**CAKE ONLINE SHOPPING**

**A Project Report**

**Submitted in Partial fulfilment**

**of the Degree of**

**Bachelors of Computer Applications**

**Supervisor’s Name :** Dr . Deepti Khanna **Submitted By :** Chirag Gupta

**Enrolment. No :** 120921014

**Semester :** VIth



**Jagan Nath University**

**Bahadurgarh (NCR)**

**(2021-24)**

**PROJECT CERTIFICATE**

This is to certify that the project report entitled Cake Online Shopping submitted to **JaganNath University, Bahadurgarh** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF COMPUTER APPLICATIONS ( BCA),** is an original work carried out by Chirag Gupta Enrolment No.: 120921014 under the guidance of Dr. Deepti Khanna . The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfilment of the requirement of any course of study.

**Name of the student** : Chirag Gupta **Name of the Guide:** Dr. Deepti Khanna

**Signature of the Student** **Signature of the Guide**

**Enrolment No.:** 120921014  **Date :**

**ACKNOWLEDGEMENT**

I would like to express a deep sense of thanks and gratitude to my project guide Dr. Deepti Khanna ma’am for guiding me immensely through the course of the project. She always evinced keen interest in my work. Her constructive advice and constant motivation have been responsible for the successful completion of this project. My sincere thanks go to Mr Mohit Mathur, our HOD of IT Department for his coordination in extending every possible support for the completion of the project. I also thanks to my parents for their motivation and support. I must thanks to my classmates for their timely help for compilation of this project. Least but not the least, I would like to thank to all those who had helped directly or indirectly towards the completion of the project.

Thanking you

Chirag Gupta

BCA SEM VI

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **TOPIC** | **PAGE NO.** |
| 1 | Introduction | 5 |
| 2 | Objectives | 6 |
| 3 | Tools/Environment | 7 |
| 4 | Analysis Documents | 8-13 |
| 5 | Design Documents | 14-18 |
| 6 | Source code | 19 - 81 |
| 7 | Testing and Validations | 82-83 |
| 8 | Input and Output Screens | 84-87 |
| 9 | Limitations of the project | 88 |
| 10 | Future Application of the project | 89 |
| 11 | Bibliography | 90 |

**INTRODUCTION**

CakeCorner is Crafted using Java Server Pages (JSP) and powered by the XAMPP server stack, CakeCorner brings the joy of indulgence right to your fingertips. Seamlessly navigate through our user-friendly interface, where you can effortlessly login or sign up to explore a delectable array of cakes. From classic Flavors to innovative creations, CakeCorner offers something to tantalize every palate. Simply add your favourites to the cart, breeze through checkout. Stay informed with our comprehensive information section and easily track your orders. With CakeCorner, every click is a step closer to sweet satisfaction. CakeCorner ensures a secure browsing experience with user authentication functionality. Users can register for an account or log in securely to access personalized features and preferences. Our platform boasts a user-friendly interface designed to streamline the browsing and shopping process. With easy navigation and intuitive controls, finding and selecting your favourite cakes is a piece of cake. Adding your favourite cakes to your cart is quick and hassle-free. Simply browse, select, and proceed to checkout with ease. Receive detailed invoices for your purchases, making it easy to track your expenses and manage your budget effectively. Keep track of your past orders and monitor the status of your current ones with our convenient order tracking feature. Never miss a beat with CakeCorner.

**OBJECTIVES**

The objective of the Cake Online Shopping project, built on JSP and XAMPP, is to create a robust and user-friendly platform for customers to explore, select, and purchase cakes online. The project aims to seamlessly integrate essential functionalities like user authentication for secure login and signup processes, ensuring users can access their accounts and preferences effortlessly. A user-centric interface will be designed, emphasizing intuitive navigation and clear categorization, enabling users to browse through a comprehensive catalogue of cakes conveniently. Features such as the shopping cart will streamline the selection process, allowing users to review and modify their orders before proceeding to checkout. Additionally, users will have the ability to manage their personal information and addresses for efficient order processing and delivery. The project will generate detailed invoices for each transaction, offering users a transparent view of their purchases, with the option to print bills for their records. Furthermore, an order management system will enable users to track their orders and view their order history, enhancing their overall shopping experience. Ultimately, the Cake Online Shopping project endeavours to provide a seamless and enjoyable platform for users to indulge in their favourite treats with ease and confidence.

**TOOLS/ENVIRONMENT**

|  |  |
| --- | --- |
| **System** | **Software/Hardware Requirements** |
| Client/server Operating System | Window 11 |
| Language Used | java |
| Frontend Used | Java Server Pages (JSP) |
| Backend Used | Xampp Server |
| Software Used | Eclipse |
| RAM | Minimum 4 GB |
| Disk Size | Minimum 2 GB |

**ANAYLSIS DOCUMENT**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**INDEX**

1. **Introduction**
   1. Purpose
   2. Problem statement
   3. Scope of the project
   4. Intended Audience

1. **Overall Description**
   1. Product Perspective
   2. Product Functions
   3. Dependencies
2. **Functional Requirements**
   1. Login
   2. View home page with cakes
   3. Add cakes to cart
   4. Open cart
   5. Payment Information and address
   6. Bill
   7. signIn
   8. My orders
3. **Non-Functional Requirements**
   1. Performances
   2. Security
   3. Reliability
   4. Usability
4. **Conclusion**

**1.INTRODUCTION**

1.1 **Purpose:**

Streamline cake purchasing with a user-friendly online platform built on JSP, offering a seamless shopping experience.Enhance accessibility and convenience by enabling customers to browse, select, and order delectable cakes from the comfort of their homes.Empower businesses to expand their reach and boost sales by establishing a robust full-stack solution that integrates secure transactions and dynamic content presentation through JSP technology.

1.2 **Problem Statement**

Despite the growing demand for online cake purchases, existing platforms lack a cohesive full-stack solution, leading to fragmented user experiences and inefficient transaction processes.Limited availability of comprehensive online cake shopping platforms built on JSP hampers businesses' ability to effectively showcase their products and engage with customers in a dynamic digital environment.Inadequate integration of frontend and backend technologies in current online cake shopping systems using JSP results in performance bottlenecks, security vulnerabilities, and suboptimal user interactions, hindering business growth and customer satisfaction.

1.3 S**cope of the project:**

The project aims to develop a comprehensive full-stack solution for online cake shopping using JSP, encompassing frontend design, backend functionality, and secure transaction processing. It involves creating an intuitive user interface for browsing cakes, implementing a robust database system for managing products and orders, and integrating secure payment gateways for seamless transactions.

1.4 **Intended Audience:**

The intended audience for the cake online shopping project includes both consumers seeking to purchase cakes conveniently online and businesses looking to expand their reach in the digital market. Consumers of various demographics and preferences who value convenience and a wide selection of cakes will benefit from the user-friendly interface and secure transaction capabilities.

**2. OVERALL DESCRIPTION**

2.1 **Product perspective:**

The cake online shopping project, built on a full-stack architecture using JSP, serves as a comprehensive platform bridging customers and businesses in the bakery industry. From a product perspective, it integrates seamlessly with existing bakery management systems, offering an enhanced digital storefront for showcasing cakes and facilitating online transactions. Its scalability allows for future expansion and integration with emerging technologies, ensuring longevity and adaptability in a dynamic market landscape.

2.2 **Product Functions:**

The cake online shopping project, leveraging JSP for full-stack development, encompasses essential functions vital for seamless transactions and user interaction. Users can register and sign in securely to access personalized features, including a user-friendly cart for selecting desired cakes. The platform facilitates smooth payment processing, ensuring transactions are encrypted and protected. Upon completion, users receive detailed bills summarizing their purchases, enhancing transparency and trust in the shopping experience.

2.3 **Dependencies:**

The cake online shopping project's successful implementation relies on several dependencies to ensure its functionality and performance. Primarily, it requires a stable web server environment such as Apache Tomcat for hosting JSP pages and servlets. Additionally, the project depends on Java for backend logic and database connectivity, often integrated with technologies like JDBC for database interaction. Moreover, frontend development necessitates HTML, CSS, and JavaScript for crafting dynamic and responsive user interfaces, completing the comprehensive full-stack architecture essential for a seamless online shopping experience.

**3.Functional requirements**

3.1 **Login**

In the login functionality of the cake online shopping project, users must be able to securely authenticate their identities to access personalized features and account information. This entails implementing user registration, login, and logout processes with robust password encryption and session management to ensure data security. Additionally, features like password recovery and account management functionalities, such as profile editing and deletion, should be included to enhance user experience and maintain account integrity throughout their interactions with the platform.

3.2 **View home page with cakes**

In the home page functionality of the cake online shopping project, users should be greeted with an enticing display of featured cakes, showcasing variety and quality. This requires dynamic content presentation, allowing users to easily browse and filter cakes based on categories, flavors, and occasions. Additionally, integration with a backend database is essential for real-time updates on cake availability and pricing, ensuring a seamless and engaging shopping experience from the moment users land on the homepage.

3.3 **Add cakes to cart**

The "Add to Cart" functionality in the cake online shopping project enables users to effortlessly select desired cakes for purchase. It should allow for seamless addition of items to the cart, including options for quantity selection and flavor variations. Integration with backend systems ensures accurate inventory management and updates cart contents in real-time, providing users with a transparent view of their selections. Additionally, the feature should support easy modification and removal of items from the cart before proceeding to checkout, enhancing user flexibility and satisfaction.

3.4 **Open cart**

The "Open Cart" functionality in the cake online shopping project allows users to view and manage items they've added for purchase. It should provide a clear summary of selected cakes, including quantities, flavors, and prices. Users should be able to easily modify quantities or remove items, with changes reflected in real-time. Integration with backend systems ensures accurate calculation of subtotal, taxes, and total cost, empowering users with transparent and convenient control over their shopping carts.

3.5 **Payment Information and address**

In the payment and address info section of the cake online shopping project, users should securely input their payment details and shipping addresses. This functionality requires integration with secure payment gateways to facilitate transactions, ensuring encryption and compliance with industry standards for data protection. Additionally, users should have the option to save multiple addresses for convenient future purchases, enhancing flexibility and streamlining the checkout process. Error handling mechanisms should be in place to address any input mistakes and provide clear guidance for completing the transaction successfully.

3.6 **Bill**

In the bill functionality of the cake online shopping project, users should receive a comprehensive summary of their purchase, including itemized details, prices, taxes, and total cost. This feature ensures transparency and clarity regarding the transaction. Additionally, users should have the option to print or download the bill for their records. Integration with backend systems ensures accuracy in billing calculations and generates a professional and user-friendly invoice, enhancing customer satisfaction and trust in the platform.

3.7 **SignIn**

The "Sign In" function of the cake online shopping project enables users to securely access their accounts. It should provide a user-friendly interface for entering credentials and include features for password recovery and account registration if necessary. Integration with backend systems ensures authentication accuracy and session management for secure user interactions throughout the shopping experience. Additionally, error handling mechanisms should be implemented to guide users in case of incorrect login attempts, enhancing usability and security.

3.8 **My orders**

The "My Orders" functionality in the cake online shopping project allows users to view their order history and track current orders. It should provide a clear and organized display of past purchases, including order details such as date, items purchased, and order status. Integration with backend systems ensures real-time updates on order statuses, enhancing transparency and user satisfaction. Additionally, users should have the option to filter and search through their order history for quick access to specific transactions, improving usability and convenience.

**4.Non-Functional Requirements**

4.1 **Performance:**

In terms of non-functional requirements for performance in the cake online shopping project, it's essential to ensure fast response times, even under peak load conditions. This involves optimizing database queries, caching frequently accessed data, and employing efficient algorithms to minimize processing time. Additionally, the system should be scalable to accommodate growing user traffic and should undergo rigorous performance testing to identify and address any bottlenecks.

4.2 **Security:**

In terms of security non-functional requirements for the cake online shopping project, stringent measures should be implemented to safeguard user data and transactions. This includes encryption of sensitive information such as passwords and payment details, adherence to industry-standard security protocols like HTTPS, and protection against common web vulnerabilities such as SQL injection and cross-site scripting. Additionally, access control mechanisms should be in place to restrict unauthorized access to system resources, and regular security audits and updates should be conducted to mitigate emerging threats and vulnerabilities, ensuring a robust and resilient security posture for the platform.

4.3 **Reliability:**

Reliability in the cake online shopping project necessitates a stable and consistent user experience, with minimal downtime and errors. This entails deploying redundant systems for high availability, implementing automated monitoring and alerting to detect and resolve issues promptly, and conducting regular backups to prevent data loss. Additionally, the system should gracefully handle unexpected errors and recover seamlessly to ensure uninterrupted service for users. Adherence to industry best practices and standards in software development and maintenance is crucial to maintaining reliability and building trust among users.

4.4 **Usability**

In a cake online shopping platform developed with JSP and XAMPP, non-functional requirements for usability emphasize fast page loading times to enhance user experience. Responsive design ensures smooth navigation across various devices, optimizing accessibility. Intuitive search and filtering options streamline product discovery, maximizing user satisfaction and retention.

**5.Conclusion**

In summary, harnessing the power of JSP and XAMPP for cake online shopping results in a dynamic and versatile platform. Through JSP's flexibility and XAMPP's comprehensive suite of tools, developers can create a seamless browsing and purchasing experience. However, vigilance in security measures and continuous optimization is essential to uphold user trust and satisfaction. Ultimately, this collaborative approach empowers businesses to offer a delightful online shopping experience, catering to the cravings of cake lovers while ensuring scalability and reliability for future growth.

**DESIGN DOCUMENT**

**ENTITY RELATIONSHIP**

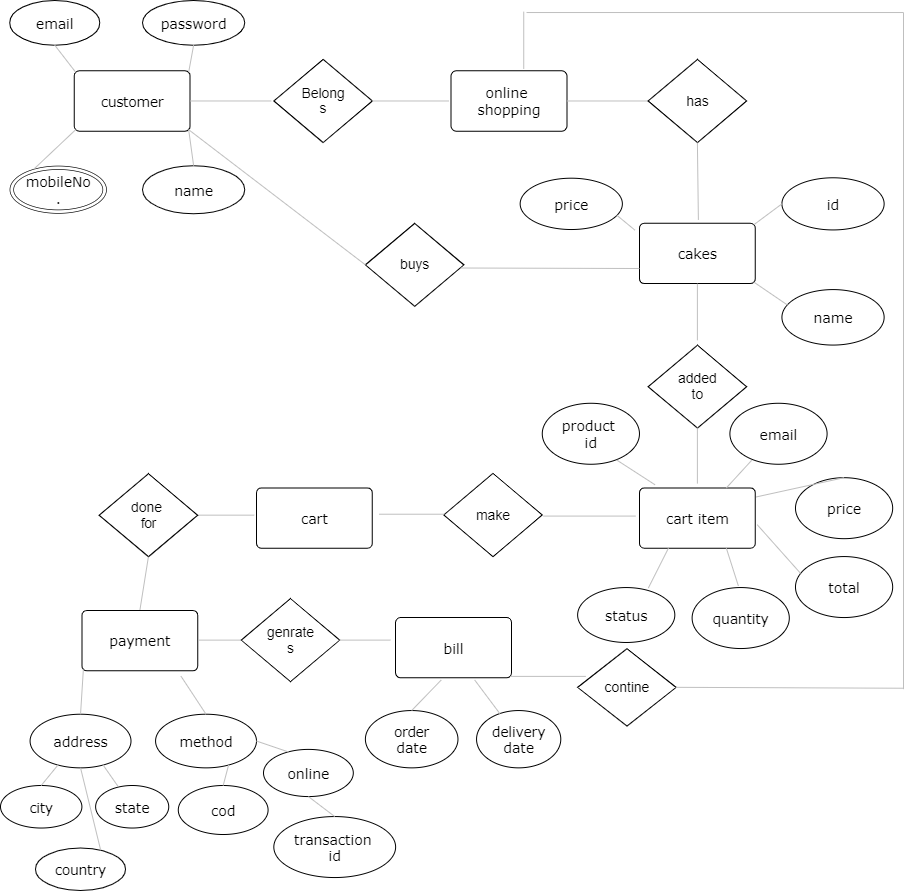


Fig 5.1 Entity Relationship diagram for cake online shopping

**DATA FLOW DIAGRAM(LEVEL 0)**

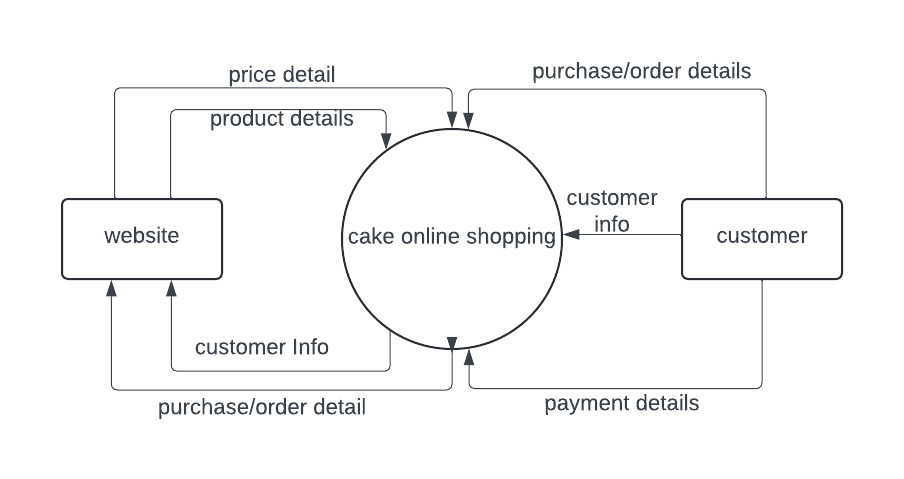


Fig 5.2 data flow diagram for cake online shopping

**DATA FLOW DIAGRAM(LEVEL 1)**

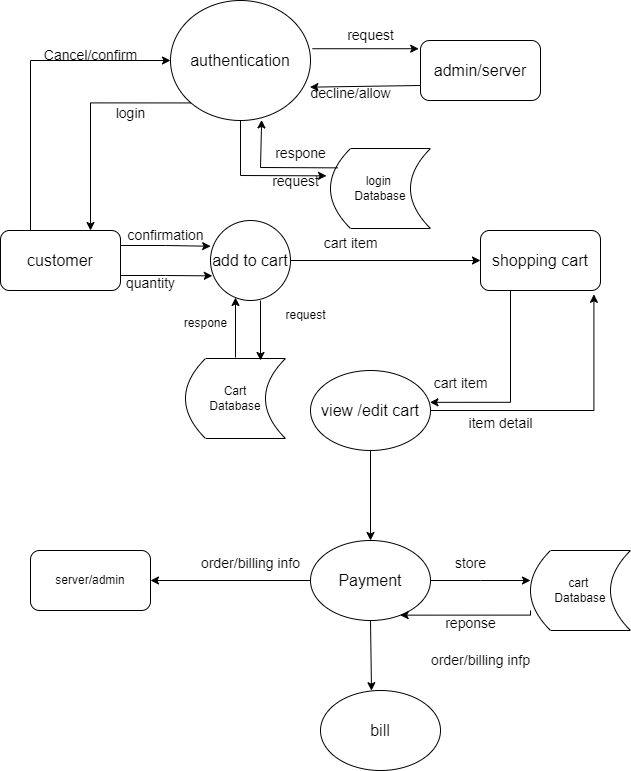


Fig 5.3 Data flow diagram level 2 of cake online shopping

**USE CASE DIAGRAM**

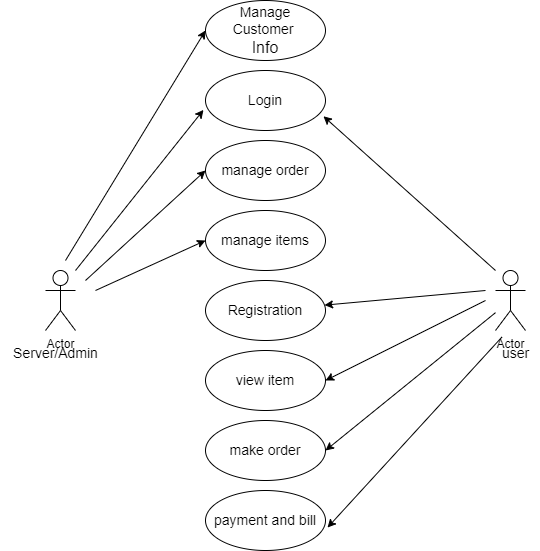
****

Fig 5.4 User case diagram of cake online shopping

**DATA DICTIONARY**

1. **Cakes Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN** | **TYPE** | **NULL** | **DESCIPTION** |
| Id | Int(15) | No | Id of the cake |
| Name | Varchar(30) | No | Name of the cake |
| Price | Int(30) | No | Price of the cake |

Fig 5.5 Data Dictionary for cakes table

1. **LogIn Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN** | **TYPE** | **NULL** | **DESCRIPTION** |
| Name | Varchar(50) | Yes | Name of the user |
| Email | Varchar(50) | No | Email of the user |
| Mobilenumber | Bigint(30) | Yes | Mobile of user |
| Password | Varchar(50) | Yes | Password by user |
| Address | Varchar(100) | Yes | Address of user |
| City | Varchar(50) | Yes | City of user |
| State | Varchar(50) | Yes | State of user |
| Country | Varchar(50) | Yes | Country of user |

Fig 5.6 Data dictionary of login table

1. **Cart Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN** | **TYPE** | **NULL** | **DESCRIPTION** |
| Email | Varchar(100) | Yes | Email of user |
| Product\_id | int(11) | Yes | Id of the cake |
| Quantity | Int(11) | Yes | Quantity of order |
| Price | Int(11) | Yes | Price of cake |
| Total | Int(11) | Yes | Total of order |
| Address | Varchar(500) | Yes | Address to delivery |
| City | Varchar(100) | Yes | City of user |
| State | Varchar(100) | Yes | State of user |
| Country | Varchar(100) | Yes | County of user |
| Mobile | Bigint(20) | Yes | Mobile of user |
| Orderdate | Varchar(100) | Yes | When user order |
| Deliverydate | Varchar(100) | Yes | Date of delivery |
| Paymentmethod | Varchar(100) | Yes | Payment by user |
| Transactioinid | Varchar(100) | Yes | Online id |
| Status | Varchar(10) | Yes | Status of order |

Fig 5.7 Data Dictionary of cart table

**Source Code**

**Web.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Cake Corner</title>

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

</head>

<body class="showcart">

<div class="web">

<img src="cakes/logo.png" class="logo">

<% String email = (String)session.getAttribute("email");

int total=0;

try {

Class.forName("com.mysql.jdbc.Driver");

try (Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "")) {

Statement st=conn.createStatement();

%>

<h1>CakeCorner</h1>

<ul>

<li><img src="cakes/icons8-logout-50.png" style="height: 45px; width: 35px;"></li>

<li><a href="cakes.jsp">Log Out</a></li>

<li><img src="cakes/icons8-contact-us-32.png" href="#c1" style="height: 45px; width: 35px;"></li>

<li><a href="#c1">Contact Us</a></li>

<% ResultSet rs1 = st.executeQuery("SELECT SUM(quantity) FROM cart WHERE email='" + email + "' AND address IS NULL");

while(rs1.next())

{

total=rs1.getInt(1);

}

%>

<li><div class="icon-cart"><img src="cakes/iconscart.png" href="mycart.jsp" style="height: 30px; width: 25px;"><span><%out.println(total); %></span></div></li>

<li> <a href="mycart.jsp">My Cart</a></li>

<li><img src="cakes/purchase-order-24.png" href="myorder.jsp" style="height: 45px; width: 30px;"></li>

<li><a href="myorder.jsp">My Orders</a></li>

<li><img src="cakes/icons8-time-machine-50.png" style="height: 45px; width: 35px;"></li>

<li> <div id="last-visited">Last visited: <span id="last-visited-time"></span></div>

<script>

document.addEventListener("DOMContentLoaded", function() {

if (typeof(Storage) !== "undefined") {

if (localStorage.getItem("lastVisited")) {

var lastVisitedTime = localStorage.getItem("lastVisited");

document.getElementById("last-visited-time").textContent = lastVisitedTime;

} else {

document.getElementById("last-visited-time").textContent = "Never visited before";

}

var currentDate = new Date();

var dateString = currentDate.toLocaleString();

localStorage.setItem("lastVisited", dateString);

} else {

document.getElementById("last-visited-time").textContent = "Sorry, your browser does not support Web Storage.";

}

});

</script>

</li>

<li><img src="cakes/user-24.png" style="height: 45px; width: 35px;"></li>

<li><p style="color:white;"><% out.println(email); %></p></li>

</ul><%

PreparedStatement pst = conn.prepareStatement("SELECT \* FROM cakes WHERE id = ?");

pst.setInt(1, 1);

ResultSet rs = pst.executeQuery();

while (rs.next()) {

%>

<div class="container">

<div class="box">

<img src="cakes/cheesecake.jpg" height="250px" width="250px" alt="cheesecake" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs.getString(1) %>

<b><%= rs.getString(2) %></b>

"Indulge in a slice of creamy, dreamy cheesecake bliss!<br>

<b> Price <%= rs.getString(3) %>/kg</b>

<button class="button"><a href="addtocartaction.jsp?id=<%= rs.getString(1) %>">Add to cart</a></button>

</div>

<%

}

pst.setInt(1, 2);

ResultSet rs2 = pst.executeQuery();

while (rs2.next()) {

%>

<div class="box">

<img src="cakes/blackforest.jpg" height="250px" width="250px" alt="blackforest" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs2.getString(1) %>

<b><%= rs2.getString(2) %></b>

"Indulge in the rich layers of chocolate and cherries with every bite of our Black Forest Cake!"<br>

<b> Price <%= rs2.getString(3) %>/kg</b>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs2.getString(1) %>">Add to cart</a></button>

</div>

<%

}

pst.setInt(1, 3);

ResultSet rs3 = pst.executeQuery();

while (rs3.next()) {

%>

<div class="box">

<img src="cakes/brownie.jpg" height="250px" width="250px" alt="brownie" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs3.getString(1) %>

<b><%= rs3.getString(2) %></b>

"Indulge in a slice of rich, gooey brownie cake, a blissful blend of chocolatey decadence.<br>

<b> Price <%= rs3.getString(3) %>/kg</b>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs3.getString(1) %>">Add to cart</a></button>

</div><br>

<%

}

pst.setInt(1, 4);

ResultSet rs4 = pst.executeQuery();

while (rs4.next()) {

%>

<div class="box">

<img src="cakes/blueberry.jpg" height="250px" width="250px" alt="blueberry" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs4.getString(1) %>

<b><%= rs4.getString(2) %></b>

"Indulge in the exquisite taste of blueberry cake, a burst of fruity bliss in every bite!"<br>

<B> Price <%= rs4.getString(3) %>/kg</B>

<button class="button"><a href="addtocartaction.jsp?id=<%= rs4.getString(1) %>">Add to cart</a></button>

</div>

</div>

<%

}

pst.setInt(1, 5);

ResultSet rs5 = pst.executeQuery();

while (rs5.next()) {

%>

<div class="container">

<div class="box">

<img src="cakes/icecream.jpg" height="250px" width="250px" alt="icecream" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs5.getString(1) %>

<b><%= rs5.getString(2) %></b>

"Indulge in the heavenly fusion of creamy ice cream and decadent cake layers, a divine treat for your taste buds!<br>

<b> Price <%= rs5.getString(3) %>/kg</b>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs5.getString(1) %>">Add to cart</a><br></button>

</div>

<%

}

pst.setInt(1, 6);

ResultSet rs6 = pst.executeQuery();

while (rs6.next()) {

%>

<div class="box">

<img src="cakes/chocolate.jpg" height="250px" width="250px" alt="chocolate" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs6.getString(1) %>

<b><%= rs6.getString(2) %></b>

"Indulge in a slice of rich, decadent chocolate cake and let your taste buds dance with delight!"<br>

<b> Price <%= rs6.getString(3) %>/kg</b>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs6.getString(1) %>">Add to cart</a></button>

</div>

<%

}

pst.setInt(1, 7);

ResultSet rs7 = pst.executeQuery();

while (rs7.next()) {

%>

<div class="box">

<img src="cakes/fruit.jpg" height="250px" width="250px" alt="fuit" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs7.getString(1) %>

<b><%= rs7.getString(2) %></b>

"Indulge in a slice of heaven with our luscious fruit cake, bursting with flavor in every bite!"<br>

<b> Price <%= rs7.getString(3) %>/kg</b>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs7.getString(1) %>">Add to cart</a></button>

</div>

<%

}

pst.setInt(1, 8);

ResultSet rs8 = pst.executeQuery();

while (rs8.next()) {

%>

<div class="box">

<img src="cakes/pineapple.jpg" height="250px" width="250px" alt="pineapple" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs8.getString(1) %>

<b><%= rs8.getString(2) %></b>

"Savor the tropical sweetness with every bite of our luscious pineapple cake!<br>

<b> Price <%= rs8.getString(3) %>/kg</b>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs8.getString(1) %>">Add to cart</a></button>

</div>

</div>

<%

}

pst.setInt(1, 9);

ResultSet rs9 = pst.executeQuery();

while (rs9.next()) {

%>

<div class="container">

<div class="box">

<img src="cakes/rainbow.jpg" height="250px" width="250px" alt="rainbow" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs9.getString(1) %>

<b><%= rs9.getString(2) %></b>

"Indulge in a slice of happiness with our vibrant rainbow cake!"<br>

<b>Price <%= rs9.getString(3) %>/kg</b>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs9.getString(1) %>">Add to cart</a><br></button>

</div>

<%

}

pst.setInt(1, 10);

ResultSet rs10 = pst.executeQuery();

while (rs10.next()) {

%>

<div class="box">

<img src="cakes/redvelvet.jpg" height="250px" width="250px" alt="redvelvet" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs10.getString(1) %>

<b><%= rs10.getString(2) %></b>

"Indulge in a slice of happiness with our vibrant rainbow cake!"<br>

<b> Price <%= rs10.getString(3) %>/kg</b>

<button class="button"><a href="addtocartaction.jsp?id=<%= rs10.getString(1) %>">Add to cart</a></button>

</div>

<%

}

pst.setInt(1, 11);

ResultSet rs11 = pst.executeQuery();

while (rs11.next()) {

%>

<div class="box">

<img src="cakes/stawberry.jpg" height="250px" width="250px" alt="strawberry" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs11.getString(1) %>

<b><%= rs11.getString(2) %></b>

"Indulge in the luscious delight of strawberry cake, a sweet symphony of freshness and flavor!"<br>

<B> Price <%= rs11.getString(3) %>/kg</B>

<button class="button"> <a href="addtocartaction.jsp?id=<%= rs11.getString(1) %>">Add to cart</a>

</div>

<%

}

pst.setInt(1, 12);

ResultSet rs12 = pst.executeQuery();

while (rs12.next()) {

%>

<div class="box">

<img src="cakes/butterscotch.jpg" height="250px" width="250px" alt="butterscotch" onmouseover="enlarge(this)" onmouseleave="resize(this)">

<%= rs12.getString(1) %>

<b><%= rs12.getString(2) %></b>

"Delight in the rich and creamy goodness of butterscotch cake, a taste of pure indulgence!"<br>

<b>Price <%= rs12.getString(3) %>/kg</b>

<button class="button"><a href="addtocartaction.jsp?id=<%= rs12.getString(1) %>">Add to cart</a></button>

</div>

</div>

<script>

function enlarge(p) {

p.style.height = "300px";

p.style.width = "300px";

}

function resize(p)

{

p.style.height = "250px";

p.style.width = "250px";

}

</script>

<%

}

}

} catch (Exception e) {

out.println("Error: " + e.getMessage());

}

%>

<footer id="c1">

<pre>

<b>Get in Touch Address Contact No. </b>

cakecorner@gmail.com B-20 Jhilmil colony, +91-9821998409

Cakecorner11@yahoo.in Delhi-95 +011-22163925

It’s all about the memories.

</pre>

</footer>

</body>

</html>

**Signup.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>CakeCorner</title>

</head>

<body>

<%

String name=request.getParameter("name");

String mail=request.getParameter("mail");

String password=request.getParameter("password");

String mobile=request.getParameter("phone");

if (mail != null && !mail.isEmpty()) {

try {

Class.forName("com.mysql.jdbc.Driver");

try (Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "")) {

String sql = "INSERT INTO login (name,email,password,mobilenumber) values (?,?,?, ?)";

try (PreparedStatement ps = conn.prepareStatement(sql)) {

ps.setString(1, name);

ps.setString(2, mail);

ps.setString(3, password);

ps.setString(4, mobile);

int x = ps.executeUpdate();

response.sendRedirect("cakes.jsp");

}

}

} catch (Exception e) {

out.println("Error: " + e);

}

} else {

out.println("Error: Username cannot be null or empty.");

}

%>

</body>

</html>

**Signup.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>SignUp.CakeCorner</title>

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

</head>

<body>

<h1>CakeCorner</h1>

<img src="cakes/logo.png" class="logo">

<div class="container">

<form action="signup.jsp" method="post" class="form">

<h2>Sign In</h2>

<input type="text" class="box" name="name" placeholder="enter name" required>

<input type="email" class="box" placeholder="Mail" name="mail" pattern="[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}" title="Please enter a valid email address"required>

<input type="tel" id="phone" name="phone" pattern="[0-9]{10}" placeholder="enter the mobile number" class="box" title="Please enter a 10-digit phone number" required>

<input type="password" class="box" placeholder="password" name="password" required>

<input type="submit" class="box" value="signup" id="submit">

</form>

<div class="image"><img src="cakes/images.png"></div>

</div>

</body>

</html>

**Remove.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@ page import="java.sql.\*" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<%

String email=session.getAttribute("email").toString();

String product\_id=request.getParameter("id");

try{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

Statement st=conn.createStatement();

st.executeUpdate("delete from cart where email='"+email+"' and product\_id='"+product\_id+"' and address is NULL");

response.sendRedirect("mycart.jsp?msg=removed");

}

catch(Exception e)

{

System.out.println(e);

}

%>

</body>

</html>

**Payment.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>CakeCorner</title>

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

</head>

<body>

<h1>CakeCorner</h1>

<img src="cakes/logo.png" class="logo">

<script>

if(window.history.forward(1)!+null)

window.history.forward(1)</script>

<%

String email=session.getAttribute("email").toString();

int total=0;

int sno=0;

try {

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

Statement st=conn.createStatement();

ResultSet rs1=st.executeQuery("select sum(total) from cart where email='"+email+"' and address is NULL");

while(rs1.next()) {

total=rs1.getInt(1);

}

%>

<a href="mycart.jsp" style="font-size: 24px; font-family: Arial, sans-serif; color: purple; text-decoration: none; transition: color 0.3s;"><u>Back</u></a>

<hr>

<table id="payment">

<thead>

<tr>

<th scope="col">S.No</th>

<th scope="col">Product name</th>

<th scope="col">Price</th>

<th scope="col">Quantity</th>

<th scope="col">Sub Total</th>

</tr>

</thead>

<tbody>

<%

ResultSet rs2=st.executeQuery("select \* from cakes inner join cart on cakes.id=cart.product\_id and cart.email='"+email+"' and cart.address is Null");

while(rs2.next()) {

sno++;

%>

<tr>

<td><%= sno %></td>

<td><%= rs2.getString(2) %></td>

<td><%= rs2.getString(3) %></td>

<td><%= rs2.getString(6) %></td>

<td><%= rs2.getString(8) %></td>

</tr>

<%

}

ResultSet rs3=st.executeQuery("select \* from login where email='"+email+"'");

if(rs3.next()) {

%>

</tbody>

</table><br>

<hr>

<div style="text-align: center;"><h2 class="invoice">Total:<%= total %></h2></div>

<hr><br>

<form action="addresspayment.jsp" method="post">

<center><h2 style="font-family:courier;">Order Details:</h2></center>

<h3>Address</h3>

<input type="text" name="address" value="<%= rs3.getString(5) %>" placeholder="Enter address" required>

<h3>Enter the city</h3>

<input type="text" name="city" value="<%= rs3.getString(6) %>" placeholder="Enter city" required>

<h3>State</h3>

<input type="text" name="state" value="<%= rs3.getString(7) %>" placeholder="Enter state" required>

<h3>Country</h3>

<input type="text" name="country" value="<%= rs3.getString(8) %>" placeholder="Enter country" required>

<h3>Select way of payment</h3>

<select name="paymentmethod">

<option value="Cash on Delivery(cod)">Cash on Delivery(cod)</option>

<option value="Online Payment">Online Payment</option>

</select>

<h3>Pay on cakecorner@paytm.com</h3>

<input type="text" name="transactionid" placeholder="Transaction Id">

<h5>If you select online payment, enter the transaction ID; otherwise, leave this blank</h5>

<h3>Mobile Number</h3>

<input type="tel" id="phone" name="phone" pattern="[0-9]{10}" value="<%= rs3.getString(3) %>" placeholder="Enter the mobile number" title="Please enter a 10-digit phone number" required>

<button type="submit">Proceed to generate bill & save</button>

</form>

<%

}

}catch(Exception e) {

System.out.println(e);

}

%>

</body>

</html>

**Payment.css**

@font-face {

font-family: myFont;

src: url(cakes/Cedarville-Cursive.ttf);

}

h3, h5 {

font-family: myFont, sans-serif;

}

h1{

font-family: myFont;

color: purple;

}

body {

background-color: #f2f2f2;

}

a:hover {

color: #6a5acd;

}

.invoice{

font-family:courier;

}

.logo {

height: 100px;

width: 150px;

position: absolute;

top: 0;

right: 10px;

}

#payment {

font-family: Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

#payment td, #payment th {

border: 1px solid #ddd;

padding: 8px;

}

#payment tr:nth-child(even) {

background-color: #f2f2f2;

}

#payment tr:hover {

background-color: #ddd;

}

#payment th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: purple;

color: white;

}

form {

max-width: 600px;

margin: 0 auto;

padding: 20px;

background-color: #f2f2f2;

border-radius: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

margin-top: 20px;

}

form h3, form h5 {

color: purple;

font-family: 'Montserrat', sans-serif;

}

form input[type="text"], form select {

width: calc(100% - 22px);

padding: 10px;

margin: 10px 0;

border: 2px solid #ddd;

border-radius: 5px;

box-sizing: border-box;

}

form button[type="submit"] {

width: 100%;

padding: 10px;

margin-top: 10px;

border: none;

border-radius: 5px;

background-color: purple;

color: white;

cursor: pointer;

}

form button[type="submit"]:hover {

background-color: #6a5acd;

}

**Myorder.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>CakeCorner</title>

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

</head>

<body>

<h1>CakeCorner</h1>

<img src="cakes/logo.png" class="logo">

<% String email=session.getAttribute("email").toString();%>

<table id="payment">

<thead>

<tr>

<th scope="col">S.no</th>

<th scope="col">Product name</th>

<th scope="col">price</th>

<th scope="col">quantity</th>

<th scope="col">Sub total</th>

<th scope="col">order date</th>

<th scope="col">Expected delivery date</th>

<th scope="col">payment method</th>

<th scope="col">Transaction Id</th>

<th scope="col">Status</th>

</tr>

</thead>

<tbody>

<%

int sno=0;

try{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

Statement st=conn.createStatement();

ResultSet rs=st.executeQuery("select \* from cart inner join cakes on cart.product\_id=cakes.id where cart.email='"+email+"' and cart.orderdate is not NULL");

while(rs.next()) {

sno++;

%>

<tr>

<td><%= sno %></td>

<td><%= rs.getString("name") %></td>

<td><%= rs.getString("price") %></td>

<td><%= rs.getString("quantity") %></td>

<td><%= rs.getString("total") %></td>

<td><%= rs.getString("orderdate") %></td>

<td><%= rs.getString("deliverydate") %></td>

<td><%= rs.getString("paymentmethod") %></td>

<td><%= rs.getString("transactionid") %></td>

<td><%= rs.getString("status") %></td>

</tr>

<%

}

}

catch(Exception e) {

out.println(e);

}

%>

</tbody>

</table>

</body>

<footer id="c1">

<pre>

<b>Get in Touch Address Contact No. </b>

cakecorner@gmail.com B-20 Jhilmil colony, +91-9821998409

Cakecorner11@yahoo.in Delhi-95 +011-22163925

It’s all about the memories.

</pre>

</footer>

</html>

**Mycart.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

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

<meta charset="UTF-8">

<title>CakeCorner</title>

</head>

<body>

<h1>CakeCorner</h1>

<img src="cakes/logo.png" class="logo">

<%

String email = session.getAttribute("email").toString();

int total = 0;

int sno = 0;

try {

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

Statement st = conn.createStatement();

ResultSet rs1 = st.executeQuery("select sum(total) from cart where email='" + email + "' and address is NULL");

while (rs1.next()) {

total = rs1.getInt(1);

}

%>

<hr>

<div style="text-align: center;"><h2 class="invoice">Total:<%out.println(total); %></h2></div>

<hr>

<% if(total > 0) { %>

<a href="payment.jsp" style="font-size: 24px; font-family: Arial, sans-serif; color: purple; text-decoration: none; transition: color 0.3s;"><u>Proceed To Order</u></a>

<% } %>

<div class="container">

<%

ResultSet rs = st.executeQuery("select \* from cakes inner join cart on cakes.id=cart.product\_id and cart.email='" + email + "' and cart.address is Null");

while (rs.next()) {

%>

<% sno = sno + 1; %>

<div class="box">

<b>S.no: <%= sno %></b>

<b>Name: <%= rs.getString(2) %></b>

<b>Price(kg): <%= rs.getString(3) %></b>

<b>Quantity: <%= rs.getString(6) %></b>

<b>Sub-Total: <%= rs.getString(8) %></b>

<a href="remove.jsp?id=<%= rs.getString(1) %>">Remove</a>

</div>

<%

}

} catch(Exception e) {

}

%>

</div>

</body>

</html>

**Login.css**

@font-face {

font-family: myFont;

src: url(cakes/Cedarville-Cursive.ttf);

}

h1 {

font-family: myFont;

color: purple;

}

.logo {

height: 100px;

width: 150px;

position: absolute;

top: 0px;

right: 10px;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

.container {

display: flex;

height: 550px;

width: 900px;

box-shadow: 15px 15px 20px black;

margin: auto;

margin-top: 50px;

}

.form {

display: flex;

flex-direction: column;

width: 50%;

align-items: center;

background-color: whitesmoke;

}

.form h2 {

font-size: 3rem;

font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;

margin: 40px;

}

.box {

padding: 12px;

width: 75%;

margin: 15px;

border: 1px solid black;

outline: none;

border-radius: 20px;

}

#submit {

padding: 12px 30px;

width: 60%;

margin-top: 40px;

background-color: black;

color: white;

font-weight: bold;

border: none;

outline: none;

border-radius: 20px;

}

#submit:hover {

cursor: pointer;

background-color: purple;

}

.form a {

text-decoration: none;

color: black;

margin-top: 20px;

}

.form a:hover {

color: #fd4e4e;

}

.image {

display: flex;

justify-content: center;

align-items: center;

width: 50%;

}

.image img {

width: 450px;

height: 450px;

}

**Design.css**

@font-face {

font-family: myFont;

src: url(cakes/Cedarville-Cursive.ttf);

}

@font-face {

font-family: myFont2;

src: url(cakes/EveryDreams-L3Mpy.ttf);

}

h1{

font-family: myFont;

color: purple;

}

footer{

padding:3px;

background-color: black;

color:white;

font-size: 14;

}

.loginn{

position: absolute;

right: 280px;

}

.newuser{

position: absolute;

right: 300px;

bottom: 320px;

font-size: 18px;

}

.button1{

height: 50px;

width: 450px;

background-color:#b28f69 ;

border-radius: 8px;

border-width: 1mm;

font-size: 22px;

}

.logo

{

height: 100px;

width: 150px;

position:absolute;

top: 0px;

right: 10px;

}

.box {

border: none;

box-shadow: 4px 4px 7px rgba(0, 0, 0, 0.8);

width: 700px;

margin: 20px auto;

display: flex;

flex-direction: column;

align-items: center;

padding: 10px;

text-align: center;

}

.button {

margin-left: auto;

height: 250px;

width: 250px;

background-color: yellowgreen;

text-align: center;

}

.container {

display: flex;

flex-wrap: wrap;

}

.box img {

order: -1;

height: 250px;

width: 250px;

}

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: black;

}

li {

float: right;

}

li a {

display: block;

color: white;

text-align: center;

padding: 16px 18px;

text-decoration: none;

}

li a:hover {

background-color:purple;

}

body.showcart.web

{

transform: translateX(-250px);

}

.web

{

transition:transform 0.5s

}

.button{

text-align: center;

background-color: yellowgreen;

height: 40px;

width: 90px;

}

.listcart .quantity span{

display: inline-block;

width: 25px;

height: 25px;

background-color: #eee;

color:#555;

border-radius: 50%;

cursor: pointer;

}

.listcart .quantity span:nth-child(2)

{

background-color: transparent;

color:#eee

}

.icon-cart

{

position: relative;

}

#last-visited {

display: block;

color: white;

text-align: center;

padding: 16px 18px;

text-decoration: none;

}

svg{

width: 30px;

}

.icon-cart span{

display: flex;

width: 20px;

height: 20px;

background-color: red;

justify-content: center;

align-items: center;

color:#fff;

border-radius: 50%;

position: absolute;

}

**Continueshopping.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>CakeCorner</title>

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

</head>

<body>

<h1>CakeCorner</h1>

<img src="cakes/logo.png" class="logo">

<%

String email=session.getAttribute("email").toString();

String status="processing";

try{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

PreparedStatement ps=conn.prepareStatement("update cart set status=? where email=? and status='bill'");

ps.setString(1,status);

ps.setString(2,email);

ps.executeUpdate();

response.sendRedirect("web.jsp");

}catch(Exception e)

{

System.out.println(e);

}

%>

</body>

</html>

**Cart.css**

@font-face {

font-family: myFont;

src: url(cakes/Cedarville-Cursive.ttf);

}

.invoice {

font-family: courier;

}

h1 {

font-family: myFont;

color: purple;

}

.logo {

height: 100px;

width: 150px;

position: absolute;

top: 0;

right: 10px;

}

.box {

width: 310px;

height: 400px;

border: 1px solid #ccc;

border-radius: 5px;

padding: 20px;

margin: 20px;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

text-align: center;

box-shadow: 0px 0px 5px 2px rgba(0, 0, 0, 0.1);

background-color: #fff;

}

.box b {

color: purple;

font-size: 20px;

margin-bottom: 10px;

}

.box p {

font-size: 16px;

line-height: 1.5;

margin-bottom: 10px;

}

.box a {

color: #6a5acd;

text-decoration: none;

font-size: 16px;

}

.box a:hover {

text-decoration: underline;

}

.container {

display: flex;

flex-wrap: wrap;

}

**Cakes.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>CakeCorner</title>

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

</head>

<body class="wallpaper">

<h1>CakeCorner</h1>

<img src="cakes/logo.png" class="logo">

<div class="container">

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

<h2>LOGIN</h2>

<input type="email" id="email" placeholder="Enter Your Email" class="box" name="email" required>

<input type="password" id="password" placeholder="Enter The Password" class="box" name="password" required>

<button type="submit" class="button1" id="submit">Log In</button>

<a href='signup.html'><u>New to CakeCorner...?</u></a>

</form>

<div class="image"><img src="cakes/images.png"></div>

</div><br><br>

<%@page import="java.sql.\*" %>

<%

if ("POST".equalsIgnoreCase(request.getMethod())) {

String email = request.getParameter("email");

String password = request.getParameter("password");

try {

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

PreparedStatement ps = conn.prepareStatement("select \* from login WHERE email= ? AND password = ?");

ps.setString(1, email);

ps.setString(2, password);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

session.setAttribute("email", email);

response.sendRedirect("web.jsp");

} else {

out.println("Invalid username or password");

}

conn.close();

} catch (Exception e) {

out.println("An error occurred: " + e.getMessage());

}

}

%>

</body>

</html>

**Bill.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

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

<title>CakeCorner</title>

</head>

<body>

<%

String email=session.getAttribute("email").toString();

try{

int total=0;

int sno=0;

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

Statement st=conn.createStatement();

ResultSet rs=st.executeQuery("select sum(total) from cart where email='"+email+"' and status='bill'");

while(rs.next())

{

total=rs.getInt(1);

}

ResultSet rs2=st.executeQuery("select \* from login inner join cart on login.email=cart.email where cart.email='"+email+"' and cart.status='bill'");

while(rs2.next())

{

%>

<h1>CakeCorner</h1>

<img src="cakes/logo.png" class="logo">

<hr>

<div style="text-align: center;"><h2 class="invoice">INVOICE</h2></div>

<hr><br><BR>

<div class="container">

<div class="box">

<b>

<h1>From:</h1>

CakeCorner<br>

Jhilmil Colony, Delhi-95

cakecorner@gmail.com <br>

+91-9821998409 <br>

Indulge in happiness, one bite at a time. Welcome to our cake paradise!

</b>

</div>

<div class="box">

<b>

<h1>Invoice To:</h1>

<%=rs2.getString(1) %>,<Br>

<%=email %>,<Br>

<%=rs2.getString(18)%>,<Br>

<%=rs2.getString(15)%>,<Br>

<%=rs2.getString(14)%>,

<%=rs2.getString(16)%>,<Br>

<%=rs2.getString(17)%><Br></b>

</div></div><br><br>

<div class="order-details">

<div class="order-detail">

<span class="heading">Order Date:</span> <%=rs2.getString(19)%>

</div>

<div class="order-detail">

<span class="heading">Payment Method:</span> <%=rs2.getString(21)%>

</div>

</div>

<div class="order-details">

<div class="order-detail">

<span class="heading">Expected Delivery:</span> <%=rs2.getString(20)%>

</div>

<div class="order-detail">

<span class="heading">Transaction Id:</span> <%=rs2.getString(22)%>

</div>

</div>

<hr>

<div style="text-align: center;"><h2 class="invoice">PRODUCT DETAILS</h2></div>

<hr><br>

<table id="bill">

<tr>

<th>S.No</th>

<th>Product Name</th>

<th>price</th>

<th>Quantity</th>

<th>Sub Total</th>

</tr>

<%

ResultSet rs1 = st.executeQuery("SELECT \* FROM cart INNER JOIN cakes ON cart.product\_id = cakes.id WHERE cart.email = '" + email + "' AND cart.status = 'bill'");

while(rs1.next())

{

sno=sno+1;

%>

<tr>

<td><%out.println(sno); %></td>

<td><%=rs1.getString(17) %></td>

<td><%=rs1.getString(18)%></td>

<td><%=rs1.getString(3)%></td>

<td><%=rs1.getString(5)%></td>

</tr>

<% } %>

</table><Br>

<hr><div style="text-align: center;"><h2 class="invoice">Total: <%out.println(total); %></h2></div>

<hr>

<a href="continueshopping.jsp" class="pink-button"><button>Continue Shopping</button></a>

<a onclick="window.print();" class="pink-button"><button>Print</button></a>

<%

}

}catch(Exception e)

{

System.out.println(e);

}

%>

</body>

</html>

**Bill.css**

@charset "UTF-8";

@font-face {

font-family: myFont;

src: url(cakes/Cedarville-Cursive.ttf);

}

.logo {

height: 100px;

width: 150px;

position: absolute;

top: 0px;

right: 10px;

}

.container {

display: flex;

justify-content: space-between;

}

.box {

width: 48%;

height: 250px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 0px 0px 2px 2px rgb(0, 0, 0);

font-family:'Roboto',sans-serif;

padding-left:10px;

}

h1{

font-family: myFont;

color: purple;

}

#bill {

font-family: Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

#bill td, #bill th {

border: 1px solid #ddd;

padding: 8px;

}

#bill tr:nth-child(even){

background-color: #f2f2f2;

}

#bill tr:hover {

background-color: #ddd;

}

#bill th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: purple;

color: white;

}

.invoice{

font-family:courier;

}

.order-details {

display: flex;

margin-bottom: 20px;

}

.order-detail {

flex: 1;

margin-right: 20px;

}

.heading {

font-weight: bold;

color: purple;

font-family: Arial, sans-serif;

}

.detail-value {

color: #333;

}

.pink-button button {

background-color: purple;

color: white;

border: none;

padding: 10px 20px;

border-radius: 5px;

cursor: pointer;

text-decoration: none;

transition: background-color 0.3s ease;

}

hr {

border: none;

border-top: 2px solid black;

}

**Addtocartaction.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>CakeCorner</title>

</head>

<body>

<%

String email =session.getAttribute("email").toString();

String product\_id=request.getParameter("id");

int quantity=1;

int product\_price=0;

int product\_total=0;

int cart\_total=0;

int z=0;

try{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

Statement st=conn.createStatement();

ResultSet rs=st.executeQuery("select \* from cakes where id='"+product\_id+"'");

while(rs.next())

{

product\_price=rs.getInt(3);

product\_total=product\_price;

}

ResultSet rs1=st.executeQuery("select \* from cart where product\_id="+product\_id+" and email='"+email+"' and address is NULL");

while (rs1.next()) {

cart\_total = rs1.getInt(5);

cart\_total = cart\_total + product\_total;

quantity = rs1.getInt(3);

quantity = quantity + 1;

z = 1;

}

if (z == 1) {

st.executeUpdate("update cart set total='" + cart\_total + "',quantity='" + quantity + "' where product\_id='" + product\_id + "' and email='" + email + "' and address is NULL");

response.sendRedirect("web.jsp?msg=exist");

}

if (z == 0) {

PreparedStatement ps = conn.prepareStatement("insert into cart(email,product\_id,quantity,price,total) values(?,?,?,?,?)");

ps.setString(1, email);

ps.setString(2, product\_id);

ps.setInt(3, quantity);

ps.setInt(4, product\_price);

ps.setInt(5, product\_total);

ps.executeUpdate();

response.sendRedirect("web.jsp?msg=added");

}

}

catch(Exception e){

System.out.print(e);

response.sendRedirect("web.jsp?msg=Invalid");

}

%>

</body>

</html>

**Addresspayment.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>CakeCorner</title>

</head>

<body>

<%

String email=session.getAttribute("email").toString();

String address=request.getParameter("address");

String city=request.getParameter("city");

String state=request.getParameter("state");

String country=request.getParameter("country");

String mobilenumber=request.getParameter("phone");

String paymentMethod=request.getParameter("paymentmethod");

String transactionid="";

transactionid=request.getParameter("transactionid");

String status="bill";

try{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cakeonline", "root", "");

PreparedStatement ps=conn.prepareStatement("update login set address=?,city=?,state=?,country=?,mobilenumber=? where email=?");

ps.setString(1,address);

ps.setString(2,city);

ps.setString(3,state);

ps.setString(4,country);

ps.setString(5,mobilenumber);

ps.setString(6,email);

ps.executeUpdate();

PreparedStatement ps1=conn.prepareStatement("update cart set address=?,city=?,state=?,country=?,mobile=?,orderdate=now(),deliverydate=DATE\_ADD(orderDate, INTERVAL 7 DAY),paymentmethod=?,transactionid=?,status=? where email=? and address is NULL");

ps1.setString(1,address);

ps1.setString(2,city);

ps1.setString(3,state);

ps1.setString(4,country);

ps1.setString(5,mobilenumber);

ps1.setString(6,paymentMethod);

ps1.setString(7,transactionid);

ps1.setString(8,status);

ps1.setString(9,email);

ps1.executeUpdate();

response.sendRedirect("bill.jsp");

}catch(Exception e)

{

out.println("Error: " + e.getMessage());

}

%>

</body>

</html>

**TESTING AND VALIDATIONS**

**USER LOGIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test data** | **Expected Output** | **Remarks** |
| 1 | If username is empty | Username | Please fill out this field | SUCCESSFUL |
| 2 | If password is empty | Password | Please fill out this field | SUCCESSFUL |
| 3 | If entered username and password is not valid | Username,password | Invalid username or password | SUCCESSFUL |

**SIGNIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test data** | **Expected Output** | **Remarks** |
| 1 | If email is empty | email | Please fill out this field | SUCCESSFUL |
| 2 | if password is empty | password | Please fill out this field | SUCCESSFUL |
| 3 | if mobileNo. is empty | mobileNo. | Please fill out this field | SUCCESSFUL |
| 4 | If name is empty | name | Please fill out this field | SUCCESSFUL |
| 5 | if number is less than 10 and user is typing alphabets | MobileNo. | Please enter a 10-digit phone number | SUCCESSFUL |
| 6 | if user is typing email without @ and . | email | Please enter email address | SUCCESSFUL |

**Payment Information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test data** | **Expected data** | **Remarks** |
| 1 | if address is empty | Address | Please fill out this field | SUCCESSFUL |
| 2 | if city is empty | City | Please fill out this field | SUCCESSFUL |
| 3 | if state is empty | State | Please fill out this field | SUCCESSFUL |
| 4 | if country is empty | Country | Please fill out this field | SUCCESSFUL |
| 5 | if mobile number is empty | Mobile number | Please fill out this field | SUCCESSFUL |
| 6 | if number is less than 10 and user is typing alphabets | Mobile number | Please enter a 10-digit phone number | SUCCESSFUL |

**INPUT AND OUTPUT SCREEN**

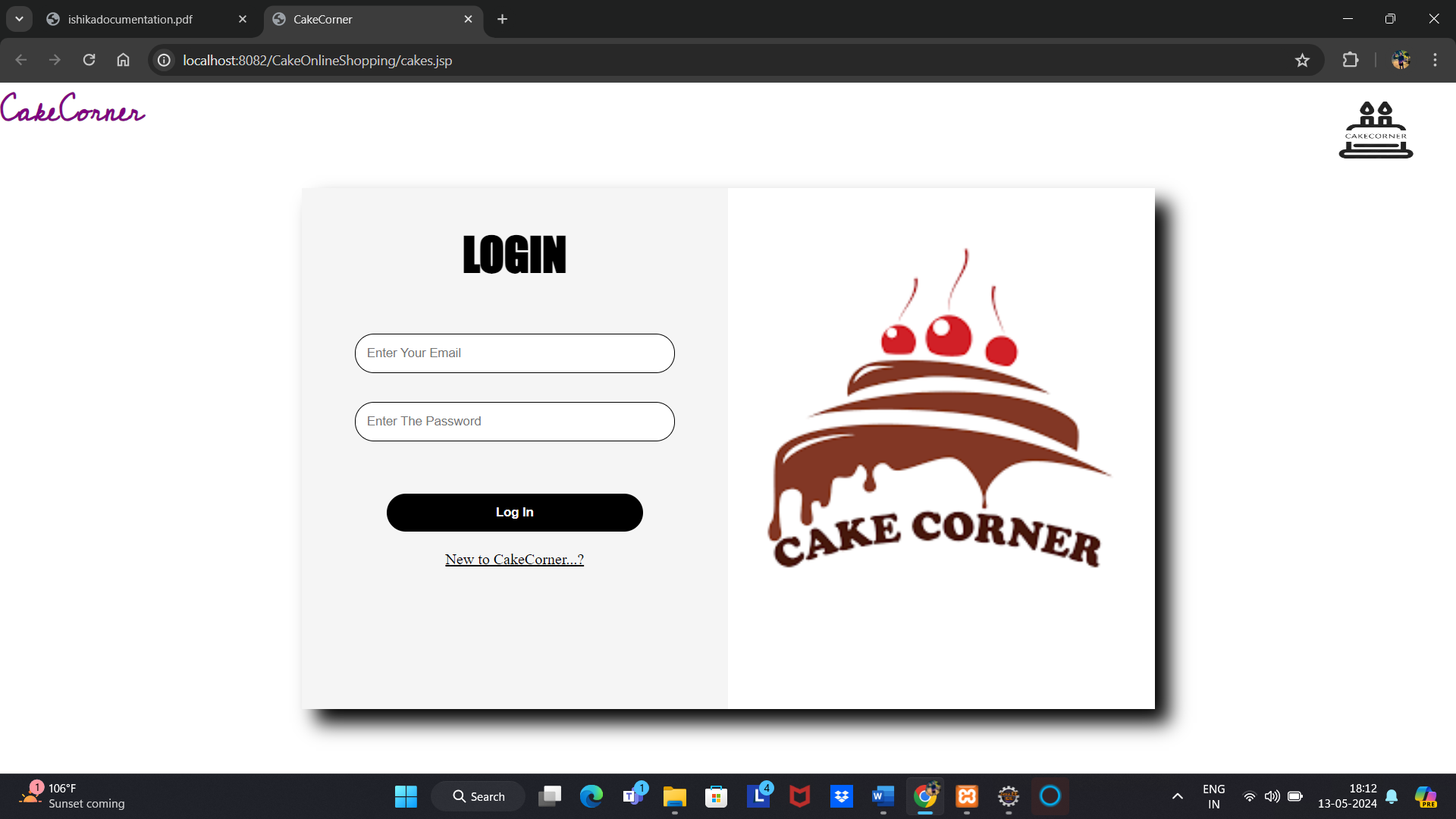


Fig 8.1 Login Page

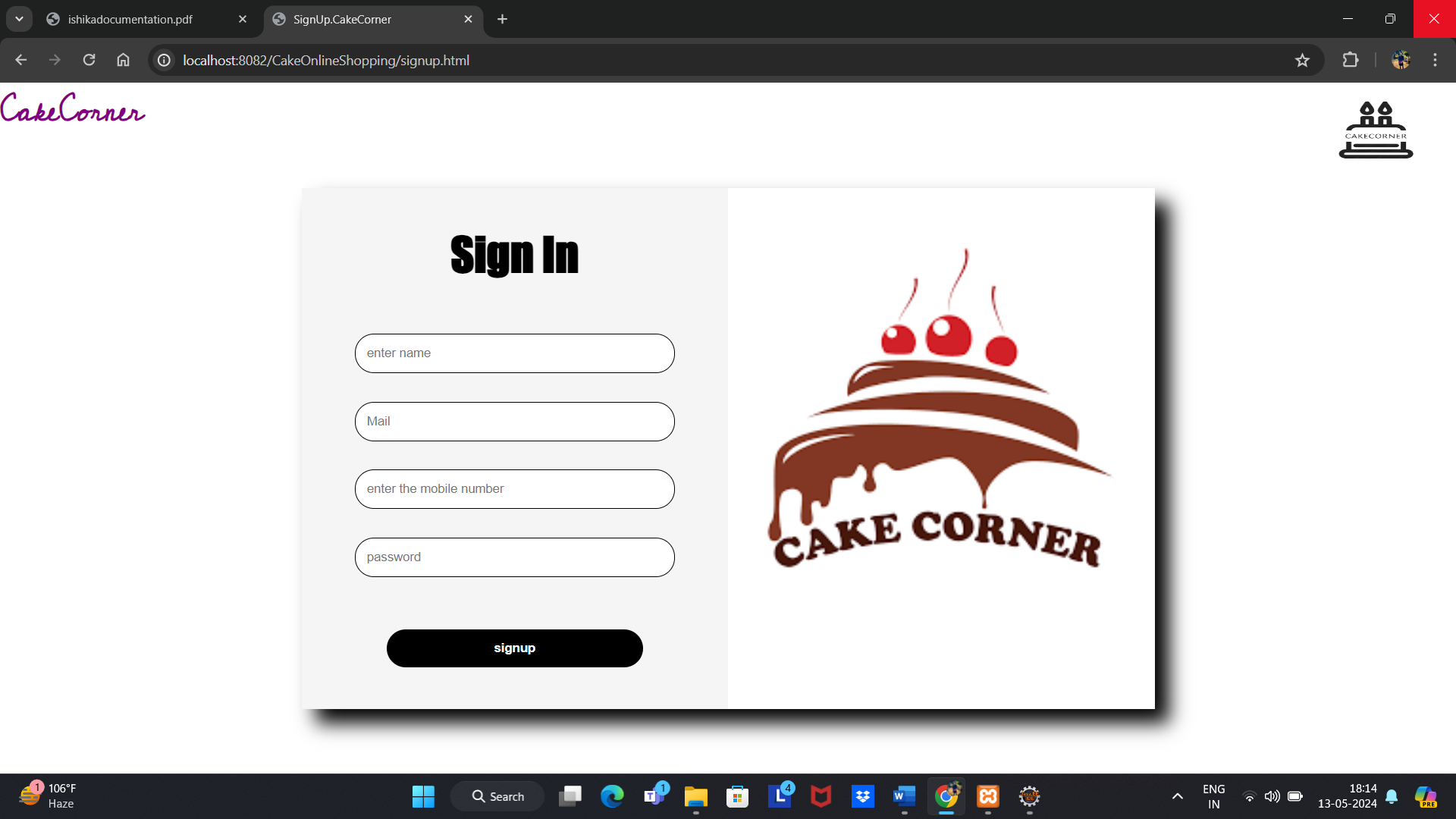


Fig 8.2 SignIn page

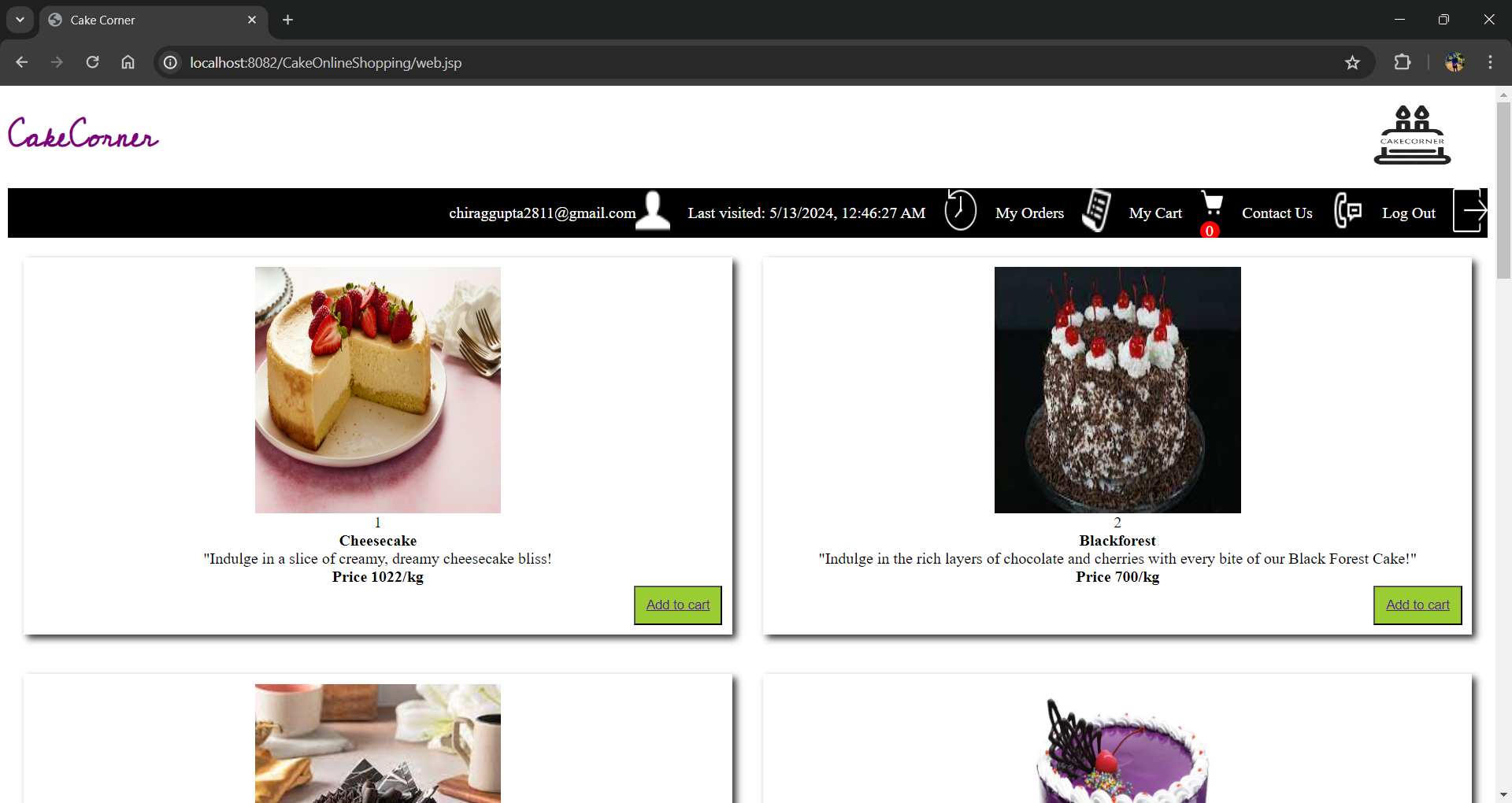


Fig 8.3 Home Page

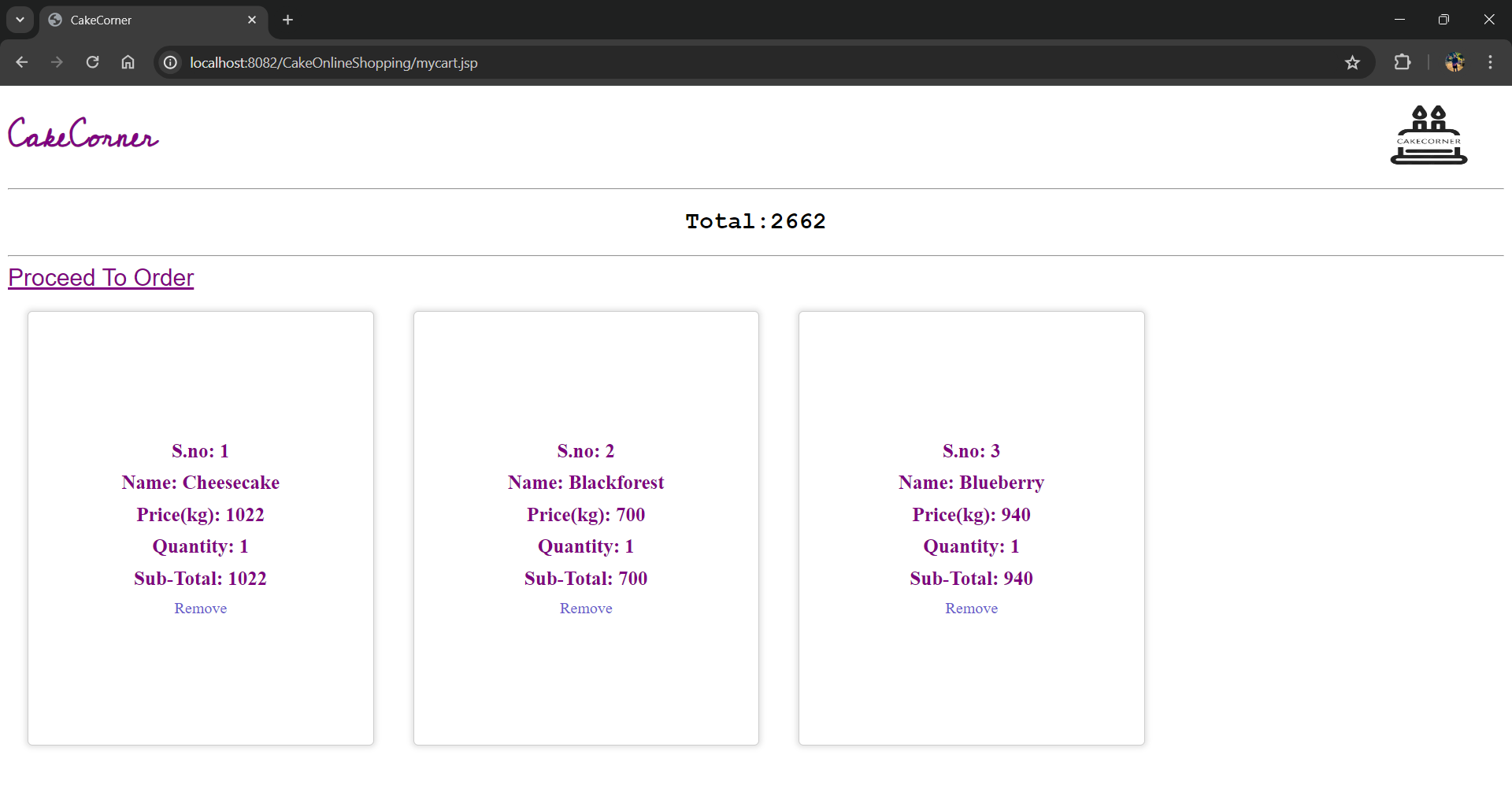


Fig 8.4 Cart Page

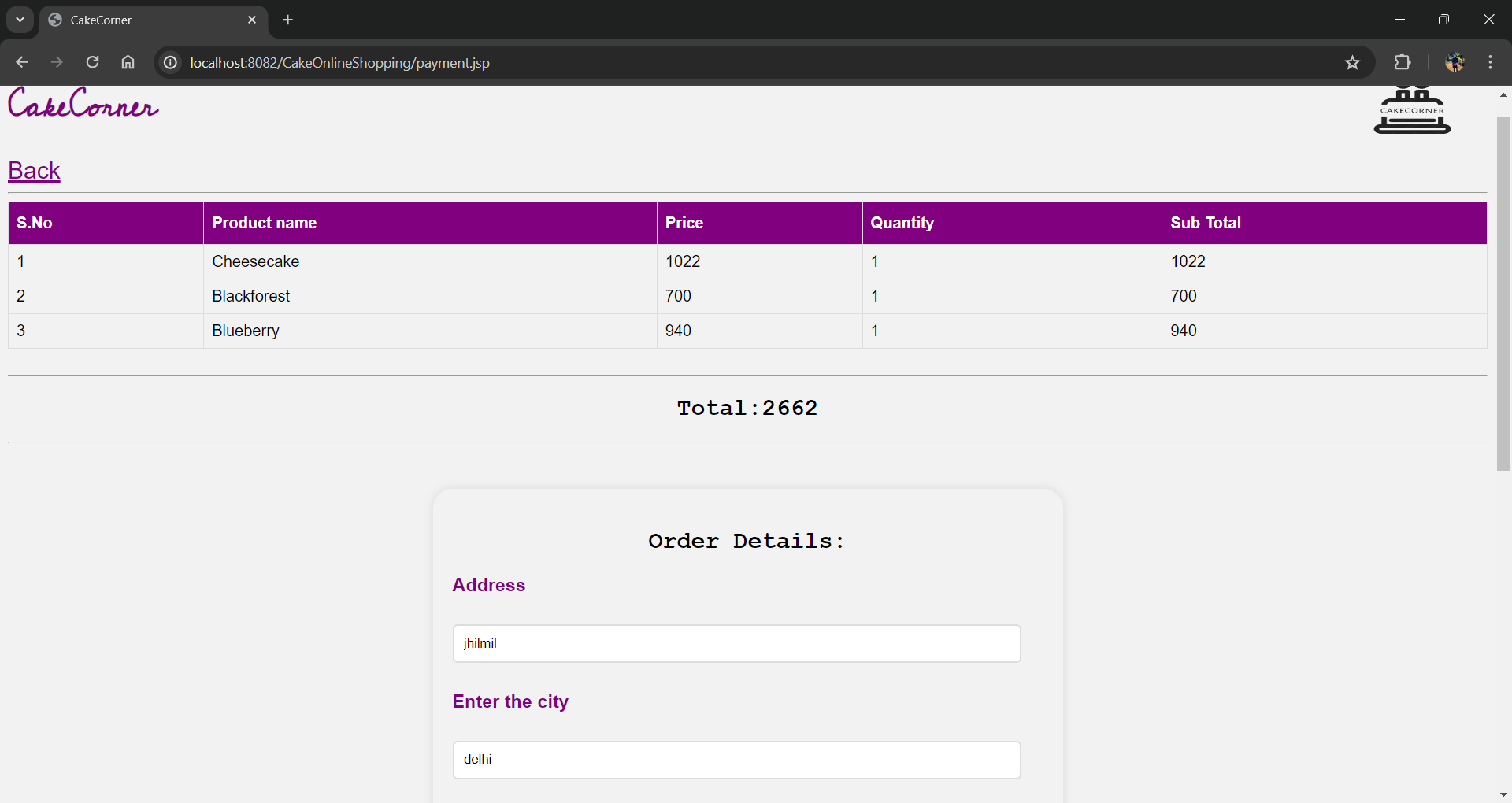


Fig 8.5 Information page

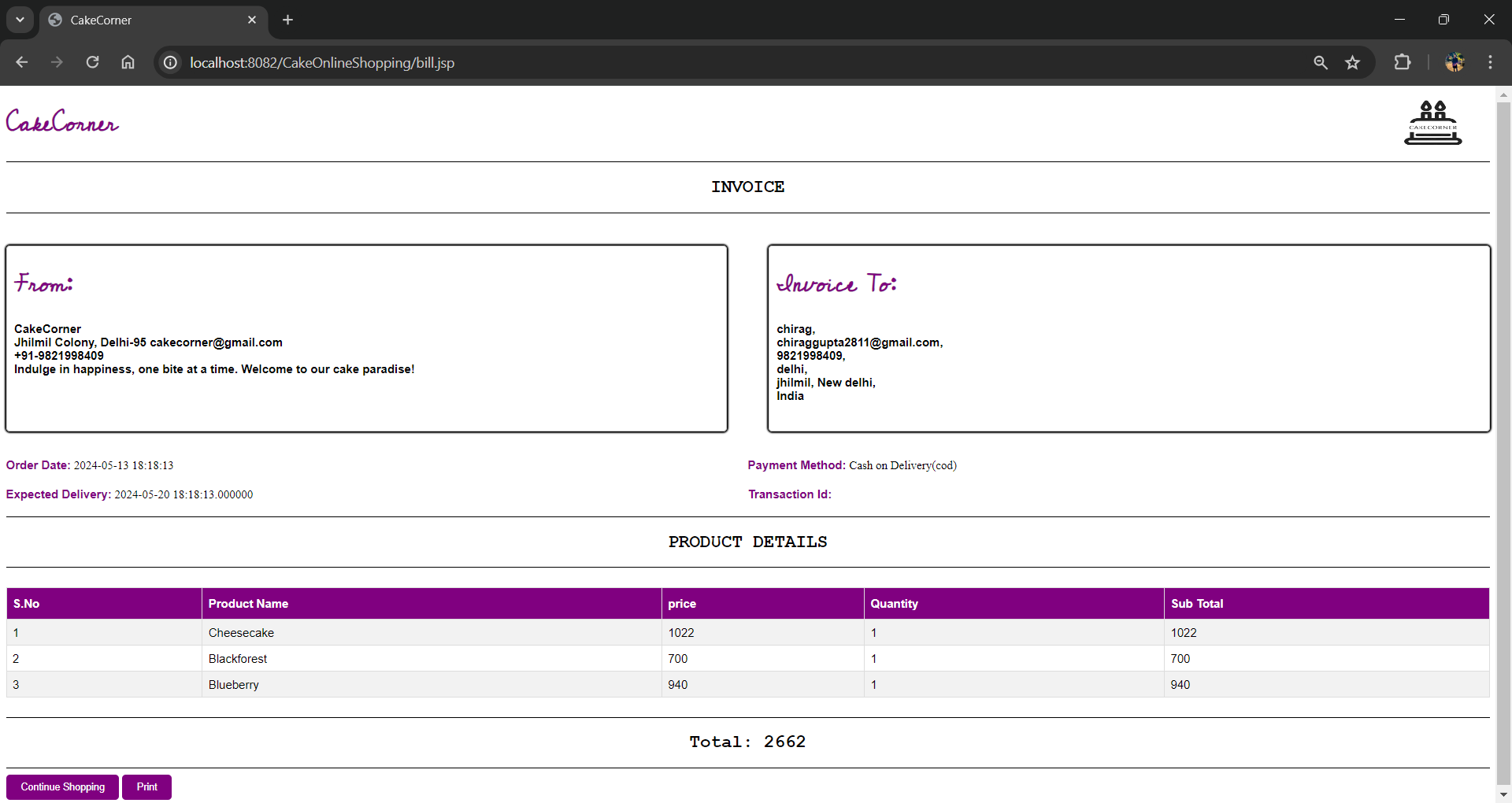


Fig 8.6 Bill Page

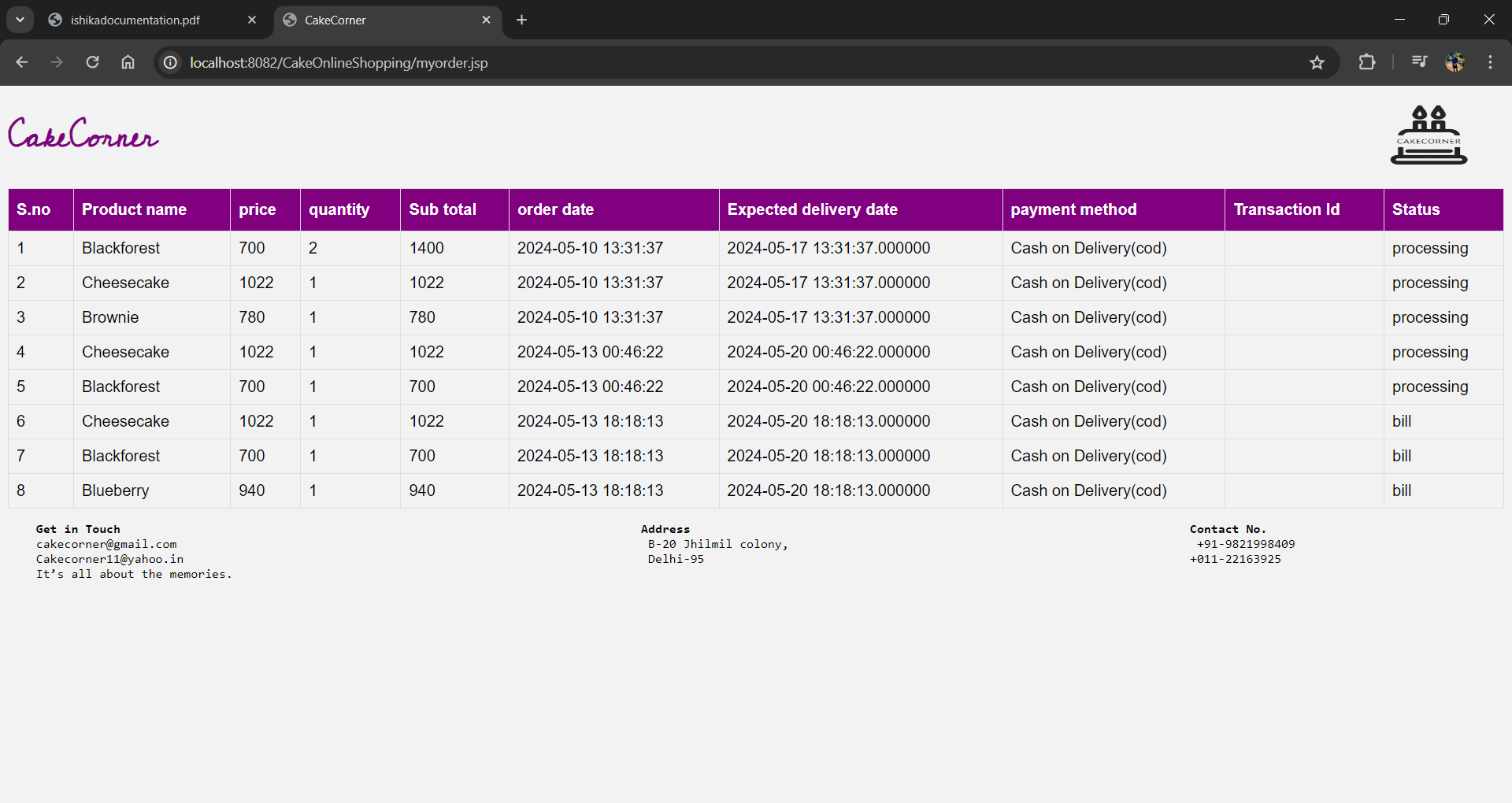


Fig 8.7 Myorders

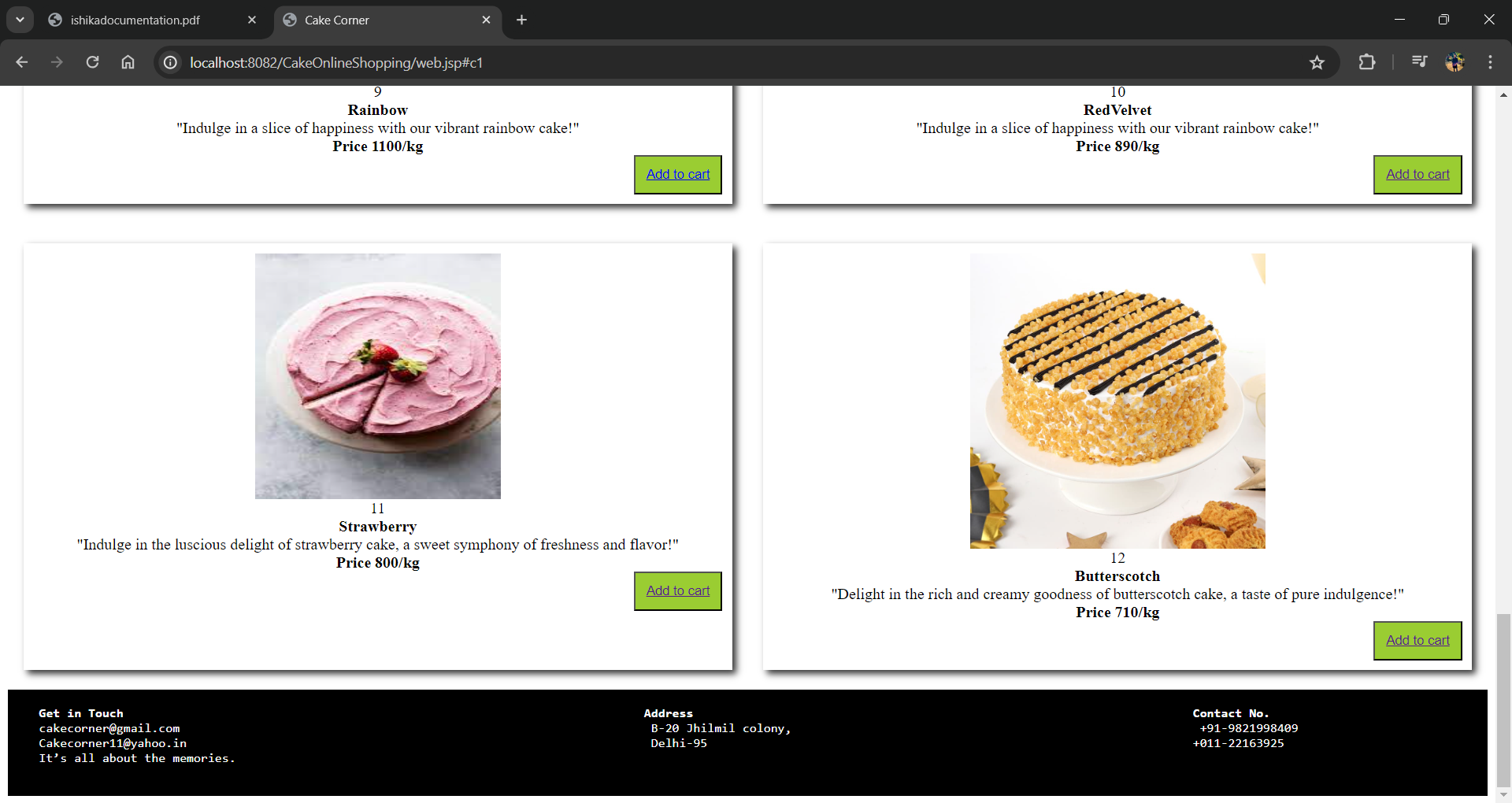


Fig 8.8 Contact Us

**LIMITATIONS OF THE PROJECT**

1. **Performance:** Java and JSP can be resource-intensive, especially for larger applications. Ensuring optimal performance, especially during high traffic times, might require careful optimization and possibly additional server resources.
2. **Scalability:** As your online cake shop grows, you may encounter scalability issues with your Java-based solution. Ensuring that your architecture can scale horizontally to handle increased load might require additional planning and possibly re-architecting parts of your system.
3. **Learning Curve:** Java and JSP might have a steeper learning curve compared to other technologies. If you or your team are not already proficient in these technologies, it could take some time to ramp up, which might delay the project or introduce more errors initially.
4. **Frontend Flexibility**: JSP, while powerful for server-side rendering, might not offer the same level of flexibility and interactivity as modern frontend frameworks like React or Angular. This could limit the user experience you can provide to your customers.
5. **Browser Compatibility**: Ensuring cross-browser compatibility can sometimes be challenging with JSP, especially when dealing with older versions of Internet Explorer or other less standards-compliant browsers.
6. **Security:** Java and JSP are relatively secure technologies, but as with any web application, you need to be vigilant about security vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
7. **Maintenance:** Java and JSP projects can sometimes require more maintenance compared to projects built with newer technologies. Keeping your codebase up-to-date with the latest security patches and libraries might require ongoing effort.
8. **Limited Ecosystem:** While Java has a rich ecosystem of libraries and frameworks, the ecosystem for JSP specifically might be more limited compared to other frontend technologies. This could make it harder to find ready-made solutions for certain problems.

**FUTURE APPLICATION OF THE PROJECT**

1. **Mobile Application Integration:** You can extend your online cake shopping platform by developing mobile applications for iOS and Android platforms. These apps can provide a more seamless and convenient shopping experience for users on their smartphones and tablets.
2. **Enhanced User Experience:** Implementing features such as personalized recommendations based on past purchases, advanced search capabilities, and interactive product visualization can enhance the user experience and drive customer engagement.
3. **Internationalization and Localization:** Expanding your platform to support multiple languages, currencies, and regional preferences can help you tap into new markets and reach a more diverse customer base.
4. **Social Media Integration:** Integrating social media features such as sharing cake designs, customer reviews, and social login options can increase brand visibility, foster community engagement, and drive organic growth through word-of-mouth marketing.
5. **Subscription Services:** Offering subscription-based services for cake deliveries, monthly cake boxes, or special occasion packages can create recurring revenue streams and foster customer loyalty.
6. **Analytics and Business Intelligence:** Implementing advanced analytics and business intelligence tools can provide valuable insights into customer behaviour, sales trends, and inventory management, enabling you to make data-driven decisions and optimize your business operations.
7. **Collaborations with Local Bakeries:** Partnering with local bakeries or independent cake designers to offer a wider range of cake options, exclusive collaborations, and limited-edition releases can differentiate your platform and attract new customers.

**BIBLOGRAPHY**

During this project, I have referred a numerous website from which some are given below :-

1. <https://www.google.com/>
2. <https://www.flaticon.com/free-icons/cart>
3. <https://www.youtube.com/>
4. <https://images.google.com/>
5. <https://www.w3schools.com/java/>
6. <https://www.tutorialspoint.com/jsp/index.html>
7. <http://localhost/phpmyadmin/>