

Name: Shruti
Prasad Saigal

T23051

Java EE CA 2 

Software Development Life Cycle (SDLC) for *Sweet Delights Bakery Web Application*

1. Requirement Analysis

The main goal of the project was to build a bakery management web application for **small bakery businesses and budding entrepreneurs**. These businesses usually lack the budget for complex e-commerce platforms but still need a professional digital presence. The application helps them showcase products, manage orders, and interact with customers more efficiently.

Functional Requirements

- **User Management:** User registration (Sign Up), Login, and Logout.
- **Product Management:** Categories like cakes, bread, cookies, pastries. Admins can add/edit/delete products.
- **Cart Management:** Customers can add items to cart, update quantity, and remove items.
- **Checkout & Payment:** Customers can place orders and choose from multiple payment modes: *Cash on Delivery (COD), UPI, Card*.
- **Order History:** Customers can view their past orders.
- **Admin Access:** Bakery owners/admins can view and manage products, categories, and orders.

Non-Functional Requirements

- Simple, **responsive UI** that works on both desktop and mobile devices.
- **Secure login system** with session management.
- Smooth navigation and optimized performance for a better customer experience.
- Compatibility with widely used environments like **XAMPP (Apache + MySQL)** for local deployment.

- Scalability for future business growth.
-

2. Planning

This project was designed with the idea of helping **budding small-scale businesses** go digital without requiring costly systems. By using open-source technologies, the development and maintenance cost is minimized.

Scope

The system allows bakery owners to:

- Upload and manage their products.
- Provide customers a way to browse, order, and pay.
- Track sales and customer orders through a single platform.

Technology Stack

- **Frontend:** JSP, HTML, CSS, JavaScript
 - **Backend:** Java (Servlets, JSP), JDBC
 - **Database:** MySQL via XAMPP
 - **Server:** Apache Tomcat
 - **Build Tool:** Maven
-

3. System Design

Architecture

The system follows the **MVC (Model–View–Controller)** architecture:

- **Model:** Java classes, database connectivity, and SQL queries.
- **View:** JSP pages (index.jsp, products.jsp, checkout.jsp, etc.).

Database Design

- **Users:** Stores customer/admin info.
- **Categories:** Stores product categories.
- **Products:** Product details with pricing.
- **Orders:** Order details (amount, status, payment method).
- **Order_Items:** Mapping of products to orders.
- **Order_Shipping:** Shipping details like name, address, and contact.

UI Layout

- Homepage with hero banner and categories.
- Product listing page (products.jsp).
- Cart page (cart.jsp).
- Checkout page (checkout.jsp) with multiple payment options.
- Order confirmation page (order_success.jsp).
- Order history page (order_history.jsp).

4. Implementation

Frontend JSPs

- index.jsp – Homepage with categories.
- products.jsp – Displays bakery products.
- cart.jsp – Shows cart items.
- checkout.jsp – Checkout form + payment selection.
- order_success.jsp – Order confirmation.
- order_history.jsp – Displays user's past orders.
- contact_queries.jsp- Lets the user add their query.
- login.jsp-Allows the user to login after signup.
- signup.jsp- Allows the user to signup.
- Order_sucess.jsp- Displays Order Successful on the screen.

Backend Java Files

- **DBConnection.java** – Connects to MySQL using XAMPP.

CSS & UI Design

- Theme inspired by a bakery: pastel pink + golden tones.
- Buttons, banners, and cards designed for **user-friendliness**.
- Responsive for both desktop and mobile.

5. Testing

Unit Testing

- Verified database connection with **XAMPP MySQL**.
- Tested login/signup using test accounts.

- Checked add-to-cart, update, and delete functionality.

Integration Testing

- Full flow testing: Login → Add to Cart → Checkout → Select Payment Method → Order Success → Order History.
- Verified order and payment data are correctly saved in MySQL.

User Acceptance Testing (UAT)

- Feedback ensured the system was easy to use for **non-technical users**.
-

6. Deployment

Development Environment

- Local machine with **XAMPP (Apache + MySQL)** for database.
- **Tomcat Server** for hosting the JSP/Servlet application.

Deployment Steps

1. Export project as .war file.
 2. Deploy .war file on Apache Tomcat server.
 3. Import MySQL schema using **phpMyAdmin** (provided by XAMPP).
 4. Application accessible via <http://localhost:8080/BakeryApp/>.
-

7. Maintenance

Ongoing Support

- Fixing bugs (e.g., database column mismatches, null values).
- Optimizing queries for speed.
- Updating product categories and UI as required.

Future Enhancements

- Integration with online payment gateways (Razorpay, PayPal).
- Order delivery tracking.
- Discount codes and seasonal offers.
- Multi-language & multi-currency support to expand the customer base.

Process Model Used: Waterfall Model

1. Overview

The **Waterfall model** is a linear-sequential software development methodology, where each phase must be fully completed before moving to the next. This model is ideal for projects with **clearly defined requirements**, such as the *Sweet Delights Bakery Web Application*.

Since the functionalities — product catalog, cart, checkout, payment, and order history — were well understood from the beginning, the Waterfall approach ensured smooth and systematic development.

2. Phases Applied to Sweet Delights Bakery

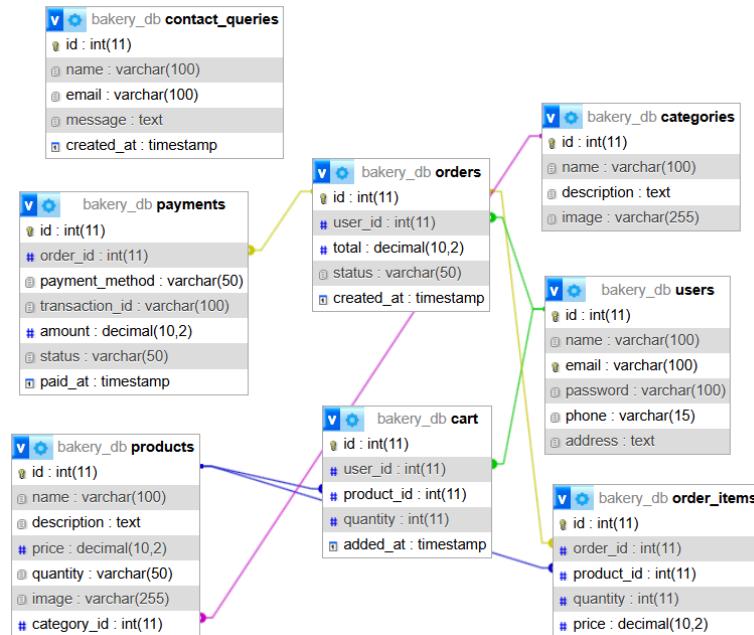
Phase	Description & Implementation in Sweet Delights Bakery
-------	---

Requirement Analysis	Requirements were gathered by analyzing how small bakery businesses operate and identifying their need for an affordable e-commerce solution. Functional requirements such as user management, product catalog, cart, checkout with multiple payment modes, and order history were defined. Non-functional requirements included simplicity, responsiveness, and database connectivity through XAMPP MySQL . Deliverable: <i>Requirement Specification Document (RSD)</i> .
System Design	Database schema was designed with tables such as users, categories, products, orders, order_items, and payments . Relationships were defined using foreign keys and auto-increment IDs. UI wireframes were created for homepage, product listings, cart, checkout, and order success pages . Deliverables: <i>Database schema, ER Diagram, UI Wireframes</i> .
Implementation (Coding)	JSP pages were developed for the frontend (index.jsp, products.jsp, cart.jsp, checkout.jsp, order_success.jsp, order_history.jsp). Java Servlets handled the backend (DBConnection.java, CartServlet.java, CheckoutServlet.java, AuthServlet.java). Cart logic was implemented using session management . Payments were simulated via three modes: <i>Cash on Delivery (COD), UPI, and Card</i> . Styling was done using pastel pink and golden bakery-themed CSS. Deliverable: <i>Fully functional Bakery Application Code</i> .
Testing	<ul style="list-style-type: none"> - Unit Testing: Verified DB connections with XAMPP, login/signup validation, add-to-cart operations. - Integration Testing: Ensured end-to-end flow worked (Signup/Login → Add to Cart → Checkout → Payment → Order Success → Order History). - Acceptance Testing: Confirmed system usability with small bakery users and students. Deliverables: <i>Test cases, screenshots of successful flows</i> .
Deployment	Deployed locally using XAMPP (Apache + MySQL) and Apache Tomcat for JSP/Servlet execution. The application was exported as a WAR file and tested in Tomcat. Database schema was imported via phpMyAdmin (XAMPP). Deliverables: <i>Deployment guide, WAR file</i> .
Maintenance	Planned updates include bug fixing, security improvements (password hashing, SSL), and performance optimization . Future upgrades may add payment gateway integration (Razorpay/PayPal), delivery tracking, discount codes, and multilingual support . Deliverable: <i>Maintenance and Enhancement Plan</i> .

3. Why Waterfall Was Suitable

- **Clear Requirements:** The features (cart, checkout, payment, order history) were clearly defined from the start.
- **Sequential Development:** Each JSP/Servlet depended on the database and session structures, which were designed first.
- **Ease of Management:** Linear structure made it easy to track progress and milestones phase by phase.
- **Strong Documentation:** Each phase produced outputs like the **RSD**, **ER diagram**, **UI wireframes**, and **test cases**, ensuring proper project documentation.

Relation between tables of Database bakery_db



Code

1.index.jsp

```
<%@ page import="java.sql.*" %>
<%@ page import="com.bakery.util.DBConnection" %>
<%
    Connection conn = DBConnection.getConnection();
```

```

Integer userId = (Integer) session.getAttribute("user_id");
String userName = (String) session.getAttribute("name");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Sweet Delights Bakery</title>
<link href="https://fonts.googleapis.com/css2?family=Georgia&display=swap" rel="stylesheet">
<style>
body { font-family: 'Georgia', serif; margin:0; padding:0; background:#ffe6f0; color:#b8860b; }
a { text-decoration:none; transition:0.3s; }
a:hover { color:#ff69b4; }

/* Header */
header { background:#fff; color:#b8860b; padding:20px 40px; text-align:center; box-shadow:0 4px 10px rgba(0,0,0,0.1); }
header h1 { margin:0; font-size:2.8em; }

/* Navigation */
nav { text-align:right; padding:10px 40px; background:#ffe6f0; }
nav a { margin-left:15px; font-weight:bold; color:#b8860b; }

/* Hero Section */
.hero { position:relative;
background:url('https://www.ukiosks.com/wp-content/uploads/2023/11/9-bakery-shop-2.jpg')
no-repeat center center/cover; height:450px; display:flex; justify-content:center; align-items:center;
flex-direction:column; text-align:center; font-size:2.2em; font-weight:bold; color:#fff; }
.hero::before { content:""; position:absolute; top:0; left:0; right:0; bottom:0;
background:rgba(0,0,0,0.5); z-index:0; }
.hero * { position:relative; z-index:1; }
.hero button { margin-top:20px; padding:12px 25px; background:#b8860b; color:#fff; border:none;
border-radius:10px; cursor:pointer; font-size:0.6em; font-weight:bold; transition:0.3s; }
.hero button:hover { background:#ff69b4; }

/* About Section */
.about { display:flex; flex-wrap:wrap; justify-content:center; align-items:center; padding:60px 20px;
background:#fff; margin:20px; border-radius:20px; box-shadow:0 8px 20px rgba(0,0,0,0.05); }
.about img { width:500px; max-width:100%; border-radius:15px; margin:20px; }
.about div { max-width:500px; margin:20px; font-size:1.2em; line-height:1.8; }

/* Categories Section */

```

```

.categories-container { display:grid; grid-template-columns:repeat(auto-fit,minmax(250px,1fr));
gap:35px; padding:50px 40px; }
.category { background:#fff; border-radius:20px; overflow:hidden; text-align:center; box-shadow:0
8px 20px rgba(0,0,0,0.1); transition: transform 0.3s, box-shadow 0.3s; cursor:pointer; }
.category:hover { transform:translateY(-8px); box-shadow:0 15px 25px rgba(0,0,0,0.2); }
.category img { width:100%; height:250px; object-fit:cover; transition: transform 0.4s; }
.category img:hover { transform:scale(1.05); }
.category a { display:block; padding:18px; color:#b8860b; font-weight:700; font-size:1.1em; }
.category a:hover { color:#ff69b4; letter-spacing:1px; }

/* Footer */
footer { background:#fff; color:#b8860b; text-align:center; padding:30px 20px; margin-top:50px;
box-shadow:0 -4px 10px rgba(0,0,0,0.05); }
footer a { color:#b8860b; font-weight:bold; }
footer a:hover { color:#ff69b4; }

@media(max-width:768px){ .hero{font-size:1.8em;height:350px;}}
.categories-container{padding:30px 20px; gap:25px;} .about{flex-direction:column;text-align:center; }
@media(max-width:480px){ .hero{font-size:1.5em;height:250px;}.about{padding:30px
15px;font-size:1em;margin:15px;} .categories-container{padding:20px 15px; gap:20px;}}
nav{text-align:center;} }

</style>
</head>
<body>

<header>
  <h1>Sweet Delights Bakery</h1>
</header>

<nav>
<%
  if(userId != null){
%>
    Welcome, <%= userName %> |
    <a href="checkout.jsp">Cart</a> |
    <a href="order_history.jsp">My Orders</a> |
    <a href="logout.jsp">Logout</a>
    <a href="contact_queries.jsp">Contact Us</a>
<%
  } else {
%>
    <a href="login.jsp">Login</a> |

```

```

<a href="signup.jsp">Sign Up</a>
<%
    }
%>
</nav>

<div class="hero">
    Freshly Baked Happiness Every Day!
    <button onclick="window.location.href='products.jsp'">Shop Now</button>
</div>

<div class="about">
    
    <div>
        <h2>About Sweet Delights</h2>
        <p>We bake with love and passion, creating cakes, pastries, bread, and cookies with the freshest ingredients.</p>
        <p>Every product is handcrafted to ensure a sweet experience for you and your loved ones.</p>
    </div>
</div>

<h2 style="text-align:center; margin: 60px 0 20px;">Explore Our Categories</h2>
<div class="categories-container">
<%
    if(conn != null){
        try (Statement st = conn.createStatement(); ResultSet rs = st.executeQuery("SELECT * FROM categories")){
            while(rs.next()){
                int catId = rs.getInt("id");
                String catName = rs.getString("name");

                // Map category names to image files
                String imageName = "";
                switch(catName.toLowerCase()){
                    case "cakes": imageName="cakes.jpg"; break;
                    case "bread": imageName="bread.jpg"; break;
                    case "pastries": imageName="pastries.jpg"; break;
                    case "cookies": imageName="cookies.jpg"; break;
                    default: imageName="default.jpg"; // fallback image
                }
            }
        }
    }
%>
<div class="category">
```

```

/images/categories/<%=imageName%>">
alt="<%="catName%>">
<a href="products.jsp?categoryId=<%=catId%>"><%=catName %></a>
</div>
<%
}
} catch(Exception e){
    out.println("<p style='color:red;'>Error loading categories: "+e.getMessage()+"</p>");
}
} else {
    out.println("<p style='color:red;'>Database connection not available!</p>");
}
%>
</div>

<footer>
    <p>Contact us: <a href="mailto:info@sweetdelights.com">info@sweetdelights.com</a> | Phone:
+91-9876543210</p>
    <p>Address: 123, Bakery Street, Mumbai, India</p>
    <p>&copy; 2025 Sweet Delights Bakery. All rights reserved.</p>
</footer>

</body>
</html>

```

2. login.jsp

```

<%@ page import="java.sql.*" %>
<%@ page import="com.bakery.util.DBConnection" %>
<%
Connection conn = DBConnection.getConnection();
if(session.getAttribute("user_id") != null){
    response.sendRedirect("index.jsp");
}
%>
<!DOCTYPE html>
<html>
<head>
    <title>Login - Bakery</title>
    <link href="https://fonts.googleapis.com/css2?family=Georgia:wght@400;700&display=swap"
rel="stylesheet">
    <style>
        body {

```

```

font-family: 'Georgia', serif;
margin: 0; padding: 0;
background: #ffe6f0; /* baby pink */
color: #b8860b; /* gold text */
display: flex; justify-content: center; align-items: center; height: 100vh;
}

.login-container {
background: #ffffff;
padding: 40px 30px;
border-radius: 15px;
box-shadow: 0 8px 20px rgba(0,0,0,0.1);
width: 350px;
text-align: center;
}

h2 { color: #b8860b; margin-bottom: 20px; }

input[type=email], input[type=password] {
width: 100%; padding: 12px; margin: 10px 0 20px 0; border-radius: 8px; border: 1px solid #ddd;
}

input[type=submit] {
width: 100%; background: #b8860b; color: white; border: none; padding: 12px; border-radius: 8px;
font-size: 16px; cursor: pointer; transition: background 0.3s;
}

input[type=submit]:hover { background: #a17405; }

.error { color: red; margin-bottom: 15px; }

.success { color: green; margin-bottom: 15px; }

a { text-decoration: none; color: #b8860b; }

a:hover { color: #ff69b4; }

</style>

<script>

function validateForm(){
var email=document.forms["loginForm"]["email"].value;
var pass=document.forms["loginForm"]["password"].value;
if(email=="" || pass==""){ alert("Please fill all fields"); return false; }
return true;
}

</script>

</head>

<body>

<div class="login-container">

<h2>Login</h2>

<form name="loginForm" method="post" onsubmit="return validateForm()">

```

```

Email: <input type="email" name="email" required><br>
Password: <input type="password" name="password" required><br>
<input type="submit" value="Login">
</form>
<%
if(request.getMethod().equalsIgnoreCase("POST") && conn != null){
    String email=request.getParameter("email");
    String password=request.getParameter("password");
    try {
        String sql="SELECT * FROM users WHERE email=? AND password=?";
        PreparedStatement ps=conn.prepareStatement(sql);
        ps.setString(1,email);
        ps.setString(2,password);
        ResultSet rs=ps.executeQuery();
        if(rs.next()){
            int userId = rs.getInt("id"); // <-- changed from "user_id" to "id"
            session.setAttribute("user_id", userId);
            session.setAttribute("name", rs.getString("name"));
        }
    } catch(Exception e){ out.println("<p class='error'>Error: "+e.getMessage()+"</p>"); }
}
%>
<script>
    window.location.href='index.jsp';
</script>
<%
} else {
    out.println("<p class='error'>Invalid email or password!</p>");
}
rs.close(); ps.close();
} catch(Exception e){ out.println("<p class='error'>Error: "+e.getMessage()+"</p>"); }
%>

<p>Don't have an account? <a href="signup.jsp">Sign Up</a></p>
</div>
</body>
</html>

```

3.signup.jsp

```

<%@ page import="java.sql.*" %>
<%@ page import="com.bakery.util.DBConnection" %>
<%
// Get database connection
Connection conn = DBConnection.getConnection();

```

```
// Redirect if already logged in
if(session.getAttribute("user_id") != null){
    response.sendRedirect("index.jsp");
    return;
}
%>
<!DOCTYPE html>
<html>
<head>
    <title>Sign Up - Bakery</title>
    <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">
    <style>
        body {
            font-family: 'Georgia', serif;
            margin: 0;
            padding: 0;
            background: #ffe6f0; /* baby pink */
            color: #b8860b; /* gold */
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .signup-container {
            background: #ffffff;
            padding: 40px 35px;
            border-radius: 15px;
            box-shadow: 0 10px 25px rgba(0,0,0,0.15);
            width: 360px;
        }
        h2 {
            text-align: center;
            color: #b8860b;
            margin-bottom: 25px;
        }
        input[type=text], input[type=email], input[type=password] {
            width: 100%;
            padding: 12px;
            margin: 10px 0 20px 0;
            border-radius: 8px;
        }
    </style>

```

```
border: 1px solid #ddd;
font-size: 1em;
}
input[type=submit] {
width: 100%;
background: #b8860b;
color: white;
padding: 12px;
border-radius: 8px;
border: none;
font-size: 1.1em;
cursor: pointer;
transition: background 0.3s ease;
}
input[type=submit]:hover {
background: #a17405;
}
.error { color: red; text-align: center; }
.success { color: green; text-align: center; }
</style>
<script>
function validateForm(){
var name=document.forms["signupForm"]["name"].value.trim();
var email=document.forms["signupForm"]["email"].value.trim();
var pass=document.forms["signupForm"]["password"].value.trim();
if(name=="" || email=="" || pass==""){
alert("Please fill all fields");
return false;
}
return true;
}
</script>
</head>
<body>
<div class="signup-container">
<h2>Sign Up</h2>
<form name="signupForm" method="post" onsubmit="return validateForm()">
Name: <input type="text" name="name" required><br>
Email: <input type="email" name="email" required><br>
Password: <input type="password" name="password" required><br>
<input type="submit" value="Sign Up">
</form>
```

```

<%
if(request.getMethod().equalsIgnoreCase("POST") && conn != null){
    String name=request.getParameter("name");
    String email=request.getParameter("email");
    String password=request.getParameter("password");
    try {
        String check="SELECT * FROM users WHERE email=?";
        PreparedStatement ps1=conn.prepareStatement(check);
        ps1.setString(1,email);
        ResultSet rs1=ps1.executeQuery();
        if(rs1.next()){
            out.println("<p class='error'>Email already registered!</p>");
        } else {
            String sql="INSERT INTO users(name,email,password) VALUES(?, ?, ?)";
            PreparedStatement ps=conn.prepareStatement(sql);
            ps.setString(1,name); ps.setString(2,email); ps.setString(3,password);
            ps.executeUpdate();
            out.println("<p class='success'>Signup successful! Redirecting to login...</p>");
        }
    } catch(Exception e){ out.println("<p class='error'>Error: "+e.getMessage()+"</p>"); }
} else if(conn == null){
    out.println("<p class='error'>Database connection not available!</p>");
}
%>
</div>
</body>
</html>

```

4.products.jsp

```

<%@ page import="java.sql.* , java.util.*" %>
<%@ page import="com.bakery.util.DBConnection" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

```

```
<%
```

```
Connection conn = DBConnection.getConnection();
Integer userId = (Integer) session.getAttribute("user_id");
String categoryIdParam = request.getParameter("categoryId");
%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Products - Sweet Delights Bakery</title>
    <link href="https://fonts.googleapis.com/css2?family=Georgia&display=swap" rel="stylesheet">
    <style>
        body { font-family:'Georgia', serif; margin:0; padding:0; background:#ffe6f0; color:#b8860b; }
        a { text-decoration:none; transition:0.3s; }
        a:hover { color:#ff69b4; }

        header { background:#fff; color:#b8860b; padding:20px 40px; text-align:center; box-shadow:0 4px 10px rgba(0,0,0,0.1); }
        header h1 { margin:0; font-size:2.5em; }

        nav { text-align:right; padding:10px 40px; background:#ffe6f0; }
        nav a { margin-left:15px; font-weight:bold; color:#b8860b; }
        nav a:hover { color:#ff69b4; }

        h2 { text-align:center; margin:40px 0 20px 0; }

.products-container {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 30px;
    padding: 50px 40px;
}

.product-card {
    background: #fff;
    border-radius: 20px;
    overflow: hidden;
    text-align: center;
    box-shadow: 0 8px 20px rgba(0,0,0,0.1);
    transition: transform 0.3s, box-shadow 0.3s;
    display: flex;
    flex-direction: column;
```

```
justify-content: flex-start;
height: 450px;
}

.product-card:hover {
  transform: translateY(-8px);
  box-shadow: 0 15px 25px rgba(0,0,0,0.2);
}

.product-card img {
  width: 100%;
  height: 260px;
  object-fit: cover;
  border-top-left-radius: 20px;
  border-top-right-radius: 20px;
  transition: transform 0.4s;
}

.product-card img:hover {
  transform: scale(1.05);
}

.product-card h3 {
  margin: 10px 0 5px 0;
}

.product-card p {
  margin: 5px 0 10px 0;
  font-weight: bold;
  font-size: 1.1em;
}

.product-card button {
  margin-top: auto;
  margin-bottom: 20px;
  padding: 10px 20px;
  background: #b8860b;
  color: #fff;
  border: none;
  border-radius: 8px;
  font-weight: bold;
  cursor: pointer;
```

```
        transition: 0.3s;
    }

.product-card button:hover {
    background: #ff69b4;
}

@media (max-width: 992px) {
    .products-container {
        grid-template-columns: repeat(2, 1fr);
        padding: 30px 20px;
    }
}

@media (max-width: 600px) {
    .products-container {
        grid-template-columns: 1fr;
        padding: 20px 15px;
    }
    nav { text-align:center; }
}

footer {
    background:#fff;
    color:#b8860b;
    text-align:center;
    padding:30px 20px;
    margin-top:50px;
    box-shadow:0 -4px 10px rgba(0,0,0,0.05);
}

footer p { margin:5px 0; }
footer a { color:#b8860b; font-weight:bold; }
footer a:hover { color:#ff69b4; }

</style>
</head>
<body>
<header>
    <h1>Sweet Delights Bakery</h1>
</header>

<nav>
```

```

<%
    if(userId != null){
%>
    Welcome, <%=session.getAttribute("name") %> | 
    <a href="checkout.jsp">Cart</a> | 
    <a href="contact_queries.jsp">Contact Queries</a> | 
    <a href="logout.jsp">Logout</a>
<%
} else {
%>
    <a href="login.jsp">Login</a> | <a href="signup.jsp">Sign Up</a>
<%
}
%>
</nav>

<h2>
<%
if(categoryIdParam != null){
    try{
        int catId = Integer.parseInt(categoryIdParam);
        PreparedStatement ps = conn.prepareStatement("SELECT name FROM categories WHERE id=?");
        ps.setInt(1, catId);
        ResultSet rsCat = ps.executeQuery();
        if(rsCat.next()){
            out.print(rsCat.getString("name") + " Products");
        } else {
            out.print("Products");
        }
        rsCat.close(); ps.close();
    } catch(Exception e){ out.print("Products"); }
} else {
    out.print("All Products");
}
%>
</h2>

<div class="products-container">
<%
if(conn != null){
    try {
        String query;

```

```

PreparedStatement ps;
if(categoryIdParam != null){
    query = "SELECT p.id, p.name, p.price, c.name AS category_name FROM products p JOIN
categories c ON p.category_id = c.id WHERE p.category_id = ?";
    ps = conn.prepareStatement(query);
    ps.setInt(1, Integer.parseInt(categoryIdParam));
} else {
    query = "SELECT p.id, p.name, p.price, c.name AS category_name FROM products p JOIN
categories c ON p.category_id = c.id";
    ps = conn.prepareStatement(query);
}

ResultSet rs = ps.executeQuery();
while(rs.next()){
    int productId = rs.getInt("id");
    String productName = rs.getString("name");
    double price = rs.getDouble("price");
    String categoryName = rs.getString("category_name");

    String folderName = categoryName.toLowerCase();
    String imageName = productName.toLowerCase().replace(" ", "_") + ".jpg";
%>
<div class="product-card">
    ">
    <h3><%=productName %></h3>
    <p>₹<%=String.format("%.2f", price)%></p>
    <form method="post" action="add_to_cart.jsp">
        <input type="hidden" name="product_id" value="<%=productId%>">
        <button type="submit">Add to Cart</button>
    </form>
</div>
<%
}
rs.close(); ps.close();
} catch(Exception e){
    out.println("<p style='color:red;'>Error loading products: "+e.getMessage()+"</p>");
}
} else {
    out.println("<p style='color:red;'>Database connection not available!</p>");
}
%>

```

```

</div>

<footer>
    <p>Contact us: <a href="mailto:info@sweetdelights.com">info@sweetdelights.com</a> | Phone:
+91-9876543210</p>
    <p>Address: 123, Bakery Street, Mumbai, India</p>
    <p>Follow us:
        <a href="#">Instagram</a> |
        <a href="#">Facebook</a> |
        <a href="#">Twitter</a>
    </p>
    <p>&copy; 2025 Sweet Delights Bakery. All rights reserved.</p>
</footer>
</body>
</html>

```

5.add_to_cart.jsp

```

<%@ page import="java.sql.* , java.util.*" %>
<%@ page import="com.bakery.util.DBConnection" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%
String productIdStr = request.getParameter("product_id");
String qtyStr = request.getParameter("qty");
int productId = -1;
int qty = 1; // Default quantity

// Validate product ID
if (productIdStr != null && !productIdStr.trim().isEmpty()) {
    try {
        productId = Integer.parseInt(productIdStr);
    } catch (NumberFormatException e) {
        out.println("<p style='color:red;'>X Invalid product ID format.</p>");
        return;
    }
}

// Validate quantity
if (qtyStr != null && !qtyStr.trim().isEmpty()) {
    try {
        qty = Integer.parseInt(qtyStr);
        if (qty <= 0) {

```

```

        out.println("<p style='color:red;'>Quantity must be a positive integer.</p>");
        return;
    }
} catch (NumberFormatException e) {
    out.println("<p style='color:red;'>Invalid quantity format.</p>");
    return;
}
}

// Fetch product details from database
if (productId != -1) {
    try (Connection conn = DBConnection.getConnection()) {
        if (conn != null) {
            PreparedStatement ps = conn.prepareStatement("SELECT name, price FROM products WHERE id=?");
            ps.setInt(1, productId);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                String name = rs.getString("name");
                double price = rs.getDouble("price");

                // Retrieve or initialize cart
                List<Map<String, Object>> cart = (List<Map<String, Object>>) session.getAttribute("cart");
                if (cart == null) {
                    cart = new ArrayList<>();
                    session.setAttribute("cart", cart);
                }

                // Check if product already in cart
                boolean found = false;
                for (Map<String, Object> item : cart) {
                    if (item.get("name").equals(name)) {
                        int existingQty = (Integer) item.get("qty");
                        item.put("qty", existingQty + qty);
                        found = true;
                        break;
                    }
                }

                // Add new item to cart
                if (!found) {

```

```

        Map<String, Object> newItem = new HashMap<>();
        newItem.put("name", name);
        newItem.put("qty", qty);
        newItem.put("price", price);
        cart.add(newItem);
    }

    // Redirect to cart page
    response.sendRedirect("cart.jsp");
} else {
    out.println("<p style='color:red;'>X Product not found.</p>");
}
} else {
    out.println("<p style='color:red;'>X Database connection failed.</p>");
}
} catch (SQLException e) {
    out.println("<p style='color:red;'>X Database error: " + e.getMessage() + "</p>");
}
} else {
    out.println("<p style='color:red;'>X Invalid product ID.</p>");
}
%>

```

6. checkout.jsp

```

<%@ page import="java.util.*" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Your Cart - Sweet Delights Bakery</title>
    <link href="https://fonts.googleapis.com/css2?family=Georgia&display=swap" rel="stylesheet">
    <style>
        body { font-family:'Georgia', serif; margin:0; padding:0; background:#ffe6f0; color:#b8860b; }
        header { background:#fff; color:#b8860b; padding:20px 40px; text-align:center; box-shadow:0 4px 10px rgba(0,0,0,0.1); }
        header h1 { margin:0; font-size:2.5em; }

        .cart-container { max-width:900px; margin:50px auto; background:#fff; padding:30px; border-radius:20px; box-shadow:0 8px 20px rgba(0,0,0,0.1); }
        table { width:100%; border-collapse:collapse; margin-bottom:20px; }
        th, td { padding:12px; border-bottom:1px solid #ddd; text-align:center; }
    </style>

```

```

th { background:#ffe6f0; color:#b8860b; font-size:1.1em; }
td { font-size:1em; }
.total-row td { font-weight:bold; font-size:1.2em; border-top:2px solid #b8860b; }

.btn { padding:12px 20px; border:none; border-radius:10px; font-weight:bold; cursor:pointer;
transition:0.3s; }
.btn-checkout { background:#b8860b; color:#fff; margin-right:10px; }
.btn-checkout:hover { background:#ff69b4; }
.btn-reset { background:#ff4444; color:#fff; }
.btn-reset:hover { background:#cc0000; }

.actions { text-align:right; margin-top:20px; }
.empty { text-align:center; font-size:1.2em; color:#ff4444; margin-top:20px; }

footer { background:#fff; color:#b8860b; text-align:center; padding:30px 20px; margin-top:50px;
box-shadow:0 -4px 10px rgba(0,0,0,0.05); }
footer p { margin:5px 0; }
footer a { color:#b8860b; font-weight:bold; }
footer a:hover { color:#ff69b4; }
</style>
</head>
<body>
<header>
  <h1>Sweet Delights Bakery</h1>
</header>

<div class="cart-container">
  <h2>Your Shopping Cart</h2>
  <%
    List<Map<String, Object>> cart = (List<Map<String, Object>>) session.getAttribute("cart");
    double total = 0.0;

    // Handle Reset Cart
    if ("reset".equals(request.getParameter("action"))) {
      session.removeAttribute("cart");
      cart = null;
    }

    if (cart != null && !cart.isEmpty()) {
  %>
  <form method="post" action="checkout.jsp">
    <table>

```

```

<tr>
    <th>Item</th>
    <th>Quantity</th>
    <th>Price (₹)</th>
    <th>Subtotal (₹)</th>
</tr>
<%
for (Map<String, Object> item : cart) {
    String name = (String) item.get("name");
    int qty = (Integer) item.get("qty");
    double price = (Double) item.get("price");
    double subtotal = qty * price;
    total += subtotal;
%>
<tr>
    <td><%= name %></td>
    <td><%= qty %></td>
    <td><%= String.format("%.2f", price) %></td>
    <td><%= String.format("%.2f", subtotal) %></td>
</tr>
<% } %>
<tr class="total-row">
    <td colspan="3" align="right">Total:</td>
    <td>₹<%= String.format("%.2f", total) %></td>
</tr>
</table>

<div class="actions">
    <button type="submit" class="btn btn-checkout">Proceed to Checkout</button>
    <a href="cart.jsp?action=reset" class="btn btn-reset">Reset Cart</a>
</div>
</form>
<%
} else {
%>
<p class="empty">Your cart is empty! <img alt="empty cart icon" style="vertical-align: middle;" /></p>
<div class="actions">
    <a href="products.jsp" class="btn btn-checkout">Browse Products</a>
</div>
<%
}
%>

```

```

</div>

<footer>
    <p>Contact us: <a href="mailto:info@sweetdelights.com">info@sweetdelights.com</a> | Phone:
+91-9876543210</p>
    <p>Address: 123, Bakery Street, Mumbai, India</p>
    <p>&copy; 2025 Sweet Delights Bakery. All rights reserved.</p>
</footer>
</body>
</html>

```

7.checkout.jsp

```

<%@ page import="java.sql.* , java.util.*" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    Integer userId = (Integer) session.getAttribute("user_id");
    String userName = (String) session.getAttribute("name");

    if (userId == null) {
        response.sendRedirect("login.jsp");
        return;
    }

    List<Map<String, Object>> cart = (List<Map<String, Object>>) session.getAttribute("cart");
    double totalAmount = 0;

    if (cart != null && !cart.isEmpty()) {
        for (Map<String, Object> item : cart) {
            double price = (Double) item.get("price");
            totalAmount += price;
        }
    }
%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Checkout - Sweet Delights Bakery</title>
    <style>
        body {
            font-family: 'Georgia', serif;

```

```
background: #ffe6f0;
color: #b8860b;
padding: 40px;
}

.container {
  display: flex;
  gap: 40px;
  max-width: 1200px;
  margin: auto;
  flex-wrap: wrap;
}

.card, .form {
  background: #fff;
  border-radius: 20px;
  box-shadow: 0 8px 20px rgba(0,0,0,0.05);
  padding: 30px;
  flex: 1 1 45%;
}

h2, h3 {
  color: #e63946;
}

table {
  width: 100%;
  border-collapse: collapse;
}

th, td {
  padding: 10px;
  border: 1px solid #ccc;
}

input, select {
  width: 100%;
  padding: 12px;
  margin: 10px 0;
  border-radius: 8px;
  border: 1px solid #ccc;
}

.payment-methods label {
  display: block;
  margin: 10px 0;
}

.card-details {
  display: none;
```

```

        }
    button {
        background-color: #b8860b;
        color: #fff;
        padding: 12px 25px;
        border: none;
        border-radius: 10px;
        font-weight: bold;
        cursor: pointer;
    }
    button:hover {
        background-color: #ff69b4;
    }
</style>

<script>
    function toggleCardFields() {
        const method = document.querySelector('input[name="payment_method"]:checked').value;
        const cardFields = document.getElementById("card-fields");
        if (method === "Credit/Debit Card") {
            cardFields.style.display = "block";
        } else {
            cardFields.style.display = "none";
        }
    }
</script>
</head>
<body>

<h1>Checkout - Sweet Delights Bakery</h1>

<div class="container">
    <!-- Order Summary -->
    <div class="card">
        <h2>Order Summary</h2>
        <table>
            <tr>
                <th>Product</th>
                <th>Price</th>
            </tr>
            <%
                if (cart != null && !cart.isEmpty()) {

```

```

        for (Map<String, Object> item : cart) {
            String name = (String) item.get("name");
            double price = (Double) item.get("price");
        %>
        <tr>
            <td><%= name %></td>
            <td>₹<%= price %></td>
        </tr>
        <%
        }
    %>
    <tr>
        <td><strong>Total</strong></td>
        <td><strong>₹<%= totalAmount %></strong></td>
    </tr>
    <%
    } else {
    %>
    <tr>
        <td colspan="2">Your cart is empty.</td>
    </tr>
    <% } %>
</table>
</div>

<!-- Payment Form -->
<div class="form">
    <h3>Payment Method</h3>
    <form method="post" action="order_success.jsp">
        <div class="payment-methods" onchange="toggleCardFields()">
            <label><input type="radio" name="payment_method" value="Cash on Delivery" required> Cash on Delivery</label>
            <label><input type="radio" name="payment_method" value="UPI"> UPI</label>
            <label><input type="radio" name="payment_method" value="Credit/Debit Card"> Credit/Debit Card</label>
        </div>
        <!-- Card Details Section (Initially Hidden) -->
        <div id="card-fields" class="card-details">
            <h4>Card Information</h4>
            <input type="text" name="card_number" placeholder="Card Number" maxlength="16">
            <input type="text" name="card_expiry" placeholder="Expiry (MM/YY)">
        </div>
    </form>
</div>
```

```

<input type="password" name="card_cvv" placeholder="CVV" maxlength="4">
</div>

<input type="hidden" name="amount" value="<%= totalAmount %>">

<button type="submit">Place Order</button>
</form>
</div>
</div>
</body>
</html>

```

7.order_success.jsp

```

<%@ page import="java.sql.* , com.bakery.util.DBConnection" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%
Connection conn = DBConnection.getConnection();
Integer userId = (Integer) session.getAttribute("user_id");

if(userId != null && request.getMethod().equalsIgnoreCase("POST")) {
    String paymentMethod = request.getParameter("payment_method");
    String amount = request.getParameter("amount");

    try {
        String orderSql = "INSERT INTO orders (user_id, payment_method, amount, status) VALUES (?, ?, ?, 'Pending')";
        PreparedStatement ps = conn.prepareStatement(orderSql,
Statement.RETURN_GENERATED_KEYS);
        ps.setInt(1, userId);
        ps.setString(2, paymentMethod);
        ps.setString(3, amount);
        ps.executeUpdate();

        ResultSet rs = ps.getGeneratedKeys();
        int orderId = 0;
        if(rs.next()) { orderId = rs.getInt(1); }

        // Insert order items from cart
        java.util.List cart = (java.util.List) session.getAttribute("cart");
        if(cart != null && !cart.isEmpty()) {

```

```

        for(Object itemObj : cart) {
            java.util.Map item = (java.util.Map) itemObj;
            int productId = (Integer) item.get("id");
            String itemSql = "INSERT INTO order_item (order_id, product_id) VALUES (?, ?)";
            PreparedStatement itemPs = conn.prepareStatement(itemSql);
            itemPs.setInt(1, orderId);
            itemPs.setInt(2, productId);
            itemPs.executeUpdate();
            itemPs.close();
        }
    }

    session.removeAttribute("cart");
    rs.close();
    ps.close();

} catch(Exception e) {
    out.println("<p style='color:red;'>Error: " + e.getMessage() + "</p>");
}
} else {
    response.sendRedirect("checkout.jsp");
}
%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Order Success - Sweet Delights Bakery</title>
<link href="https://fonts.googleapis.com/css2?family=Georgia&display=swap" rel="stylesheet">
<style>
    body { font-family: 'Georgia', serif; margin:0; padding:0; background:#ffe6f0; color:#b8860b;
display:flex; justify-content:center; align-items:center; height:100vh; }
    a { text-decoration:none; transition:0.3s; }
    a:hover { color:#ff69b4; }
    .success-card { background:#fff; border-radius:20px; box-shadow:0 8px 20px rgba(0,0,0,0.05);
padding:50px; max-width:500px; text-align:center; }
    .success-card h1 { color:#e63946; font-size:2.5em; margin-bottom:20px; }
    .success-card p { font-size:1.2em; line-height:1.6; margin-bottom:30px; color:#333; }
    .success-card .home-btn { background-color:#b8860b; color:#fff; padding:12px 25px;
border-radius:10px; border:none; font-weight:bold; cursor:pointer; transition:0.3s; text-decoration:none;
font-size:1em; }

```

```

.success-card .home-btn:hover { background:#ff69b4; }
.success-icon { font-size:50px; color:#e63946; margin-bottom:15px; }
</style>
</head>
<body>

<div class="success-card">
    <div class="success-icon">✓</div>
    <h1>Order Successful!</h1>
    <p>Your order has been placed successfully.<br>
        Total Amount: ₹<b><%= request.getParameter("amount") %></b><br>
        Payment Method: <b><%= request.getParameter("payment_method") %></b>
    </p>
    <a href="index.jsp" class="home-btn">Back to Home</a>
</div>

</body>
</html>

```

```

8.order_history
<%@ page import="java.sql.* , java.util.*" %>
<%@ page session="true" %>
<%@ page contentType="text/html;charset=UTF-8" %>
<%@ page import="com.bakery.util.DBConnection" %>

<%
    Integer userId = (Integer) session.getAttribute("user_id");
    if(userId == null){
        response.sendRedirect("login.jsp");
        return;
    }

    // Store orders and their items
    class OrderItem {
        String name; int quantity; double price;
        OrderItem(String n, int q, double p){ name=n; quantity=q; price=p; }
    }
    class Order {
        int id; double total; Timestamp date; String payment, address;
        List<OrderItem> items = new ArrayList<>();
        Order(int id, double total, Timestamp date, String payment, String address){

```

```

        this.id=id; this.total=total; this.date=date; this.payment=payment; this.address=address;
    }
}

List<Order> orders = new ArrayList<>();

try(Connection conn = DBConnection.getConnection()){
    String sqlOrders = "SELECT * FROM orders WHERE user_id=? ORDER BY created_at DESC";
    PreparedStatement psOrders = conn.prepareStatement(sqlOrders);
    psOrders.setInt(1, userId);
    ResultSet rsOrders = psOrders.executeQuery();
    while(rsOrders.next()){
        int orderId = rsOrders.getInt("id");
        double total = rsOrders.getDouble("total_amount");
        Timestamp date = rsOrders.getTimestamp("created_at");
        String payment = rsOrders.getString("payment_method");
        String address = rsOrders.getString("address");
        Order order = new Order(orderId, total, date, payment, address);

        // Get order items
        String sqlItems = "SELECT * FROM order_items WHERE order_id=?";
        PreparedStatement psItems = conn.prepareStatement(sqlItems);
        psItems.setInt(1, orderId);
        ResultSet rsItems = psItems.executeQuery();
        while(rsItems.next()){
            String name = rsItems.getString("product_name");
            int qty = rsItems.getInt("quantity");
            double price = rsItems.getDouble("price");
            order.items.add(new OrderItem(name, qty, price));
        }
        orders.add(order);
    }
} catch(Exception e){
    out.println("Error: "+e.getMessage());
}
%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Order History</title>

```

```

<style>
  body { font-family: Arial, sans-serif; background: #fff7f0; padding: 40px; }
  h1 { color: #d2691e; }
  .order { margin-bottom: 40px; padding: 20px; border: 1px solid #ddd; border-radius: 10px;
background: #fff2e0; }
  table { width: 100%; border-collapse: collapse; margin-top: 10px; }
  th, td { border: 1px solid #ddd; padding: 8px; text-align: center; }
  th { background-color: #f2cba0; }
</style>
</head>
<body>
  <h1>My Orders</h1>

  <%
    if(orders.isEmpty()){
  %>
    <p>You have no orders yet.</p>
  <%
    } else {
      for(Order o : orders){
  %>
    <div class="order">
      <h2>Order ID: <%= o.id %> | Total: ₹<%= o.total %> | Date: <%= o.date %></h2>
      <p>Payment: <%= o.payment %> | Shipping: <%= o.address %></p>
      <table>
        <tr>
          <th>Product</th>
          <th>Quantity</th>
          <th>Price</th>
          <th>Subtotal</th>
        </tr>
        <%
          for(OrderItem item : o.items){
  %>
        <tr>
          <td><%= item.name %></td>
          <td><%= item.quantity %></td>
          <td>₹<%= item.price %></td>
          <td>₹<%= item.price * item.quantity %></td>
        </tr>
        <%
      }
    
```

```
%>
</table>
</div>
<%
    }
}
%>
</body>
</html>

9.contact_queries.jsp
<%@ page import="java.sql.* , java.util.*" %>
<%@ page import="com.bakery.util.DBConnection" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
<title>Contact Us - Sweet Delights</title>
<meta charset="UTF-8">
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #fff8f0;
    padding: 50px;
    color: #333;
}
h2 {
    color: #b8860b;
    text-align: center;
}
form {
    max-width: 500px;
    margin: auto;
    background: #fff;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 8px 20px rgba(0,0,0,0.1);
}
input, textarea {
    width: 100%;
    padding: 10px;
    margin-top: 15px;
}
```

```

    border: 1px solid #ccc;
    border-radius: 5px;
}
button {
    margin-top: 20px;
    padding: 12px;
    width: 100%;
    background-color: #b8860b;
    color: white;
    border: none;
    font-weight: bold;
    border-radius: 5px;
    cursor: pointer;
}
button:hover {
    background-color: #ff69b4;
}
.thank-you {
    max-width: 500px;
    margin: 100px auto;
    background: #fff;
    padding: 30px;
    text-align: center;
    border-radius: 10px;
    box-shadow: 0 8px 20px rgba(0,0,0,0.1);
    color: #4CAF50;
    font-size: 1.3em;
}

```

</style>

</head>

<body>

<%

```

boolean submitted = false;

if("POST".equalsIgnoreCase(request.getMethod())) {
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String message = request.getParameter("message");

    Connection conn = DBConnection.getConnection();
    if(conn != null){

```

```

try {
    String sql = "INSERT INTO contact_queries (name, email, message, created_at) VALUES (?, ?, ?, NOW())";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setString(1, name);
    ps.setString(2, email);
    ps.setString(3, message);
    int rows = ps.executeUpdate();
    ps.close();

    if(rows > 0){
        submitted = true;
    }
}

} catch(Exception e){
    out.println("<p style='color:red; text-align:center;'>Error: " + e.getMessage() + "</p>");
}
} else {
    out.println("<p style='color:red; text-align:center;'>Database connection not available!</p>");
}
}

if(submitted){

%>
<!--  Thank You message -->
<div class="thank-you">
    <p>🎉 Thank you for contacting us! We'll get back to you shortly.</p>
    <p>Redirecting to homepage in 3 seconds...</p>
</div>

<!--  Auto redirect using JavaScript -->
<script>
    setTimeout(function(){
        window.location.href = "index.jsp";
    }, 3000);
</script>
<%
    } else {
%>

<h2>Contact Us</h2>
<form method="post" action="">

```

```

<input type="text" name="name" placeholder="Your Name" required>
<input type="email" name="email" placeholder="Your Email" required>
<textarea name="message" placeholder="Your Message" rows="5" required></textarea>
<button type="submit">Send Message</button>
</form>

<%
  }
%>

</body>
</html>

```

10. Cart.jsp

```

<%@ page import="java.util.*" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Your Cart - Sweet Delights Bakery</title>
  <link href="https://fonts.googleapis.com/css2?family=Georgia&display=swap" rel="stylesheet">
  <style>
    body { font-family:'Georgia', serif; margin:0; padding:0; background:#ffe6f0; color:#b8860b; }
    header { background:#fff; color:#b8860b; padding:20px 40px; text-align:center; box-shadow:0 4px 10px rgba(0,0,0,0.1); }
    header h1 { margin:0; font-size:2.5em; }

    .cart-container { max-width:900px; margin:50px auto; background:#fff; padding:30px; border-radius:20px; box-shadow:0 8px 20px rgba(0,0,0,0.1); }
    table { width:100%; border-collapse:collapse; margin-bottom:20px; }
    th, td { padding:12px; border-bottom:1px solid #ddd; text-align:center; }
    th { background:#ffe6f0; color:#b8860b; font-size:1.1em; }
    td { font-size:1em; }
    .total-row td { font-weight:bold; font-size:1.2em; border-top:2px solid #b8860b; }

    .btn { padding:12px 20px; border:none; border-radius:10px; font-weight:bold; cursor:pointer; transition:0.3s; }
    .btn-checkout { background:#b8860b; color:#fff; margin-right:10px; }
    .btn-checkout:hover { background:#ff69b4; }
    .btn-reset { background:#ff4444; color:#fff; }
    .btn-reset:hover { background:#cc0000; }
  </style>
</head>
<body>
  <header>
    <h1>Your Cart</h1>
  </header>
  <div class="cart-container">
    <table>
      <thead>
        <tr>
          <th>Item</th>
          <th>Description</th>
          <th>Quantity</th>
          <th>Price</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Chocolate Chip Cookies</td>
          <td>1</td>
          <td>$12.00</td>
        </tr>
        <tr>
          <td>1</td>
          <td>Vanilla Cupcakes</td>
          <td>1</td>
          <td>$8.00</td>
        </tr>
        <tr>
          <td>1</td>
          <td>Strawberry Muffins</td>
          <td>1</td>
          <td>$6.00</td>
        </tr>
        <tr class="total-row">
          <td></td>
          <td></td>
          <td></td>
          <td>Subtotal: $26.00</td>
        </tr>
      </tbody>
    </table>
    <div>
      <button class="btn btn-checkout">Checkout</button>
      <button class="btn btn-reset">Reset</button>
    </div>
  </div>
</body>

```

```

.actions { text-align:right; margin-top:20px; }
.empty { text-align:center; font-size:1.2em; color:#ff4444; margin-top:20px; }

footer { background:#fff; color:#b8860b; text-align:center; padding:30px 20px; margin-top:50px;
box-shadow:0 -4px 10px rgba(0,0,0,0.05); }
footer p { margin:5px 0; }
footer a { color:#b8860b; font-weight:bold; }
footer a:hover { color:#ff69b4; }

</style>
</head>
<body>
<header>
    <h1>Sweet Delights Bakery</h1>
</header>

<div class="cart-container">
    <h2>Your Shopping Cart</h2>
    <%
        List<Map<String, Object>> cart = (List<Map<String, Object>>) session.getAttribute("cart");
        double total = 0.0;

        // Handle Reset Cart
        if ("reset".equals(request.getParameter("action"))) {
            session.removeAttribute("cart");
            cart = null;
        }

        if (cart != null && !cart.isEmpty()) {
    %>
    <form method="post" action="checkout.jsp">
        <table>
            <tr>
                <th>Item</th>
                <th>Quantity</th>
                <th>Price (₹)</th>
                <th>Subtotal (₹)</th>
            </tr>
    <%
        for (Map<String, Object> item : cart) {
            String name = (String) item.get("name");
            int qty = (Integer) item.get("qty");

```

```

        double price = (Double) item.get("price");
        double subtotal = qty * price;
        total += subtotal;

    %>
<tr>
    <td><%= name %></td>
    <td><%= qty %></td>
    <td><%= String.format("%.2f", price) %></td>
    <td><%= String.format("%.2f", subtotal) %></td>
</tr>
<% } %>
<tr class="total-row">
    <td colspan="3" align="right">Total:</td>
    <td>₹<%= String.format("%.2f", total) %></td>
</tr>
</table>

<div class="actions">
    <button type="submit" class="btn btn-checkout">Proceed to Checkout</button>
    <a href="cart.jsp?action=reset" class="btn btn-reset">Reset Cart</a>
</div>
</form>
<%
    } else {
%>
    <p class="empty">Your cart is empty! 

```

```
11.logout.jsp
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%
// Invalidate session safely
if (session != null) {
    session.invalidate();
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Logout - Sweet Delights Bakery</title>
<link href="https://fonts.googleapis.com/css2?family=Georgia:wght@400,700&display=swap"
rel="stylesheet">
<style>
body {
    font-family: 'Georgia', serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    background: #ffe6f0;
    color: #b8860b;
    flex-direction: column;
}
.logout-container {
    background: #fff;
    padding: 40px 30px;
    border-radius: 20px;
    box-shadow: 0 8px 20px rgba(0,0,0,0.1);
    text-align: center;
    animation: fadeln 1s ease-in-out;
}
h1 {
    margin-bottom: 20px;
}
p {
    font-size: 1.1em;
    margin-bottom: 10px;
```

```

        }
    @keyframes fadeIn {
        from {opacity: 0; transform: translateY(-20px);}
        to {opacity: 1; transform: translateY(0);}
    }
</style>
<!--  Auto redirect after 3 seconds -->
<meta http-equiv="refresh" content="3;url=login.jsp">
<script>
let countdown = 3;
function updateCountdown() {
    if(countdown > 0){
        document.getElementById("count").innerText = countdown;
        countdown--;
        setTimeout(updateCountdown, 1000);
    }
}
window.onload = updateCountdown;
</script>
</head>
<body>
<div class="logout-container">
    <h1>Logging out...</h1>
    <p>You will be redirected to the login page in <span id="count">3</span> seconds.</p>
    <p>Thank you for visiting <b>Sweet Delights Bakery!</b></p>
</div>
</body>
</html>

```

12.DBConnection.java

```

package com.bakery.util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    // Static method to get database connection
    public static Connection getConnection() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver"); // Load driver

```

```

        return DriverManager.getConnection(
"jdbc:mysql://localhost:3306/bakery_db?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC",
        "root", // XAMPP username
        "" // XAMPP password
    );
} catch (ClassNotFoundException | SQLException e) {
    return null;
}
}

// Optional: test connection
public static void main(String[] args) {
    Connection c = getConnection();
    if(c != null) System.out.println("✅ DB Connected!");
    else System.out.println("❌ DB Connection Failed!");
}
}

```

13.pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.bakery</groupId>
    <artifactId>bakery</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>bakery-1.0-SNAPSHOT</name>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <jakartaee>10.0.0</jakartaee>
        <maven.compiler.release>11</maven.compiler.release>
    </properties>

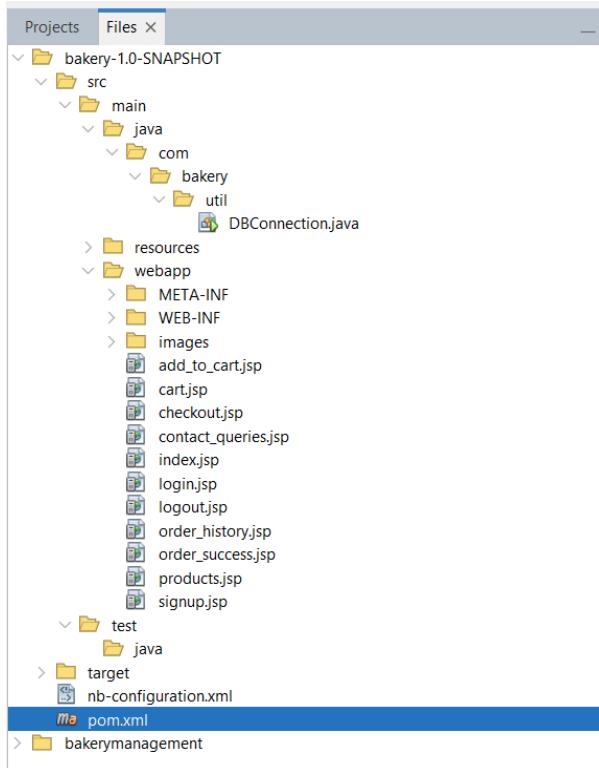
    <dependencies>
        <dependency>

```

```
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.33</version>
</dependency>
<!-- Jakarta EE API (for Servlets, JSP, etc.) -->
<dependency>
<groupId>jakarta.platform</groupId>
<artifactId>jakarta.jakartaee-api</artifactId>
<version>${jakartaee}</version>
<scope>provided</scope>
</dependency>
</dependencies>

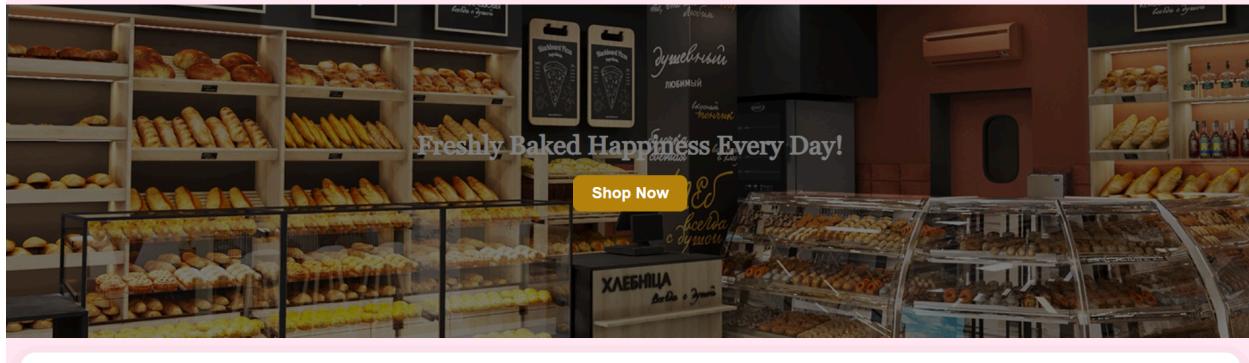
</project>
```

Screenshots



Sweet Delights Bakery

Welcome, shruti |
[Cart](#)
[My Orders](#)
[Logout](#)
[Contact Us](#)



Freshly Baked Happiness Every Day!

[Shop Now](#)



About Sweet Delights

We bake with love and passion, creating cakes, pastries, bread, and cookies with the freshest ingredients.

Every product is handcrafted to ensure a sweet experience for you and your loved ones.

Explore Our Categories



Cakes



Pastries



Bread



Cookies

Sign Up

Name:

Email:

Password:

[Sign Up](#)

Login

Email:

Password:

Login

[Don't have an account? Sign Up](#)

Sweet Delights Bakery

Welcome, shruti | [Cart](#) | [Contact Queries](#) | [Logout](#)

Pastries Products



Chocolate Pastry
₹80.00

[Add to Cart](#)



Strawberry Pastry
₹90.00

[Add to Cart](#)



Pineapple Pastry
₹75.00

[Add to Cart](#)

Your Shopping Cart

Item	Quantity	Price (₹)	Subtotal (₹)
Strawberry Pastry	1	90.00	90.00
Total:			₹90.00

[Proceed to Checkout](#) [Reset Cart](#)

Contact us: info@sweetdelights.com | Phone: +91-9876543210
 Address: 123, Bakery Street, Mumbai, India
 © 2025 Sweet Delights Bakery. All rights reserved.

Checkout - Sweet Delights Bakery

Order Summary

Product	Price
Strawberry Pastry	₹90.0
Total	₹90.0

Payment Method

Cash on Delivery
 UPI
 Credit/Debit Card

Place Order



Order Successful!

Your order has been placed successfully.

Total Amount: ₹90.0

Payment Method: Cash on Delivery

Back to Home

Contact Us

Your Name

Your Email

Your Message

Send Message

Database

Table	Action	Rows	Type	Collation	Size	Overhead
cart		2	InnoDB	utf8mb4_general_ci	48.0 KiB	-
categories		4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
contact_queries		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
orders		0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
order_items		0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
payments		0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
products		24	InnoDB	utf8mb4_general_ci	32.0 KiB	-
users		3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
8 tables	Sum	36	InnoDB	utf8mb4_general_ci	256.0 KiB	0 B

		id	name	description	image
<input type="checkbox"/>		1	Cakes	Delicious freshly baked cakes of various flavors	images/categories/cakes.jpg
<input type="checkbox"/>		2	Pastries	Sweet pastries perfect for every occasion	images/categories/pastries.jpg
<input type="checkbox"/>		3	Bread	Freshly baked bread and loaves	images/categories/bread.jpg
<input type="checkbox"/>		4	Cookies	Crunchy and soft cookies in different flavors	images/categories/cookies.jpg

		id	name	description	price	quantity	image	category_id
<input type="checkbox"/>		1	Chocolate Truffle Cake	Rich chocolate layered cake with truffle frosting	599.00	1kg	images/cakes/chocolate_truffle.jpg	1
<input type="checkbox"/>		2	Red Velvet Cake	Soft red velvet cake with cream cheese frosting	699.00	1kg	images/cakes/red_velvet.jpg	1
<input type="checkbox"/>		3	Black Forest Cake	Classic German chocolate cake with cherries & crea...	549.00	1kg	images/cakes/black_forest.jpg	1
<input type="checkbox"/>		4	Butterscotch Cake	Caramel butterscotch flavored creamy cake	579.00	1kg	images/cakes/butterscotch.jpg	1
<input type="checkbox"/>		5	Pineapple Cake	Fresh pineapple flavored sponge cake with cream	499.00	1kg	images/cakes/pineapple.jpg	1
<input type="checkbox"/>		6	Fruit Cake	Seasonal fruit-topped vanilla sponge cake	649.00	1kg	images/cakes/fruit_cake.jpg	1
<input type="checkbox"/>		7	Chocolate Pastry	Soft chocolate pastry slice with fudge topping	80.00	1 piece	images/pastries/chocolate_pastry.jpg	2
<input type="checkbox"/>		8	Strawberry Pastry	Strawberry flavored pastry slice with cream	90.00	1 piece	images/pastries/strawberry_pastry.jpg	2
<input type="checkbox"/>		9	Pineapple Pastry	Fresh pineapple pastry slice with cream frosting	75.00	1 piece	images/pastries/pineapple_pastry.jpg	2
<input type="checkbox"/>		10	Red Velvet Pastry	Mini red velvet pastry slice with cream cheese	95.00	1 piece	images/pastries/red_velvet_pastry.jpg	2
<input type="checkbox"/>		11	Black Forest Pastry	Chocolate pastry slice with cherries & whipped cre...	85.00	1 piece	images/pastries/black_forest_pastry.jpg	2
<input type="checkbox"/>		12	Butterscotch Pastry	Caramel butterscotch flavored pastry slice	90.00	1 piece	images/pastries/butterscotch_pastry.jpg	2
<input type="checkbox"/>		13	Garlic Bread	Soft baked garlic bread loaf	120.00	400g	images/bread/garlic_bread.jpg	3
<input type="checkbox"/>		14	Brown Bread	Healthy whole wheat brown bread	50.00	400g	images/bread/brown_bread.jpg	3
<input type="checkbox"/>		15	Multigrain Bread	Nutritious multigrain bread loaf	70.00	400g	images/bread/multigrain_bread.jpg	3
<input type="checkbox"/>		16	White Bread	Soft and fluffy white bread loaf	40.00	400g	images/bread/white_bread.jpg	3
<input type="checkbox"/>		17	Baguette	French style crispy baguette bread	80.00	500g	images/bread/baguette.jpg	3
<input type="checkbox"/>		18	Focaccia	Italian flatbread seasoned with herbs	100.00	450g	images/bread/focaccia.jpg	3
<input type="checkbox"/>		19	Choco Chip Cookies	Crunchy cookies loaded with chocolate chips	150.00	12 pcs	images/cookies/choco_chip.jpg	4
<input type="checkbox"/>		20	Oatmeal Cookies	Healthy cookies made with oats & honey	130.00	12 pcs	images/cookies/oatmeal.jpg	4
<input type="checkbox"/>		21	Butter Cookies	Classic buttery cookies with rich flavor	140.00	12 pcs	images/cookies/butter.jpg	4
<input type="checkbox"/>		22	Almond Cookies	Crispy cookies made with roasted almonds	160.00	12 pcs	images/cookies/almond.jpg	4

Console

The screenshot shows a MySQL Workbench interface with three tables displayed in tabs:

- users** table:

	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	id	name	email	password	phone	address
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	shruti	shruti@gmail.com	123	NULL	NULL
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2	param	abc@gmail	456	NULL	NULL
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3	sam	sam@gmailcom	999	NULL	NULL

- purchases** table:

	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	id	user_id	product_id	quantity	added_at
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	12	1	22	1	2025-09-10 01:34:34
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	13	1	1	2	2025-09-10 05:39:06

- messages** table:

	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	id	name	email	message	created_at
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	shruti	shruti@gmail.com	i love the cakes	2025-09-10 20:58:36
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2	param	param@gmail.com	The pastries are mouth-watering!	2025-09-10 21:01:35
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3	sam	sam@gmailcom	Can we please appreciate the red-velvet pastry!	2025-09-10 21:05:17

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: Filter rows: Sort by key: