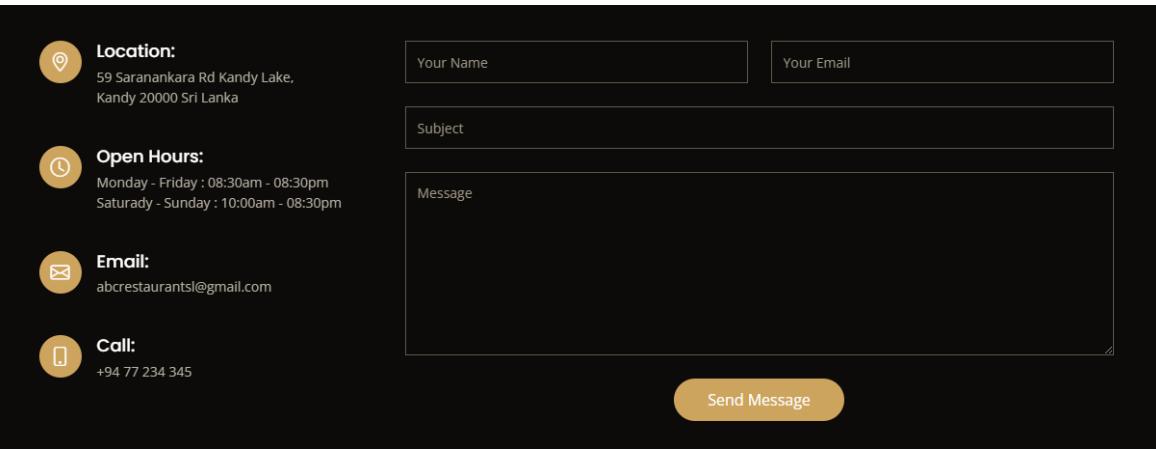
```

Figure 18 Gallery Frontend and Backend

## Queries Section

In this Customer Query section, users can submit their inquiries by filling out their name, email, subject, and message. The system captures these inputs, allowing the admin to receive and respond to customer queries efficiently. Once the form is submitted, the admin can review the message and reply via the customer's provided email. Additionally, the section includes the business location, which can be linked to Google Maps for easy navigation, as well as open hours, email, and phone details for direct communication.





```

Location:  

59 Saranankara Rd Kandy Lake,  

Kandy 20000 Sri Lanka

Open Hours:  

Monday - Friday : 08:30am - 08:30pm  

Saturday - Sunday : 10:00am - 08:30pm

Email:  

abcrestaurantsl@gmail.com

Call:  

+94 77 234 345

```

Your Name

Your Email

Subject

Message

**Send Message**

```

private void addQuery(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String subject = request.getParameter("subject");
    String message = request.getParameter("message");
}

<form action="query?action=add" method="post" role="form">
    <div class="row">
        <div class="col-md-6 form-group">
            <input type="text" name="name" class="form-control" id="name" placeholder="Your Name" required>
        </div>
        <div class="col-md-6 form-group mt-3 mt-md-0">
            <input type="email" class="form-control" name="email" id="email" placeholder="Your Email" required>
        </div>
    </div>
    <div class="form-group mt-3">
        <input type="text" class="form-control" name="subject" id="subject" placeholder="Subject" required>
    </div>
    <div class="form-group mt-3">
        <textarea class="form-control" name="message" rows="8" placeholder="Message" required></textarea>
    </div>
    <div class="my-3">
        <div class="Loading">Loading</div>
        <div class="error-message"></div>
    </div>
    <div class="text-center"><button type="submit">Send Message</button></div>
</form>

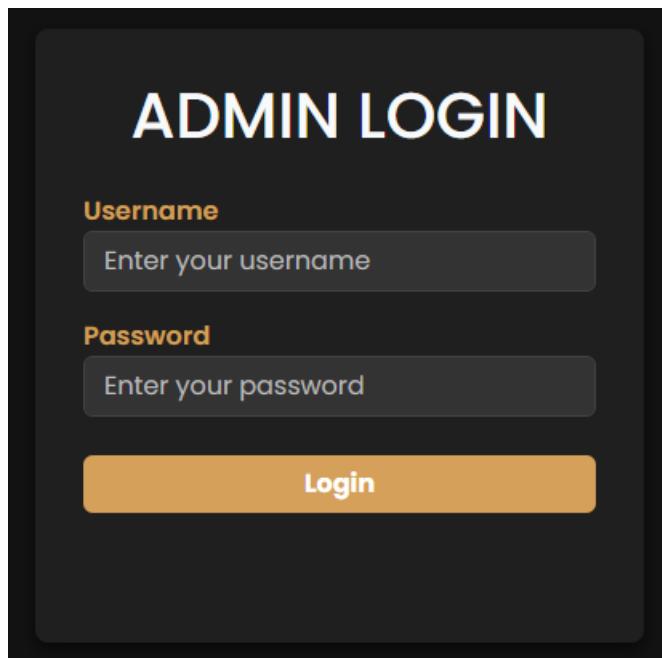
```

Figure 19 Query Servlet and frontend and backend

- Admin Area

### Admin Login

The admin login interface for the ABC Restaurant Management System allows admin to securely access the dashboard. After entering valid login credentials, the system verifies the admin's details and directs to admin dashboard.



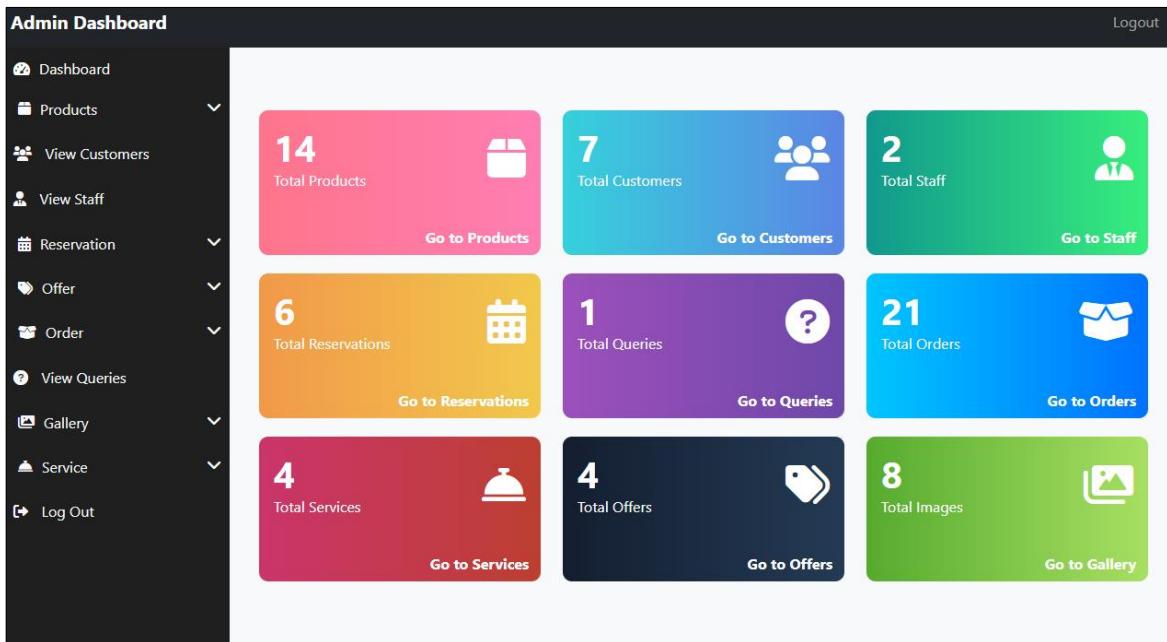
```
private void loginAdmin(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String username = request.getParameter("username");
    String password = request.getParameter("password");

    AdminService adminService = new AdminService();
    Admin admin = adminService.loginAdmin(username, password);

    if (admin != null) {
        request.getSession().setAttribute("admin", admin);
        request.getSession().setAttribute("role", "admin");
        response.sendRedirect("adminDashboard.jsp");
    } else {
        request.setAttribute("error", "Invalid username or password");
        request.getRequestDispatcher("adminLogin.jsp").forward(request, response);
    }
}
```

Figure 20 admin Login interface and admin servlet

## Admin Dashboard Interface



The Admin Dashboard provides a centralized interface for the administrator to manage all key aspects of the business. The dashboard displays important statistics like the number of products, customers, staff, reservations, queries, orders, services, offers, and gallery images. Through the dashboard, the admin can:

- Manage Products: Add, edit, or delete products.
- Manage Customers: View and manage customer information.
- Manage Staff: Control staff details and assignments.
- Manage Reservations: Oversee customer bookings.
- Manage Offers: Create and adjust promotional offers.
- Manage Orders: View customer orders.
- Manage Queries: Respond to customer inquiries.
- Manage Services: Control services.
- Manage Gallery: Add or remove images in the gallery section.

This system simplifies business management by providing easy access and control over different operations.

```

package com.abc.controller;

import com.abc.dao.ProductDAO;

@WebServlet("/admin")
public class AdminController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private ProductDAO productDAO;
    private CustomerDAO customerDAO;
    private StaffDAO staffDAO;
    private ReservationDAO reservationDAO;
    private QueryDAO queryDAO;
    private OrderDAO orderDAO;
    private ServiceDAO serviceDAO;
    private OfferDAO offerDAO;
    private GalleryDAO galleryDAO;

    @Override
    public void init() {
        productDAO = new ProductDAO();
        customerDAO = new CustomerDAO();
        staffDAO = new StaffDAO();
        reservationDAO = new ReservationDAO();
        queryDAO = new QueryDAO();
        orderDAO = new OrderDAO();
        serviceDAO = new ServiceDAO();
        offerDAO = new OfferDAO();
        galleryDAO = new GalleryDAO();
    }
}

```

Figure 21 Admin Controller Servlet

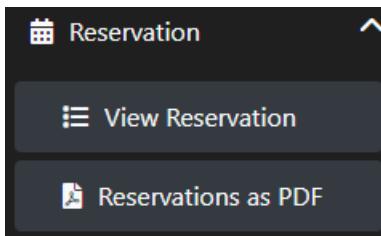
Users can locate certain products fast by using the Category filter and the "Search by Name" input area.

Product List						
Search for products...						
ID	Name	Price	Description	Image	Category	Actions
1	Penne Pasta	700	Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.		Breakfast	<button>Edit</button> <button>Delete</button>
2	Fried Rice Special	800	Experience the rich flavors of our signature fried rice, wok-charred to perfection with fragrant jasmine rice, soy sauce, garlic, and sesame oil.		Breakfast	<button>Edit</button> <button>Delete</button>
3	Fried spring rolls	350	Embark on a delightful journey with our Crispy Harmony Spring Rolls. Impeccably hand-rolled, these golden wonders unveil a symphony of textures and flavors.		Breakfast	<button>Edit</button> <button>Delete</button>
16	Loaded Tacos	200	Experience the vibrant spirit of street food with our mouthwatering tacos. Elevate your taste buds with a burst of zesty salsa and the perfect dash of lime.		Breakfast	<button>Edit</button> <button>Delete</button>
17	String hoppers	250	Transport your taste buds to the vibrant streets of Sri Lanka with our exquisite String Hoppers. Delicate nests of steamed rice flour vermicelli, artfully woven into a tapestry, serve as the perfect canvas for an array of flavorful curries.		Breakfast	<button>Edit</button> <button>Delete</button>

Figure 22 Item List with the search bar

The Admin Dashboard not only provides control over various aspects of the business but also allows the admin to generate and download reports, such as PDFs. These reports include detailed insights, making it easier for the admin to manage and track important data.

- **Reservation Reports:** Admin can download reports showing all reservations, categorized by their status accepted, rejected, and pending making it easy to keep track of booking requests and their outcomes.
- **Sales Reports:** The dashboard also allows the admin to generate daily sales reports, showing the income from sold products. This feature helps the admin monitor daily sales performance and overall revenue trends.



## Reservations List

### Accepted Reservations

ID	Name	Email	Phone	Date & Time	Status
55				2024-10-02 23:33:00	Accepted
57			0702515900	2024-09-08 10:15:00	Accepted
52	Kasun	Kasun@gmail.com	0702415678	2024-09-04 15:15:00	Accepted

### Rejected Reservations

ID	Name	Email	Phone	Date & Time	Status
56	Indian			2024-09-30 21:23:00	Rejected
53	Sadun	sadun@gmail.com	0892237383	2024-09-13 06:20:00	Rejected

### Pending Reservations

ID	Name	Email	Phone	Date & Time	Status
54				2024-09-30 17:14:00	Pending

Figure 23 Reservation report pdf

```

package com.abc.util;

import com.itextpdf.io.source.ByteArrayOutputStream;
import com.itextpdf.io.source.OutputStream;

public class PDFUtil {

    public static ByteArrayOutputStream generateReservationPDF(List<Reservation> reservations) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        try {
            PdfWriter writer = new PdfWriter(baos);
            PdfDocument pdfDoc = new PdfDocument(writer);
            Document document = new Document(pdfDoc);

            Collections.sort(reservations, new Comparator<Reservation>() {
                @Override
                public int compare(Reservation r1, Reservation r2) {
                    String dateTime1 = r1.getDate() + " " + r1.getTime();
                    String dateTime2 = r2.getDate() + " " + r2.getTime();
                    return dateTime2.compareTo(dateTime1);
                }
            });
            document.add(new Paragraph("Reservations List").setBold().setFontSize(18));

            List<Reservation> acceptedReservations = reservations.stream()
                .filter(r -> "Accepted".equalsIgnoreCase(r.getStatus()))
                .collect(Collectors.toList());
        }
    }

    document.add(new Paragraph("Reservations List").setBold().setFontSize(18));

    List<Reservation> acceptedReservations = reservations.stream()
        .filter(r -> "Accepted".equalsIgnoreCase(r.getStatus()))
        .collect(Collectors.toList());

    List<Reservation> rejectedReservations = reservations.stream()
        .filter(r -> "Rejected".equalsIgnoreCase(r.getStatus()))
        .collect(Collectors.toList());

    List<Reservation> pendingReservations = reservations.stream()
        .filter(r -> "Pending".equalsIgnoreCase(r.getStatus()))
        .collect(Collectors.toList());

    if (!acceptedReservations.isEmpty()) {
        document.add(new Paragraph("Accepted Reservations").setBold().setFontSize(14));
        document.add(createTable(acceptedReservations, new DeviceRgb(0, 128, 0)));
    }

    if (!rejectedReservations.isEmpty()) {
        document.add(new Paragraph("Rejected Reservations").setBold().setFontSize(14));
        document.add(createTable(rejectedReservations, new DeviceRgb(255, 0, 0)));
    }

    if (!pendingReservations.isEmpty()) {
        document.add(new Paragraph("Pending Reservations").setBold().setFontSize(14));
        document.add(createTable(pendingReservations, new DeviceRgb(255, 165, 0)));
    }
}

```

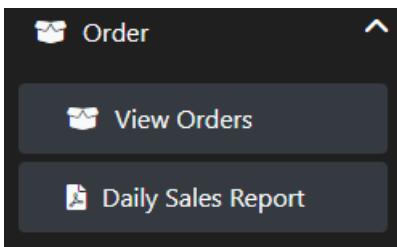
Figure 24 Reservation Generate backend code

## Reservation Management

ID	Name	Email	Phone	Date	Time	No. of People	Message	Status	Action
57				2024-09-08	10:15:00	3	I would like to reserve a table at ABC Restaurant	Accepted	<button>Accept</button> <button>Reject</button> <button>Delete</button>

## Sales Report

Date	Total Selling Products	Total Income
2024-09-03	1	Rs 650.00
2024-09-04	17	Rs 13025.00
2024-09-05	1	Rs 1500.00
2024-09-08	2	Rs 2900.00



```
document.add(new Paragraph("Sales Report").setBold().setFontSize(18));

Table table = new Table(new float[]{2, 2, 2});
table.setWidth(UnitValue.createPercentValue(100));

table.addHeaderCell(new Cell().add(new Paragraph("Date"))
.setBackgroundColor(new DeviceRgb(0, 128, 0))
.setFontColor(new DeviceRgb(255, 255, 255)));

table.addHeaderCell(new Cell().add(new Paragraph("Total Selling Products"))
.setBackgroundColor(new DeviceRgb(0, 128, 0))
.setFontColor(new DeviceRgb(255, 255, 255)));

table.addHeaderCell(new Cell().add(new Paragraph("Total Income"))
.setBackgroundColor(new DeviceRgb(0, 128, 0))
.setFontColor(new DeviceRgb(255, 255, 255)));

for (Map.Entry<String, OrderReport> entry : report.entrySet()) {
    OrderReport orderReport = entry.getValue();
    table.addCell(orderReport.getDate());
    table.addCell(String.valueOf(orderReport.getTotalProducts()));
    table.addCell(String.format("Rs %.2f", orderReport.getTotalIncome()));
}

document.add(table);
document.close();
```

Figure 25 View Sales report pdf download button and backend code

## Query Management

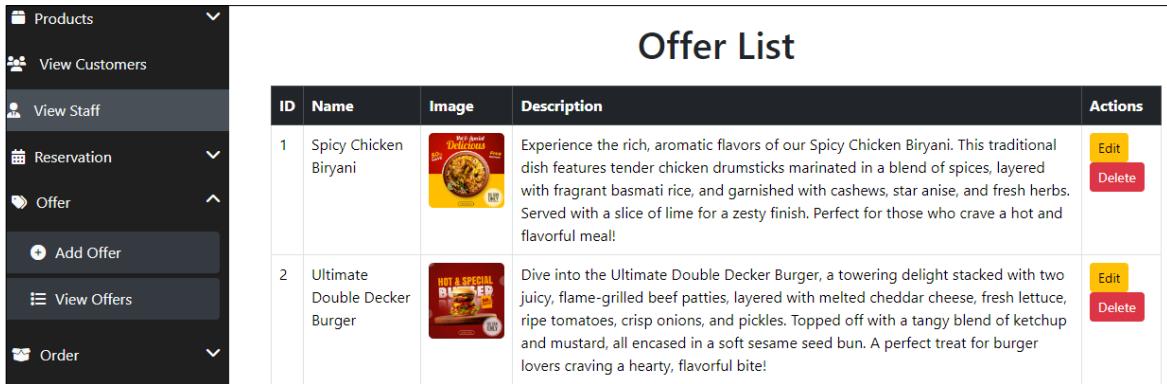
For each query, the admin has the option to Reply or Delete. Once the admin replies to a query, the state is updated to "Replied." If a query has not been responded to yet, it will show as "Pending."

- Admin Dashboard**
- Dashboard
- Products
- View Customers
- View Staff
- Reservation
- Offer
- Order
- View Queries

### Query List

ID	Name	Email	Subject	Message	State	Action
1			About Your Service	Nice Service. Friendly Customer Service	Replied	<button>Reply</button> <button>Delete</button>
2	Pasun Herath	pasanHera@gmail.com	About Your Meals	The Food was absolutely Delicious. Thank you for wonderful dining experience.	Pending	<button>Reply</button> <button>Delete</button>

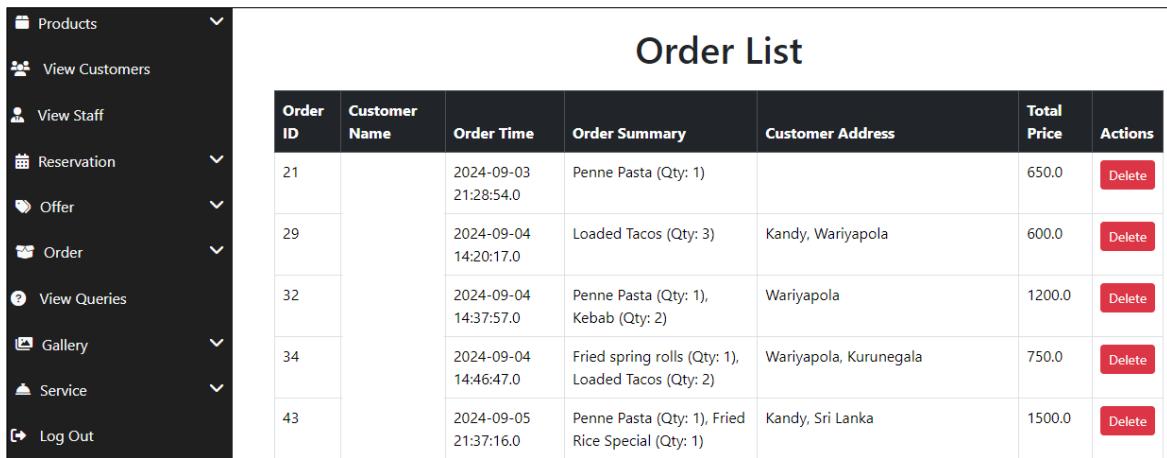
## Additional admin functions



The screenshot shows a sidebar menu on the left with options like Products, View Customers, View Staff, Reservation, Offer, Add Offer, View Offers, Order, View Queries, Gallery, Service, and Log Out. The 'Offer' option is selected. The main content area is titled 'Offer List' and displays two rows of offers:

ID	Name	Image	Description	Actions
1	Spicy Chicken Biryani		Experience the rich, aromatic flavors of our Spicy Chicken Biryani. This traditional dish features tender chicken drumsticks marinated in a blend of spices, layered with fragrant basmati rice, and garnished with cashews, star anise, and fresh herbs. Served with a slice of lime for a zesty finish. Perfect for those who crave a hot and flavorful meal!	<button>Edit</button> <button>Delete</button>
2	Ultimate Double Decker Burger		Dive into the Ultimate Double Decker Burger, a towering delight stacked with two juicy, flame-grilled beef patties, layered with melted cheddar cheese, fresh lettuce, ripe tomatoes, crisp onions, and pickles. Topped off with a tangy blend of ketchup and mustard, all encased in a soft sesame seed bun. A perfect treat for burger lovers craving a hearty, flavorful bite!	<button>Edit</button> <button>Delete</button>

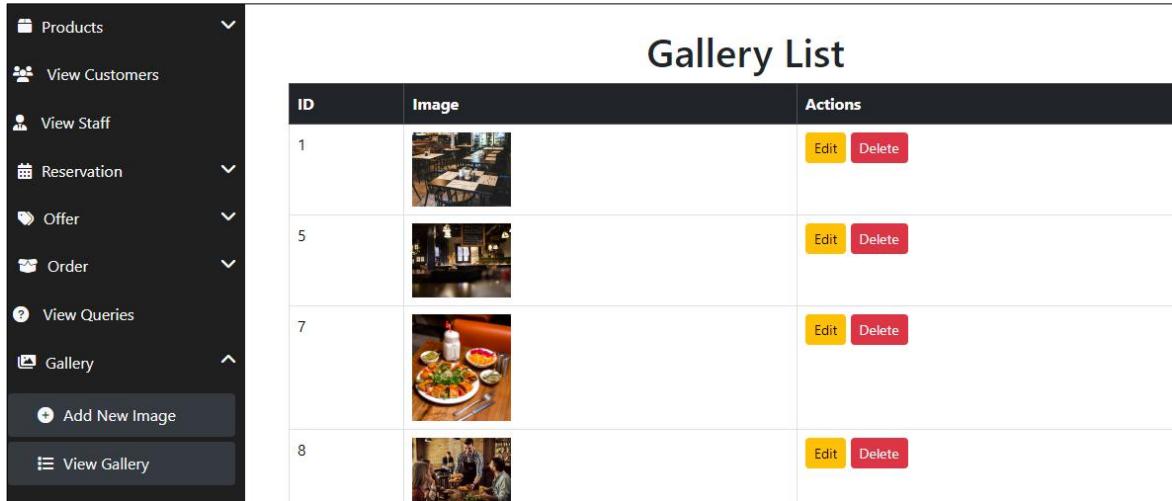
Figure 26 Offer Management



The sidebar menu on the left includes Add Offer, View Offers, Order, View Queries, Gallery, Service, and Log Out. The 'Order' option is selected. The main content area is titled 'Order List' and displays five rows of order details:

Order ID	Customer Name	Order Time	Order Summary	Customer Address	Total Price	Actions
21		2024-09-03 21:28:54.0	Penne Pasta (Qty: 1)		650.0	<button>Delete</button>
29		2024-09-04 14:20:17.0	Loaded Tacos (Qty: 3)	Kandy, Wariyapola	600.0	<button>Delete</button>
32		2024-09-04 14:37:57.0	Penne Pasta (Qty: 1), Kebab (Qty: 2)	Wariyapola	1200.0	<button>Delete</button>
34		2024-09-04 14:46:47.0	Fried spring rolls (Qty: 1), Loaded Tacos (Qty: 2)	Wariyapola, Kurunegala	750.0	<button>Delete</button>
43		2024-09-05 21:37:16.0	Penne Pasta (Qty: 1), Fried Rice Special (Qty: 1)	Kandy, Sri Lanka	1500.0	<button>Delete</button>

Figure 27 Order Management



The sidebar menu on the left includes Add New Image and View Gallery. The 'View Gallery' option is selected. The main content area is titled 'Gallery List' and displays four rows of images:

ID	Image	Actions
1		<button>Edit</button> <button>Delete</button>
5		<button>Edit</button> <button>Delete</button>
7		<button>Edit</button> <button>Delete</button>
8		<button>Edit</button> <button>Delete</button>

Figure 28 Gallery Management

- **Staff Dashboard**

- The Staff Dashboard provides an interface for staff members to manage key areas of the business. Staff can view and access sections such as products, customers, reservations, and customer queries.
- While staff members can view and manage certain aspects, they typically have limited permissions compared to the admin dashboard, focusing on tasks relevant to their role. This ensures streamlined business operations with clearly defined responsibilities.
- Using the same dashboard for both the admin and staff is a practical approach to maintain a unified interface. However, the dashboard will have role-based access control, ensuring that while both admin and staff have access to the same visual interface, the functionality and permissions differ based on their roles.
- Staff can register via a dedicated registration form, where they can provide personal details, contact information, and create login credentials.

The image shows two dark-themed registration forms side-by-side.

**STAFF LOGIN:** This form has fields for 'Email' (placeholder: Enter your email) and 'Password' (placeholder: Enter your password). It features a large orange 'Login' button at the bottom. Below the button, a link says 'Not registered? Sign Up'.

**REGISTER STAFF:** This form has fields for 'Name' (placeholder: Enter your name), 'Email' (placeholder: Enter your email), 'Phone Number' (placeholder: Enter your phone number), and 'Password' (placeholder: Enter your password). It includes a 'Role' dropdown set to 'Staff'. A large orange 'Register' button is at the bottom. Below the button, a link says 'Already a Member? Login here'.

The image shows a screenshot of the Staff Dashboard. The left sidebar contains a navigation menu with options: 'Dashboard', 'Products', 'View Customers', 'Reservation', and 'Log Out'. The main area displays four summary cards:

- 14 Total Products** (pink card, 'Go to Products')
- 6 Total Customers** (blue card, 'Go to Customers')
- 5 Total Reservations** (orange card, 'Go to Reservations')
- 2 Total Queries** (purple card, 'Go to Queries')

Figure 29 Staff Loggin, Register and Dashboard

**Task C****ABC Restaurant Management System Testing plans**

A test plan can be created for the developed ABC Restaurant as follows.

<b>WEB APP ID</b>	Web01
<b>Version</b>	0.01
<b>Tester</b>	Application developer
<b>Designed Date</b>	2024-08-15
<b>Conducted Date</b>	2024-09-10
<b>Test ID</b>	WebT01
<b>Main Objective</b>	To verify whether the developed Web App of ABC Restaurant is functioning properly
<b>Test Cases - Customer</b>	Test Case 01: Sign up (register) as a new user. Test Case 02: Incorrect username or password. Test Case 03: Customer books a table. Test Case 04: Customer sends a query. Test Case 05: Customer adds food to cart. Test Case 06: Customer makes payment. Test Case 07: Customer orders a meal online.
<b>Test Cases - Admin</b>	Test Case 01: Admin searches for products. Test Case 02: Admin adds products. Test Case 03: Admin updates products. Test Case 04: Admin deletes products. Test Case 05: Admin accepts customer reservations. Test Case 06: Admin replies to customer queries.
<b>Test Cases - Staff</b>	Test Case 01: Sign up (register) as a new Staff Member. Test Case 02: Login Staff Member.
<b>Expected Result</b>	Expected the website to function efficiently and error-free environment
<b>Actual Result</b>	The complete website functioned well without any errors.

## Customer Test Cases

<b>No of the Test Case</b>	Test Case 01		
<b>Requirement</b>	Sign up (register) as a new user.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	Name – K Email – <a href="mailto:k@abcrestaurantsl.outlook.com">k@abcrestaurantsl.outlook.com</a> Phone No – 0701234567 Password –	Send a welcome email to the customer and display their name on the navigation bar.	Pass

The figure consists of three screenshots. The top-left screenshot shows a 'REGISTER' form with fields for Name, Email, Phone Number, and Password. The top-right screenshot shows an email from 'abcrestaurantsl@outlook.com' with the subject 'Welcome to ABC Restaurant!' containing a welcome message and signature. The bottom screenshot shows a navigation bar with 'WELCOME, [REDACTED]' and links for Home, Menu, Offers, and Services.

Figure 30 Register Proof

No of the Test Case	Test Case 02		
Requirement	Incorrect username or password.		
Tester	Application developer		
Input	Data Samples	Expected Result	Received Result
Valid (Positive)	Email – Passwor	Invalid email or password	Pass

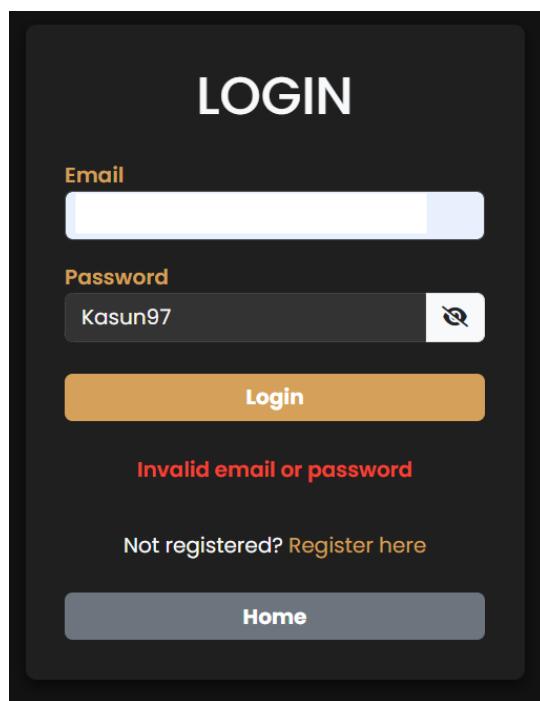
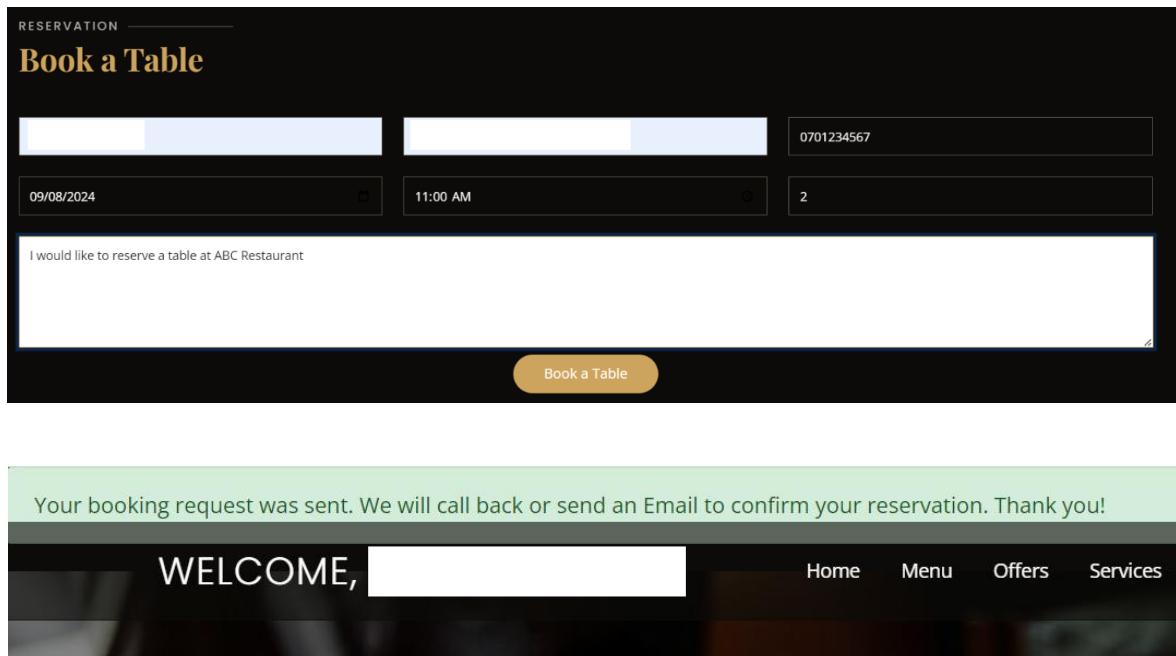


Figure 31 Invalid email or password proof

<b>No of the Test Case</b>	Test Case 03		
<b>Requirement</b>	Customer books a table.		
<b>Tester</b>	Application developer		
Input	Data Samples	Expected Result	Received Result
Valid (Positive)	Your name – <b>Kiran</b> email – <b>Kiran@abc.com</b> Phone – 0701234567 Date – 09/08/2024 Time – 11:00 A.M No of people - 2 Message - I would like to reserve a table at ABC Restaurant	Your booking request was sent. We will call back or send an Email to confirm your reservation. Thank you!	Pass



The screenshot shows a user interface for booking a table. At the top, it says "RESERVATION" and "Book a Table". Below this is a form with fields for name (partially visible), email (0701234567), phone number (0701234567), date (09/08/2024), time (11:00 AM), and number of people (2). There is also a message input field containing "I would like to reserve a table at ABC Restaurant". A "Book a Table" button is at the bottom of the form. Below the form, a green banner displays the message "Your booking request was sent. We will call back or send an Email to confirm your reservation. Thank you!". At the very bottom, there is a dark footer bar with the text "WELCOME, [REDACTED]" and navigation links for "Home", "Menu", "Offers", and "Services".

Figure 32 Reservation Success Proof

<b>No of the Test Case</b>	Test Case 04		
<b>Requirement</b>	Test Case 04: Customer sends a query.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	Name – Email – Subject - About Your Meals Message - The Food was absolutely Delicious. Thank you for wonderful dining experience.	Your query has been successfully submitted!	Pass

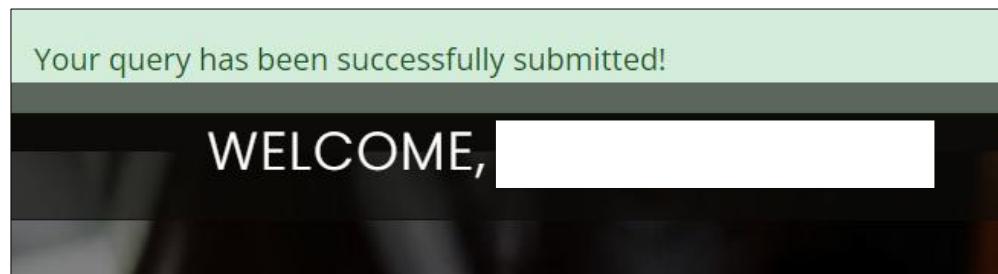
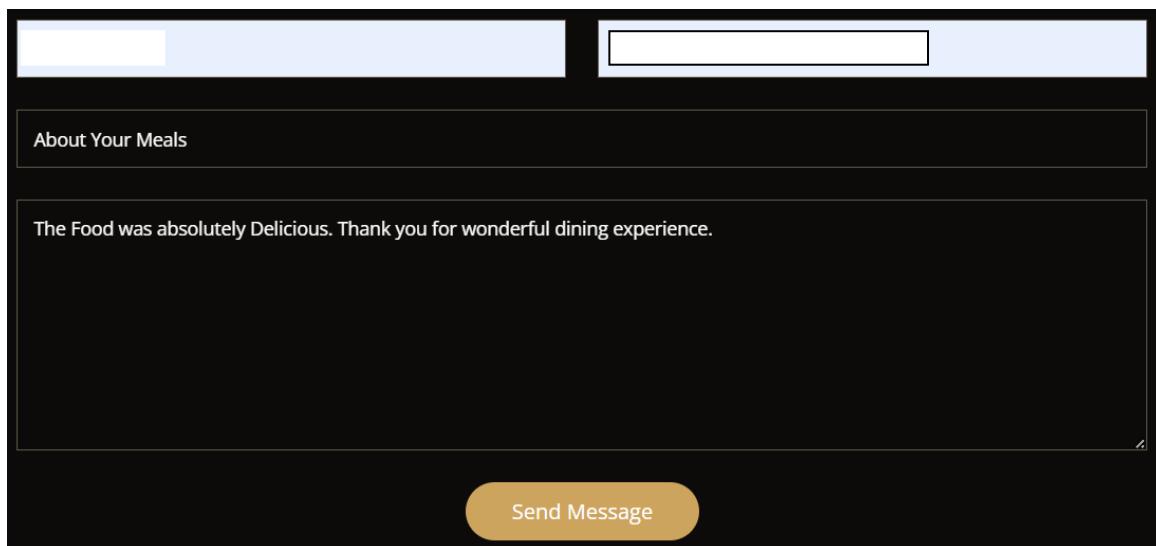


Figure 33 Query Success Proof

<b>No of the Test Case</b>	Test Case 05		
<b>Requirement</b>	Test Case 05: Customer adds food to cart.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	Add “Penne Pasta” and “Fried rice special” to the cart	Display added foods in the cart and total amount of them	Pass

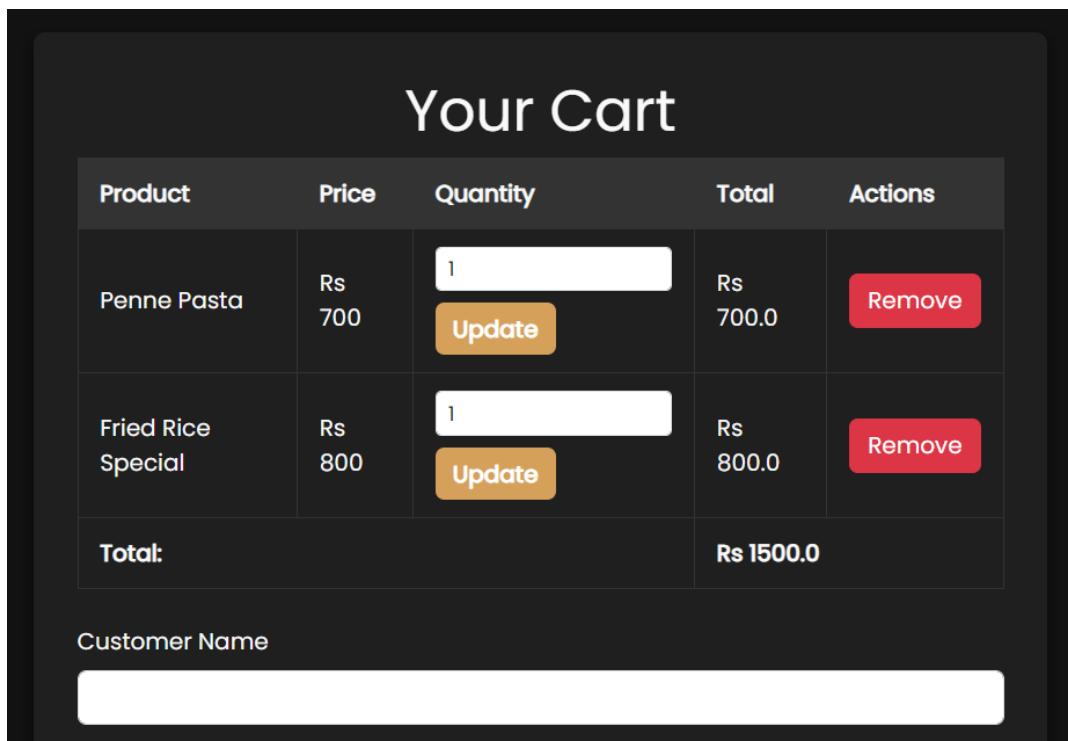


Figure 34 Add items to cart proof

<b>No of the Test Case</b>	Test Case 06		
<b>Requirement</b>	Test Case 06: Customer makes payment.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	Select Card – Visa Card number – 4444888877779999 Expiry Month – 09 Expiry Year – 2024 CVV - 345	Payment successfully	Pass

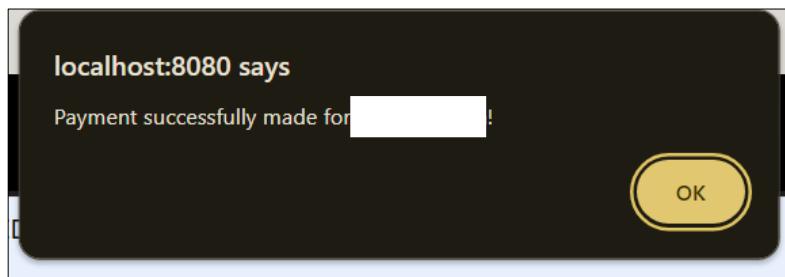


Figure 35 Makes payment successfully proof

No of the Test Case	Test Case 07		
Requirement	Test Case 07: Customer orders a meal online.		
Tester	Application developer		
Input	Data Samples	Expected Result	Received Result
Valid (Positive)	Add Items to the cart and enter card details and proceed to checkout	Show order conformation message	Pass

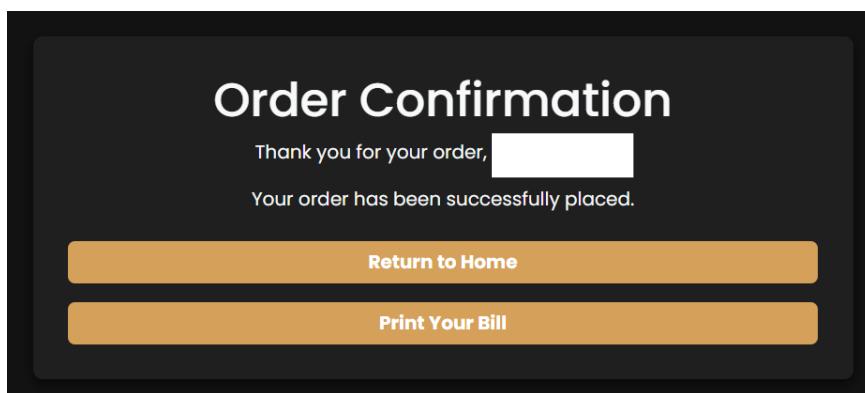


Figure 36 Order Conformation Message

### Admin Test cases

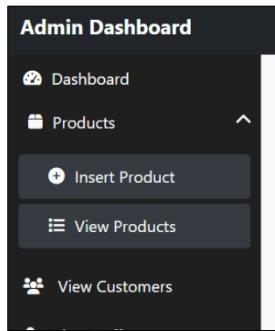
No of the Test Case	Test Case 01		
Requirement	Test Case 01: Admin searches for products		
Tester	Application developer		
Input	Data Samples	Expected Result	Received Result
Valid (Positive)	Search “Penne Pasta”	Searching only “Penne Pasta” that among all the items	Pass

Product List						
Search for products...						
ID	Name	Price	Description	Image	Category	Actions
1	Penne Pasta	700	Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.		Breakfast	<button>Edit</button> <button>Delete</button>
2	Fried Rice Special	800	Experience the rich flavors of our signature fried rice, wok-charred to perfection with fragrant jasmine rice, soy sauce, garlic, and sesame oil.		Breakfast	<button>Edit</button> <button>Delete</button>
3	Fried spring rolls	350	Embark on a delightful journey with our Crispy Harmony Spring Rolls. Impeccably hand-rolled, these golden wonders unveil a symphony of textures and flavors.		Breakfast	<button>Edit</button> <button>Delete</button>
16	Loaded Tacos	200	Experience the vibrant spirit of street food with our mouthwatering tacos. Elevate your taste buds with a burst of zesty salsa and the perfect dash of lime.		Breakfast	<button>Edit</button> <button>Delete</button>
17	String hoppers	250	Transport your taste buds to the vibrant streets of Sri Lanka with our exquisite String Hoppers. Delicate nests of steamed rice flour vermicelli, artfully woven into a tapestry, serve as the perfect canvas for an array of flavorful curries.		Breakfast	<button>Edit</button> <button>Delete</button>

Product List						
Penne Pasta						
ID	Name	Price	Description	Image	Category	Actions
1	Penne Pasta	700	Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.		Breakfast	<button>Edit</button> <button>Delete</button>

Figure 37 Admin Search Products

<b>No of the Test Case</b>	Test Case 02		
<b>Requirement</b>	Test Case 02: Admin adds products.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	<ul style="list-style-type: none"> <li>• Name : Iced coffee</li> <li>• Description : Refreshing and bold, our iced coffee is brewed to perfection and served chilled over ice. Enjoy the smooth, rich flavor of premium coffee with a cool twist.</li> <li>• Price : 250</li> <li>• Category : Beverages</li> <li>• Image : iced coffee.jpeg</li> </ul>	Product Added Successfully	Pass



### Add Product

Name:

Description:

Price:

Category:

Image:

**Add Product**

Product Added Successfully X

Figure 38 Product Added Successfully Proof

No of the Test Case	Test Case 03		
Requirement	Test Case 03: Admin updates products.		
Tester	Application developer		
Input	Data Samples	Expected Result	Received Result
Valid (Positive)	<ul style="list-style-type: none"> <li>Name : Penne Pasta</li> <li>Description : Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.</li> <li>Price : 700</li> <li>Category : Breakfast</li> <li>Image : Pasta.jpg</li> </ul>	<ul style="list-style-type: none"> <li>Name : Penne Pasta</li> <li>Description : Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.</li> <li>Price : 650</li> <li>Category : Lunch</li> <li>Image : Pasta.jpg</li> </ul>	Pass

ID	Name	Price	Description	Image	Category	Actions
1	Penne Pasta	700	Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.		Breakfast	<button>Edit</button> <button>Delete</button>

Product Updated Successfully 

ID	Name	Price	Description	Image	Category	Actions
1	Penne Pasta	650	Elevate your dining experience with our artisan-crafted pasta dishes. Immerse your senses in a symphony of flavors as al dente pasta mingles with rich, house-made sauces.		Lunch	<button>Edit</button> <button>Delete</button>

Figure 39 Product Update Proof

No of the Test Case	Test Case 04			
Requirement	Test Case 04: Admin deletes products.			
Tester	Application developer			
Input	Data Samples		Expected Result	Received Result
Valid (Positive)	Delete Chocolate Donuts		Product Delete Successfully	Pass

**Admin Dashboard**

Admin Dashboard							Logout
23	Noodles	300	Embark on a journey of taste and texture with our exquisite noodle dishes. Silken strands of perfectly cooked noodles, expertly crafted to absorb the essence of our signature sauces.		Dinner	<button>Edit</button>	<button>Delete</button>
24	Chicken Kottu	700	Immerse yourself in the rhythm of our Chicken Kottu, a beloved street food sensation elevated to gourmet perfection. Hear the sizzle as our skilled chefs chop and stir-fry tender chicken with an orchestra of spices.		Dinner	<button>Edit</button>	<button>Delete</button>
25	Rice and curry	400	Embark on a culinary journey to the teardrop isle with our Sri Lankan Rice and Curry. Fragrant basmati rice, perfectly cooked and paired with an array of vibrant curries that dance with the flavors of coconut, spices, and aromatic herbs.		Dinner	<button>Edit</button>	<button>Delete</button>
27	caramel pudding	250	A smooth and creamy caramel pudding, made with a rich blend of caramelized sugar and silky custard, offering a perfect balance of sweetness in every spoonful.		Dessert	<button>Edit</button>	<button>Delete</button>
28	Iced coffee	250	Refreshing and bold, our iced coffee is brewed to perfection and served chilled over ice. Enjoy the smooth, rich flavor of premium coffee with a cool twist.		Beverages	<button>Edit</button>	<button>Delete</button>
29	Ice Cream	400	Indulge in our creamy, handcrafted ice cream, available in a variety of classic and seasonal flavors.		Dessert	<button>Edit</button>	<button>Delete</button>
30	Chocolate donuts	200	Satisfy your sweet tooth with our rich and fluffy chocolate donuts. Coated in smooth, decadent chocolate glaze, these irresistible treats are the perfect indulgence any time of day.		Dessert	<button>Edit</button>	<button>Delete</button>

**Admin Dashboard**

Admin Dashboard							Logout
	Burger		Sink your teeth into juicy, handcrafted patties, grilled to perfection and nestled within a fresh, soft bun		Product Deleted Successfully	X	
23	Noodles	300	Embark on a journey of taste and texture with our exquisite noodle dishes. Silken strands of perfectly cooked noodles, expertly crafted to absorb the essence of our signature sauces.		Dinner	<button>Edit</button>	<button>Delete</button>
24	Chicken Kottu	700	Immerse yourself in the rhythm of our Chicken Kottu, a beloved street food sensation elevated to gourmet perfection. Hear the sizzle as our skilled chefs chop and stir-fry tender chicken with an orchestra of spices.		Dinner	<button>Edit</button>	<button>Delete</button>
25	Rice and curry	400	Embark on a culinary journey to the teardrop isle with our Sri Lankan Rice and Curry. Fragrant basmati rice, perfectly cooked and paired with an array of vibrant curries that dance with the flavors of coconut, spices, and aromatic herbs.		Dinner	<button>Edit</button>	<button>Delete</button>
27	caramel pudding	250	A smooth and creamy caramel pudding, made with a rich blend of caramelized sugar and silky custard, offering a perfect balance of sweetness in every spoonful.		Dessert	<button>Edit</button>	<button>Delete</button>
28	Iced coffee	250	Refreshing and bold, our iced coffee is brewed to perfection and served chilled over ice. Enjoy the smooth, rich flavor of premium coffee with a cool twist.		Beverages	<button>Edit</button>	<button>Delete</button>
29	Ice Cream	400	Indulge in our creamy, handcrafted ice cream, available in a variety of classic and seasonal flavors.		Dessert	<button>Edit</button>	<button>Delete</button>

Figure 40 Product Delete Proof

<b>No of the Test Case</b>	Test Case 05		
<b>Requirement</b>	Test Case 05: Admin accepts customer reservations.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	<ul style="list-style-type: none"> <li>Name – <input type="text"/></li> <li>Email – <input type="text"/> <u>com</u></li> <li>Phone – 0701234567</li> <li>Date - 2024-09-08</li> <li>Time – 11:00</li> <li>No of People – 2</li> <li>Message – I would like to reserve a table at ABC Restaurant</li> <li>Status – Pending</li> </ul>	<ul style="list-style-type: none"> <li>Update States “Pending” to “Accepted”</li> <li>Receive an email to Customer</li> <li>Show Accepted Reservation successful message</li> </ul>	Pass

Reservation List									
ID	Name	Email	Phone	Date	Time	No. of People	Message	Status	Action
59			0701234567	2024-09-08	11:00:00	2	I would like to reserve a table at ABC Restaurant	Accepted	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <span>Accept</span> </div> <div style="text-align: center;"> <span>Reject</span> </div> <div style="text-align: center;"> <span>Delete</span> </div> </div>

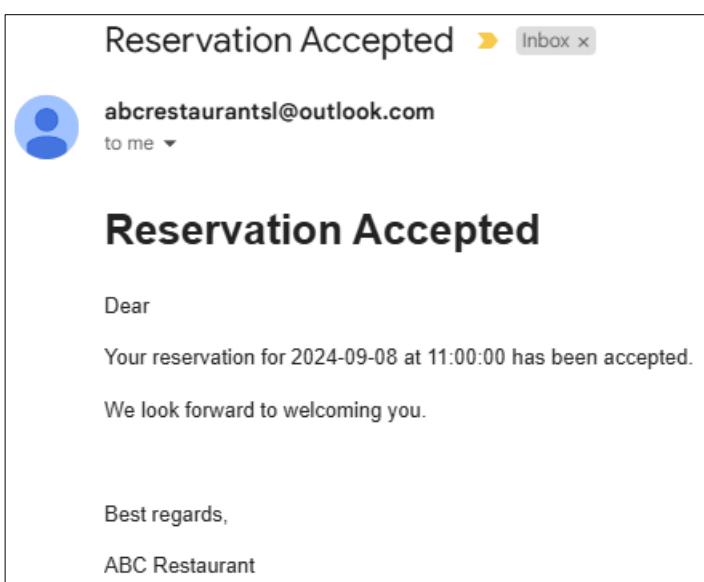
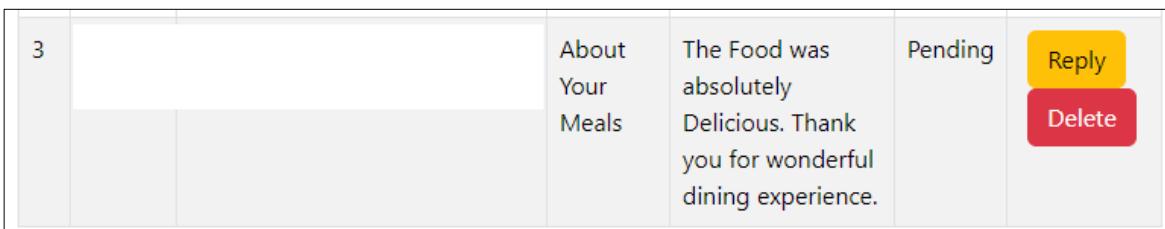


Figure 41 Reservation Accept Proof

<b>No of the Test Case</b>	Test Case 06		
<b>Requirement</b>	Test Case 06: Admin replies to customer queries.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	<ul style="list-style-type: none"> <li>• Name –</li> <li>• Email –</li> <li>• Subject - About Your Meals</li> <li>• Message - The Food was Delicious. Thank you for wonderful dining experience.</li> <li>• State – Pending</li> </ul>	<ul style="list-style-type: none"> <li>• Update States “Pending” to “Replied”</li> <li>• Receiving the reply message to the customer</li> </ul>	Pass



### Reply to Query

Name:

Email:

Subject:

Original Message:

```
The Food was absolutely Delicious. Thank you for wonderful dining experience.
```

Reply Message:

```
Thank you so much for your kind words! We're thrilled to hear that you enjoyed your meal and had a great dining experience with us. We look forward to welcoming you back again soon
```

**Send Reply**

Send Reply Successfully

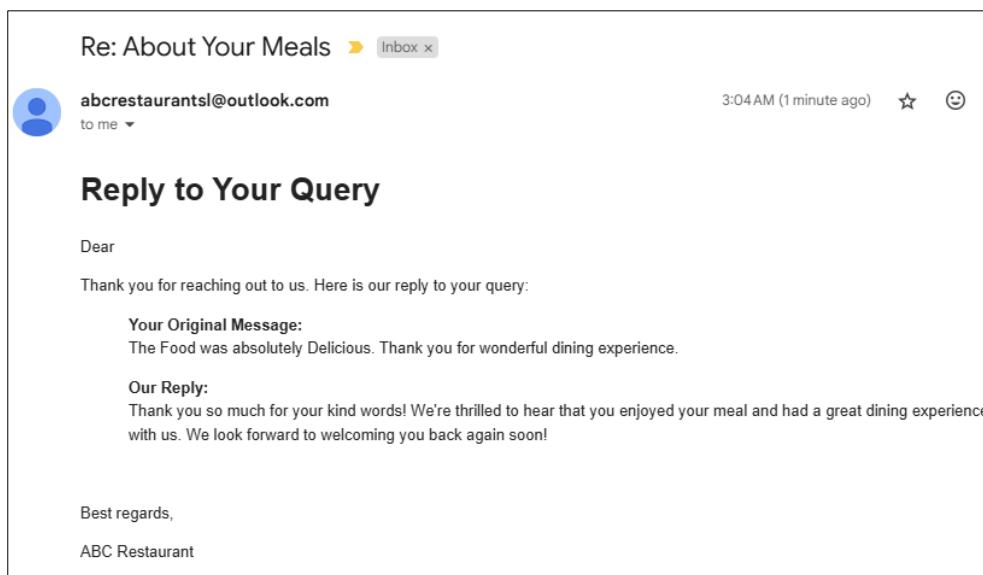


Figure 42 Reply email proof

## Staff Test Case

<b>No of the Test Case</b>	Test Case 01		
<b>Requirement</b>	Test Case 01: Login Staff Member.		
<b>Tester</b>	Application developer		
<b>Input</b>	<b>Data Samples</b>	<b>Expected Result</b>	<b>Received Result</b>
Valid (Positive)	<ul style="list-style-type: none"> <li>Email – <a href="mailto:gayan@gmail.com">gayan@gmail.com</a></li> <li>Password - Gayan@1223</li> </ul>	<ul style="list-style-type: none"> <li>Showing Staff Dashboard</li> </ul>	Pass

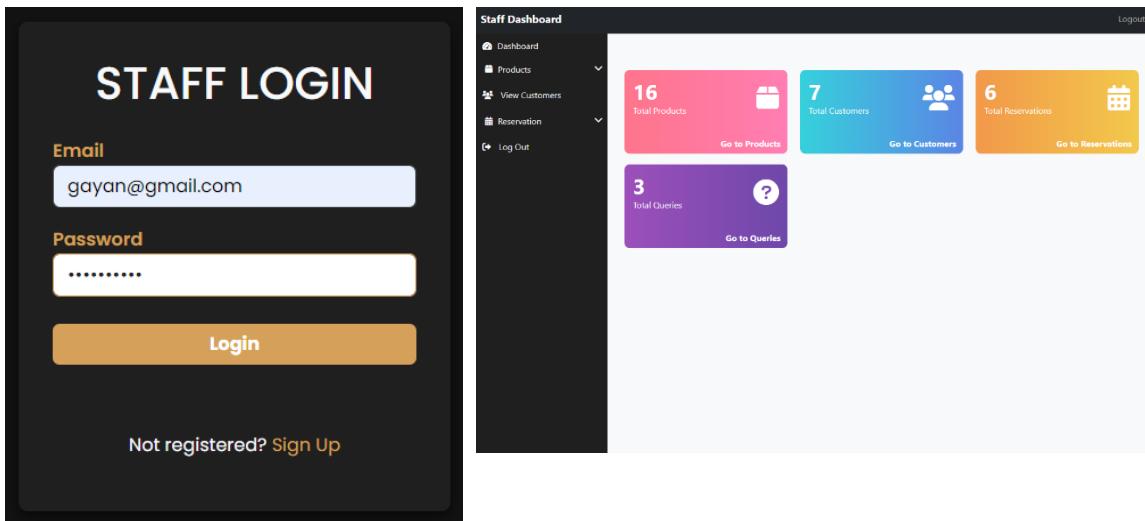


Figure 43 Staff Login Proof

## Task 4

GitHub is a powerful platform that offers comprehensive features for version control, collaboration, security, project management, and more. Additionally, it allows to collaborate efficiently with team members and integrate with many other tools that can improve the overall quality and delivery of project.

### 1. Version Control with Git

- **Branching and Merging:** In GitHub, you can create branches, allowing developers to work on different features, bug fixes, or improvements in isolation. Once the feature or fix is complete, it can be merged back into the main codebase. This avoids potential issues with conflicting changes.

### 2. Team Collaboration Features

- **Pull Requests:** Pull requests are a powerful collaboration feature. When you make a change in a branch, you can create a pull request to merge that branch back into the main project. Other team members can review the code, make comments, and suggest improvements before the merge is approved. This review process improves the quality of code and catches potential bugs or design issues early.

### 3. Continuous Integration and Continuous Delivery (CI/CD)

- **Automation:** GitHub integrates with CI/CD tools like GitHub Actions, Jenkins, and Travis CI, which can automatically run tests, check code quality, and deploy your application whenever new code is committed.

### 4. Project Management and Issue Tracking

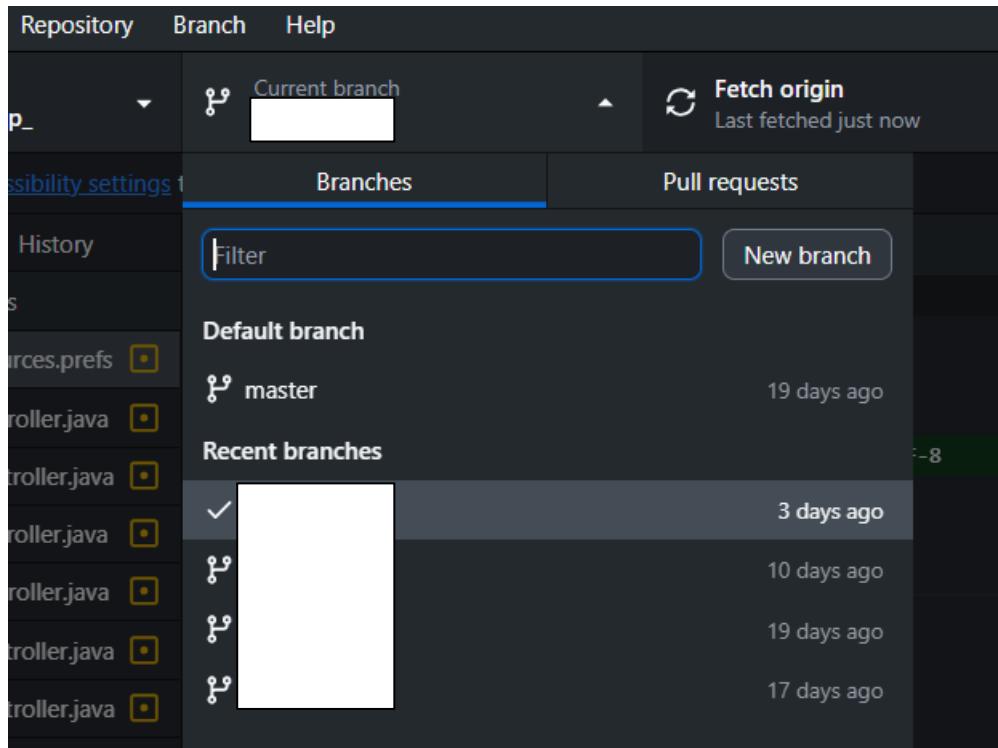
- **GitHub Issues:** Issues help manage bugs, tasks, or feature requests. Each issue can be assigned to specific developers, prioritized with labels, and tracked with deadlines. This allows the team to have a clear overview of what needs to be done and who is responsible.

### 5. Open Source and Contributions

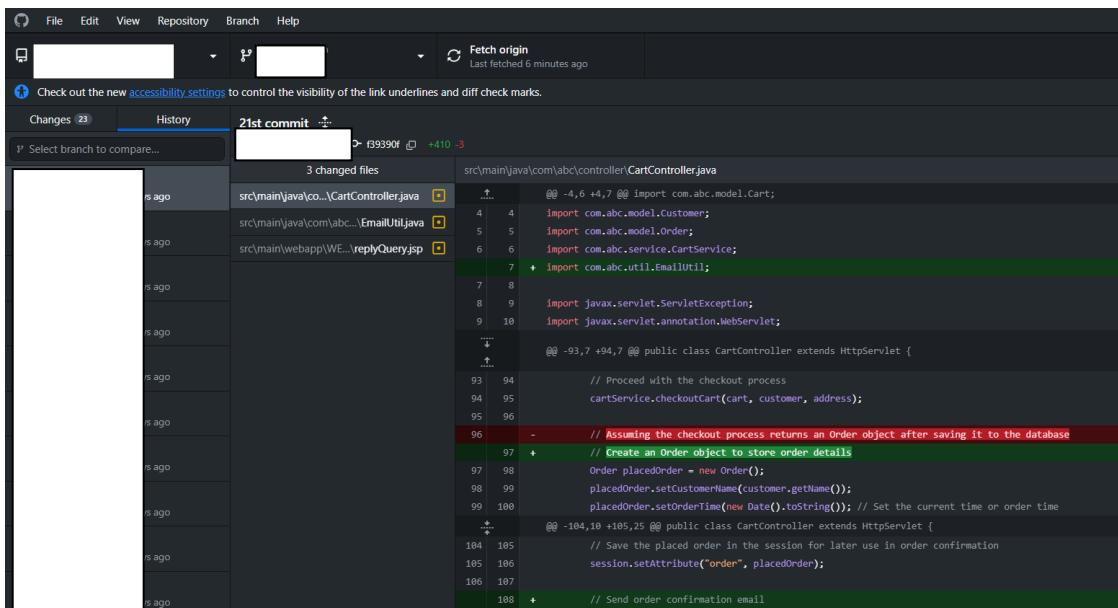
- **Open-Source Projects:** If you choose to make your restaurant reservation project open-source, other developers can fork your repository, contribute code, or suggest improvements.

To do version control using GitHub Desktop

## Branches



## Commits



This screenshot shows a GitHub repository page. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there's a search bar and a pin/unwatch/fork/star button. The main content area shows a commit history table with columns for file, message, author, and date. A message indicates the branch is 30 commits ahead of master. On the right side, there are sections for About (no description), Activity (0 stars, 1 watching, 0 forks), Releases (no releases), Packages (no packages), Languages (Java 73.8%, CSS 21.5%, JavaScript 4.7%), and Suggested workflows (based on tech stack).

File	Message	Author	Date
.settings	send		last week
src/main	21st commit		3 days ago
target	First Commit		3 weeks ago
.classpath	send		last week
.project	First Commit		last month
pom.xml	bug fix 9		3 days ago

**About**  
No description, website, or topics provided.

**Activity**  
0 stars  
1 watching  
0 forks

**Releases**  
No releases published  
[Create a new release](#)

**Packages**  
No packages published  
[Publish your first package](#)

**Languages**  
Java 73.8%  
CSS 21.5%  
JavaScript 4.7%

**Suggested workflows**  
Based on your tech stack

## Conclusion

The ABC Restaurant project effectively demonstrates the creation of an extensive web application designed to assist a chain of restaurants' everyday operations. In order to ensure effective and flawless business operations, this system includes crucial features including role-based access control, product and offer administration, online reservations, and user management.

The application showcases the comprehensive functionality of the system with its user-friendly interface, and it is built on a 3-tier architecture with advanced database capabilities. Design patterns, object-oriented concepts, and UML diagrams are important components that improve the system's maintainability and scalability.

To sum up, the online application ABC Restaurant provides a comprehensive solution for managing many aspects of restaurant operations by combining technical expertise with real-world business knowledge. It not only satisfies the project requirements but also establishes a solid basis for upcoming improvements and practical application.

## References

Aibin, M., 2024. *Baeldung*. [Online]  
Available at: <https://www.baeldung.com/java-factory-pattern>  
[Accessed 10 September 2024].

geeks., G. f., 2024. *Geeks for geeks..* [Online]  
Available at: <https://www.geeksforgeeks.org/singleton-design-pattern/>  
[Accessed 10 September 2024].