**Project: Cosmetic Ecommerce Management**

**Objective:**

The **Cosmetic Ecommerce Management** is a Java web application designed to manage the sale of cosmetic products. The system allows users to browse products, add them to a shopping cart, place orders, and manage inventory. The application includes functionalities such as user registration, login, product management (CRUD operations), shopping cart management, and order processing, all controlled through a centralized Main Controller using the MVC2 pattern.

**Database Schema (Using SQL Server)**

**1. Database Name: CosmeticEcommerce**

**2. Table: tblUsers**

This table stores user information, including credentials and roles.

| **Field Name** | **Type** | **Validation** |
| --- | --- | --- |
| userID | VARCHAR(50) | Primary Key, Not null |
| fullName | NVARCHAR(500) | Not null |
| Email | VARCHAR(100) | Not null, Unique |
| phoneNumber | VARCHAR(15) | Not null, Format: [0-9]{10,15} |
| roleID | NVARCHAR(5) | Not null |
| Password | VARCHAR(50) | Not null |

sql code

CREATE TABLE tblUsers (

userID VARCHAR(50) PRIMARY KEY NOT NULL,

fullName NVARCHAR(500) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

phoneNumber VARCHAR(15) NOT NULL,

roleID NVARCHAR(5) NOT NULL,

password VARCHAR(50) NOT NULL

);

**Sample Data for tblUsers:**

sql code

INSERT INTO tblUsers (userID, fullName, email, phoneNumber, roleID, password) VALUES

('user1', 'John Doe', 'john@example.com', '1234567890', 'USR', 'password1'),

('user2', 'Jane Smith', 'jane@example.com', '0987654321', 'USR', 'password2'),

('admin', 'Admin User', 'admin@example.com', '1122334455', 'ADM', 'adminpass');

**3. Table: tblSuppliers**

This table stores information about suppliers who provide the cosmetic products.

| **Field Name** | **Type** | **Validation** |
| --- | --- | --- |
| supplierID | VARCHAR(50) | Primary Key, Not null |
| supplierName | NVARCHAR(200) | Not null |
| contactName | NVARCHAR(200) | Not null |
| phoneNumber | VARCHAR(15) | Not null, Format: [0-9]{10,15} |
| email | VARCHAR(100) | Not null, Unique |
| address | NVARCHAR(500) | Not null |

sql code

CREATE TABLE tblSuppliers (

supplierID VARCHAR(50) PRIMARY KEY NOT NULL,

supplierName NVARCHAR(200) NOT NULL,

contactName NVARCHAR(200) NOT NULL,

phoneNumber VARCHAR(15) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

address NVARCHAR(500) NOT NULL

);

**Sample Data for tblSuppliers:**

sql code

INSERT INTO tblSuppliers (supplierID, supplierName, contactName, phoneNumber, email, address) VALUES

('SUP001', 'Beauty Co.', 'Alice Johnson', '1234567890', 'alice@beautyco.com', '123 Beauty Street, New York, NY'),

('SUP002', 'Glamour Supplies', 'Bob Smith', '0987654321', 'bob@glamoursupplies.com', '456 Glamour Ave, Los Angeles, CA'),

('SUP003', 'Cosmetic Partners', 'Carol Davis', '1122334455', 'carol@cosmeticpartners.com', '789 Cosmetic Blvd, Chicago, IL');

**4. Table: tblProducts**

This table stores information about cosmetic products.

| **Field Name** | **Type** | **Validation** |
| --- | --- | --- |
| productID | VARCHAR(50) | Primary Key, Not null |
| productName | NVARCHAR(200) | Not null |
| description | NVARCHAR(1000) | Not null |
| price | DECIMAL(18, 2) | Not null |
| quantity | INT | Not null |
| category | NVARCHAR(100) | Not null |
| supplierID | VARCHAR(50) | Foreign Key (tblSuppliers.supplierID) |

sql code

CREATE TABLE tblProducts (

productID VARCHAR(50) PRIMARY KEY NOT NULL,

productName NVARCHAR(200) NOT NULL,

description NVARCHAR(1000) NOT NULL,

price DECIMAL(18, 2) NOT NULL,

quantity INT NOT NULL,

category NVARCHAR(100) NOT NULL,

supplierID VARCHAR(50) NOT NULL,

FOREIGN KEY (supplierID) REFERENCES tblSuppliers(supplierID)

);

**Sample Data for tblProducts:**

sql code

INSERT INTO tblProducts (productID, productName, description, price, quantity, category, supplierID) VALUES

('P001', 'Lipstick', 'Red lipstick, lead-free', 200000, 100, 'Makeup', 'SUP001'),

('P002', 'Moisturizing Cream', 'Night moisturizing cream, deep hydration', 350000, 50, 'Skincare', 'SUP002'),

('P003', 'Perfume', 'Fruit-scented perfume, long-lasting fragrance', 500000, 75, 'Perfume', 'SUP003'),

('P004', 'Facial Cleanser', 'Facial cleanser for sensitive skin', 150000, 30, 'Skincare', 'SUP001');

**5. Table: tblOrders**

This table stores information about customer orders.

| **Field Name** | **Type** | **Validation** |
| --- | --- | --- |
| orderID | VARCHAR(50) | Primary Key, Not null |
| userID | VARCHAR(50) | Foreign Key (tblUsers.userID) |
| orderDate | DATE | Not null |
| totalAmount | DECIMAL(18, 2) | Not null |

sql code

CREATE TABLE tblOrders (

orderID VARCHAR(50) PRIMARY KEY NOT NULL,

userID VARCHAR(50),

orderDate DATE NOT NULL,

totalAmount DECIMAL(18, 2) NOT NULL,

FOREIGN KEY (userID) REFERENCES tblUsers(userID)

);

**6. Table: tblOrderDetails**

This table stores detailed information about products in each order.

| **Field Name** | **Type** | **Validation** |
| --- | --- | --- |
| orderDetailID | INT | Primary Key, Identity (Auto Increment) |
| orderID | VARCHAR(50) | Foreign Key (tblOrders.orderID) |
| productID | VARCHAR(50) | Foreign Key (tblProducts.productID) |
| quantity | INT | Not null |
| price | DECIMAL(18, 2) | Not null |

sql code

CREATE TABLE tblOrderDetails (

orderDetailID INT IDENTITY(1,1) PRIMARY KEY,

orderID VARCHAR(50),

productID VARCHAR(50),

quantity INT NOT NULL,

price DECIMAL(18, 2) NOT NULL,

FOREIGN KEY (orderID) REFERENCES tblOrders(orderID),

FOREIGN KEY (productID) REFERENCES tblProducts(productID)

);

**Application Features**

1. **User Registration:**
   * **Page:** register.jsp
   * **Description:** New users can register by providing their full name, email, phone number, and password. The system will store this information in the tblUsers table.
2. **Login Functionality:**
   * **Page:** login.jsp
   * **Description:** Users enter their userID and password. The system checks these credentials:
     + If correct, the user is granted access and redirected to the product list (productList.jsp).
     + If incorrect, an error message “Incorrect UserID or Password” is displayed on login.jsp.
3. **Supplier Management:**
   * **Page:** supplierList.jsp
   * **Description:** Admins can view, add, update, and delete supplier information.
4. **Product Management (CRUD Operations) for Admin:**
   * **Create Product:**
     + **Page:** createProduct.jsp
     + **Description:** Admins can create new product listings by providing product details such as name, description, price, quantity, category, and supplier.
   * **Read Products:**
     + **Page:** productList.jsp
     + **Description:** Displays a list of all products. Users can filter products by category or price.
   * **Update Product:**
     + **Page:** updateProduct.jsp
     + **Description:** Admins can update the details of an existing product.
   * **Delete Product:**
     + **Page:** productList.jsp
     + **Description:** Admins can delete products. A confirmation prompt is displayed before deletion.
5. **Shopping Cart Management:**
   * **Page:** cart.jsp
   * **Description:** Users can add products to their shopping cart, view items in the cart, update quantities, and remove items.
6. **Order Processing:**
   * **Page:** checkout.jsp
   * **Description:** Users can proceed to checkout from the shopping cart, confirm their order, and place it. The system then stores the order information in the tblOrders and tblOrderDetails tables.

**Detailed Grading Table**

| **No** | **Function** | **Description** | **Mark** |
| --- | --- | --- | --- |
| 1 | **User Registration** | - On register.jsp, new users provide their full name, email, phone number, and password. - The system validates and stores the user information in the tblUsers table. | **1.0** |
| 2 | **Login** | - On login.jsp, users enter userID and password. - The system checks:    o If the userID and password are correct, grant access permission and go to productList.jsp.    o If not, show an error message. | **1.5** |
| 3 | **Supplier Management** | - Admins can view, add, update, and delete supplier information. - Proper validation is ensured (e.g., non-empty fields, correct contact information). | **1.5** |
| 4 | **Product Management (CRUD)** | - Admins can create, read, update, and delete product listings. - Proper validation is ensured (e.g., non-empty fields, correct price formats). | **2.0** |
| 5 | **Shopping Cart Management** | - Users can add products to their shopping cart, view and update quantities, and remove items. - Ensure that the cart updates dynamically as changes are made. | **2.0** |
| 6 | **Order Processing** | - Users can proceed to checkout from the shopping cart, confirm their order, and place it. - The system stores the order information in tblOrders and tblOrderDetails tables. | **2.0** |

**Total: 10 Marks**

**Implementation Guide with Main Controller (MVC2 Pattern)**

**1. Setting Up the Project**

* Create a new Dynamic Web Project named CosmeticEcommerce.
* Set up the project structure with src for Java files and WebContent for JSP files.

**2. Database Connection Utility**

Create a utility class for establishing a connection to the CosmeticEcommerce database in SQL Server.

package com.example.util;

import java.sql.Connection;

import java.sql.DriverManager;

public class DatabaseConnection {

public static Connection initializeDatabase() throws Exception {

String jdbcURL = "jdbc:sqlserver://localhost:1433;databaseName=CosmeticEcommerce";

String jdbcUsername = "sa"; // Replace with your DB username

String jdbcPassword = "your\_password"; // Replace with your DB password

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

return DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);

}

}

**3. Organizing the Code: DAO and DTO Classes**

1. **DTO Classes:** Represent the data structure.
2. **DAO Classes:** Handle all database interactions.

**4. Main Controller (MVC2 Pattern)**

The main controller handles all incoming requests and forwards them to the appropriate handler based on the action specified.

**MainController.java:**

package com.example.controller;

import javax.servlet.ServletException;

import javax.servlet.http.\*;

import java.io.IOException;

public class MainController extends HttpServlet {

private static final String LOGIN = "Login";

private static final String LOGIN\_CONTROLLER = "LoginController";

private static final String LOGOUT = "Logout";

private static final String LOGOUT\_CONTROLLER = "LogoutController";

private static final String REGISTER = "Register";

private static final String REGISTER\_CONTROLLER = "RegisterController";

private static final String CREATE\_PRODUCT = "CreateProduct";

private static final String CREATE\_PRODUCT\_CONTROLLER = "CreateProductController";

private static final String UPDATE\_PRODUCT = "UpdateProduct";

private static final String UPDATE\_PRODUCT\_CONTROLLER = "UpdateProductController";

private static final String DELETE\_PRODUCT = "DeleteProduct";

private static final String DELETE\_PRODUCT\_CONTROLLER = "DeleteProductController";

private static final String VIEW\_SUPPLIERS = "ViewSuppliers";

private static final String VIEW\_SUPPLIERS\_CONTROLLER = "SupplierController";

private static final String VIEW\_CART = "ViewCart";

private static final String VIEW\_CART\_CONTROLLER = "CartController";

private static final String CHECKOUT = "Checkout";

private static final String CHECKOUT\_CONTROLLER = "OrderController";

private static final String SEARCH\_PRODUCT = "SearchProduct";

private static final String SEARCH\_PRODUCT\_CONTROLLER = "SearchProductController";

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

String url = "login.jsp"; // Default URL to login page

try {

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

if (LOGIN.equals(action)) {

url = LOGIN\_CONTROLLER;

} else if (LOGOUT.equals(action)) {

url = LOGOUT\_CONTROLLER;

} else if (REGISTER.equals(action)) {

url = REGISTER\_CONTROLLER;

} else if (CREATE\_PRODUCT.equals(action)) {

url = CREATE\_PRODUCT\_CONTROLLER;

} else if (UPDATE\_PRODUCT.equals(action)) {

url = UPDATE\_PRODUCT\_CONTROLLER;

} else if (DELETE\_PRODUCT.equals(action)) {

url = DELETE\_PRODUCT\_CONTROLLER;

} else if (VIEW\_SUPPLIERS.equals(action)) {

url = VIEW\_SUPPLIERS\_CONTROLLER;

} else if (VIEW\_CART.equals(action)) {

url = VIEW\_CART\_CONTROLLER;

} else if (CHECKOUT.equals(action)) {

url = CHECKOUT\_CONTROLLER;

} else if (SEARCH\_PRODUCT.equals(action)) {

url = SEARCH\_PRODUCT\_CONTROLLER;

} else {

request.setAttribute("ERROR", "Action not supported.");

}

} catch (Exception e) {

log("Error at MainController: " + e.toString());

} finally {

request.getRequestDispatcher(url).forward(request, response);

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Main Controller for Cosmetic Ecommerce Management Application";

}

}

**5. Controller Classes**

These classes handle specific actions such as login, registration, CRUD operations for products, managing suppliers, managing the shopping cart, and processing orders.

1. **LoginController.java**
2. **ProductController.java**
3. **SupplierController.java**
4. **CartController.java**
5. **OrderController.java**
6. **SearchProductController.java**

These controllers interact with the MainController to handle user requests and perform the necessary operations on the backend.

**1. LoginController.java**

package com.example.controller;

import com.example.dao.UserDAO;

import com.example.dto.User;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

@WebServlet("/LoginController")

public class LoginController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

UserDAO userDAO = new UserDAO();

try {

User user = userDAO.loginUser(userID, password);

if (user != null) {

HttpSession session = request.getSession();

session.setAttribute("userID", user.getUserID());

session.setAttribute("fullName", user.getFullName());

session.setAttribute("roleID", user.getRoleID());

response.sendRedirect("dashboard.jsp");

} else {

request.setAttribute("ERROR", "Incorrect UserID or Password.");

request.getRequestDispatcher("login.jsp").forward(request, response);

}

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("ERROR", "Login failed. Please try again.");

request.getRequestDispatcher("login.jsp").forward(request, response);

}

}

}

**2. RegisterController.java**

package com.example.controller;

import com.example.dao.UserDAO;

import com.example.dto.User;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

@WebServlet("/RegisterController")

public class RegisterController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

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

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

User user = new User();

user.setFullName(fullName);

user.setEmail(email);

user.setPhoneNumber(phoneNumber);

user.setPassword(password);

user.setRoleID("USR"); // Default role as user

UserDAO userDAO = new UserDAO();

try {

boolean isRegistered = userDAO.registerUser(user);

if (isRegistered) {

response.sendRedirect("login.jsp");

} else {

request.setAttribute("ERROR", "Registration failed. Please try again.");

request.getRequestDispatcher("register.jsp").forward(request, response);

}

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("ERROR", "Registration failed. Please try again.");

request.getRequestDispatcher("register.jsp").forward(request, response);

}

}

}

**3. ProductController.java**

package com.example.controller;

import com.example.dao.ProductDAO;

import com.example.dto.Product;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

@WebServlet("/ProductController")

public class ProductController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

ProductDAO productDAO = new ProductDAO();

try {

if ("CreateProduct".equals(action)) {

Product product = new Product();

product.setProductID(request.getParameter("productID"));

product.setProductName(request.getParameter("productName"));

product.setDescription(request.getParameter("description"));

product.setPrice(Double.parseDouble(request.getParameter("price")));

product.setQuantity(Integer.parseInt(request.getParameter("quantity")));

product.setCategory(request.getParameter("category"));

product.setSupplierID(request.getParameter("supplierID"));

productDAO.addProduct(product);

response.sendRedirect("MainController?action=ViewProducts");

} else if ("UpdateProduct".equals(action)) {

Product product = new Product();

product.setProductID(request.getParameter("productID"));

product.setProductName(request.getParameter("productName"));

product.setDescription(request.getParameter("description"));

product.setPrice(Double.parseDouble(request.getParameter("price")));

product.setQuantity(Integer.parseInt(request.getParameter("quantity")));

product.setCategory(request.getParameter("category"));

product.setSupplierID(request.getParameter("supplierID"));

productDAO.updateProduct(product);

response.sendRedirect("MainController?action=ViewProducts");

} else if ("DeleteProduct".equals(action)) {

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

productDAO.deleteProduct(productID);

response.sendRedirect("MainController?action=ViewProducts");

}

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("ERROR", "Product operation failed. Please try again.");

request.getRequestDispatcher("productList.jsp").forward(request, response);

}

}

}

**4. SupplierController.java**

package com.example.controller;

import com.example.dao.SupplierDAO;

import com.example.dto.Supplier;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

@WebServlet("/SupplierController")

public class SupplierController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

SupplierDAO supplierDAO = new SupplierDAO();

try {

if ("AddSupplier".equals(action)) {

Supplier supplier = new Supplier();

supplier.setSupplierID(request.getParameter("supplierID"));

supplier.setSupplierName(request.getParameter("supplierName"));

supplier.setContactName(request.getParameter("contactName"));

supplier.setPhoneNumber(request.getParameter("phoneNumber"));

supplier.setEmail(request.getParameter("email"));

supplier.setAddress(request.getParameter("address"));

supplierDAO.addSupplier(supplier);

response.sendRedirect("MainController?action=ViewSuppliers");

} else if ("UpdateSupplier".equals(action)) {

Supplier supplier = new Supplier();

supplier.setSupplierID(request.getParameter("supplierID"));

supplier.setSupplierName(request.getParameter("supplierName"));

supplier.setContactName(request.getParameter("contactName"));

supplier.setPhoneNumber(request.getParameter("phoneNumber"));

supplier.setEmail(request.getParameter("email"));

supplier.setAddress(request.getParameter("address"));

supplierDAO.updateSupplier(supplier);

response.sendRedirect("MainController?action=ViewSuppliers");

} else if ("DeleteSupplier".equals(action)) {

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

supplierDAO.deleteSupplier(supplierID);

response.sendRedirect("MainController?action=ViewSuppliers");

}

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("ERROR", "Supplier operation failed. Please try again.");

request.getRequestDispatcher("supplierList.jsp").forward(request, response);

}

}

}

**5. CartController.java**

package com.example.controller;

import com.example.dao.CartDAO;

import com.example.dto.CartItem;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

import java.util.List;

@WebServlet("/CartController")

public class CartController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

HttpSession session = request.getSession();

CartDAO cartDAO = new CartDAO();

try {

if ("AddToCart".equals(action)) {

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

int quantity = Integer.parseInt(request.getParameter("quantity"));

cartDAO.addToCart(session, productID, quantity);

response.sendRedirect("MainController?action=ViewCart");

} else if ("UpdateCart".equals(action)) {

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

int quantity = Integer.parseInt(request.getParameter("quantity"));

cartDAO.updateCart(session, productID, quantity);

response.sendRedirect("MainController?action=ViewCart");

} else if ("RemoveFromCart".equals(action)) {

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

cartDAO.removeFromCart(session, productID);

response.sendRedirect("MainController?action=ViewCart");

}

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("ERROR", "Cart operation failed. Please try again.");

request.getRequestDispatcher("cart.jsp").forward(request, response);

}

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session = request.getSession();

CartDAO cartDAO = new CartDAO();

try {

List<CartItem> cartItems = cartDAO.getCartItems(session);

request.setAttribute("cartItems", cartItems);

request.getRequestDispatcher("cart.jsp").forward(request, response);

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("ERROR", "Failed to retrieve cart items. Please try again.");

request.getRequestDispatcher("cart.jsp").forward(request, response);

}

}

}

**6. OrderController.java**

package com.example.controller;

import com.example.dao.OrderDAO;

import com.example.dao.CartDAO;

import com.example.dto.CartItem;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

import java.util.List;

@WebServlet("/OrderController")

public class OrderController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

HttpSession session = request.getSession();

OrderDAO orderDAO = new OrderDAO();

CartDAO cartDAO = new CartDAO();

try {

if ("Checkout".equals(action)) {

List<CartItem> cartItems = cartDAO.getCartItems(session);

request.setAttribute("cartItems", cartItems);

request.getRequestDispatcher("checkout.jsp").forward(request, response);

} else if ("PlaceOrder".equals(action)) {

String userID = (String) session.getAttribute("userID");

double totalAmount = (double) request.getAttribute("totalAmount");

orderDAO.placeOrder(userID, totalAmount, session);

session.removeAttribute("cartItems"); // Clear the cart after placing order

response.sendRedirect("orderConfirmation.jsp");

}

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("ERROR", "Order processing failed. Please try again.");

request.getRequestDispatcher("checkout.jsp").forward(request, response);

}

}

}

**6. JSP Pages**

Create the corresponding JSP pages for each functionality:

1. **login.jsp**
2. **register.jsp**
3. **productList.jsp**
4. **createProduct.jsp**
5. **updateProduct.jsp**
6. **supplierList.jsp**
7. **cart.jsp**
8. **checkout.jsp**

These JSP files will render the views for each user action and handle form submissions that interact with the Main Controller.

**1. login.jsp**

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

<html>

<head>

<title>Login</title>

</head>

<body>

<h2>Login</h2>

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

<input type="hidden" name="action" value="Login">

<label for="userID">User ID:</label>

<input type="text" name="userID" required><br><br>

<label for="password">Password:</label>

<input type="password" name="password" required><br><br>

<input type="submit" value="Login">

</form>

<%

String error = (String) request.getAttribute("ERROR");

if (error != null) {

%>

<p style="color:red;"><%= error %></p>

<%

}

%>

</body>

</html>

**2. register.jsp**

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

<html>

<head>

<title>User Registration</title>

</head>

<body>

<h2>User Registration</h2>

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

<input type="hidden" name="action" value="Register">

<label for="fullName">Full Name:</label>

<input type="text" name="fullName" required><br><br>

<label for="email">Email:</label>

<input type="email" name="email" required><br><br>

<label for="phoneNumber">Phone Number:</label>

<input type="text" name="phoneNumber" required><br><br>

<label for="password">Password:</label>

<input type="password" name="password" required><br><br>

<input type="submit" value="Register">

</form>

<%

String error = (String) request.getAttribute("ERROR");

if (error != null) {

%>

<p style="color:red;"><%= error %></p>

<%

}

%>

</body>

</html>

**3. productList.jsp**

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

<%@ page import="java.util.List" %>

<%@ page import="com.example.dto.Product" %>

<html>

<head>

<title>Product List</title>

</head>

<body>

<h2>Product List</h2>

<table border="1">

<tr>

<th>Product ID</th>

<th>Name</th>

<th>Description</th>

<th>Price</th>

<th>Quantity</th>

<th>Category</th>

<th>Actions</th>

</tr>

<%

List<Product> products = (List<Product>) request.getAttribute("products");

if (products != null) {

for (Product product : products) {

%>

<tr>

<td><%= product.getProductID() %></td>

<td><%= product.getProductName() %></td>

<td><%= product.getDescription() %></td>

<td><%= product.getPrice() %></td>

<td><%= product.getQuantity() %></td>

<td><%= product.getCategory() %></td>

<td>

<form action="MainController" method="post" style="display:inline;">

<input type="hidden" name="action" value="UpdateProduct">

<input type="hidden" name="productID" value="<%= product.getProductID() %>">

<input type="submit" value="Edit">

</form>

<form action="MainController" method="post" style="display:inline;">

<input type="hidden" name="action" value="DeleteProduct">

<input type="hidden" name="productID" value="<%= product.getProductID() %>">

<input type="submit" value="Delete">

</form>

</td>

</tr>

<%

}

}

%>

</table>

<a href="dashboard.jsp">Back to Dashboard</a>

</body>

</html>

**4. createProduct.jsp**

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

<%@ page import="javax.servlet.http.HttpSession" %>

<html>

<head>

<title>Create Product</title>

</head>

<body>

<%

HttpSession session = request.getSession(false);

if (session == null || session.getAttribute("userID") == null) {

response.sendRedirect("login.jsp");

}

%>

<h2>Create a New Product</h2>

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

<input type="hidden" name="action" value="CreateProduct">

<label for="productID">Product ID:</label>

<input type="text" name="productID" required><br><br>

<label for="productName">Product Name:</label>

<input type="text" name="productName" required><br><br>

<label for="description">Description:</label>

<textarea name="description" rows="5" cols="50" required></textarea><br><br>

<label for="price">Price:</label>

<input type="number" step="0.01" name="price" required><br><br>

<label for="quantity">Quantity:</label>

<input type="number" name="quantity" required><br><br>

<label for="category">Category:</label>

<input type="text" name="category" required><br><br>

<label for="supplierID">Supplier ID:</label>

<input type="text" name="supplierID" required><br><br>

<input type="submit" value="Create Product">

</form>

<%

String error = (String) request.getAttribute("ERROR");

if (error != null) {

%>

<p style="color:red;"><%= error %></p>

<%

}

%>

</body>

</html>

**5. updateProduct.jsp**

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

<%@ page import="javax.servlet.http.HttpSession" %>

<%@ page import="com.example.dto.Product" %>

<html>

<head>

<title>Update Product</title>

</head>

<body>

<%

HttpSession session = request.getSession(false);

if (session == null || session.getAttribute("userID") == null) {

response.sendRedirect("login.jsp");

}

Product product = (Product) request.getAttribute("product");

%>

<h2>Update Product: <%= product.getProductName() %></h2>

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

<input type="hidden" name="action" value="UpdateProduct">

<input type="hidden" name="productID" value="<%= product.getProductID() %>">

<label for="productName">Product Name:</label>

<input type="text" name="productName" value="<%= product.getProductName() %>" required><br><br>

<label for="description">Description:</label>

<textarea name="description" rows="5" cols="50" required><%= product.getDescription() %></textarea><br><br>

<label for="price">Price:</label>

<input type="number" step="0.01" name="price" value="<%= product.getPrice() %>" required><br><br>

<label for="quantity">Quantity:</label>

<input type="number" name="quantity" value="<%= product.getQuantity() %>" required><br><br>

<label for="category">Category:</label>

<input type="text" name="category" value="<%= product.getCategory() %>" required><br><br>

<label for="supplierID">Supplier ID:</label>

<input type="text" name="supplierID" value="<%= product.getSupplierID() %>" required><br><br>

<input type="submit" value="Update Product">

</form>

<%

String error = (String) request.getAttribute("ERROR");

if (error != null) {

%>

<p style="color:red;"><%= error %></p>

<%

}

%>

<a href="productList.jsp">Back to Product List</a>

</body>

</html>

**6. supplierList.jsp**

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

<%@ page import="java.util.List" %>

<%@ page import="com.example.dto.Supplier" %>

<html>

<head>

<title>Supplier List</title>

</head>

<body>

<h2>Supplier List</h2>

<table border="1">

<tr>

<th>Supplier ID</th>

<th>Name</th>

<th>Contact Name</th>

<th>Phone Number</th>

<th>Email</th>

<th>Address</th>

<th>Actions</th>

</tr>

<%

List<Supplier> suppliers = (List<Supplier>) request.getAttribute("suppliers");

if (suppliers != null) {

for (Supplier supplier : suppliers) {

%>

<tr>

<td><%= supplier.getSupplierID() %></td>

<td><%= supplier.getSupplierName() %></td>

<td><%= supplier.getContactName() %></td>

<td><%= supplier.getPhoneNumber() %></td>

<td><%= supplier.getEmail() %></td>

<td><%= supplier.getAddress() %></td>

<td>

<form action="MainController" method="post" style="display:inline;">

<input type="hidden" name="action" value="UpdateSupplier">

<input type="hidden" name="supplierID" value="<%= supplier.getSupplierID() %>">

<input type="submit" value="Edit">

</form>

<form action="MainController" method="post" style="display:inline;">

<input type="hidden" name="action" value="DeleteSupplier">

<input type="hidden" name="supplierID" value="<%= supplier.getSupplierID() %>">

<input type="submit" value="Delete">

</form>

</td>

</tr>

<%

}

}

%>

</table>

<a href="dashboard.jsp">Back to Dashboard</a>

</body>

</html>

**7. cart.jsp**

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

<%@ page import="java.util.List" %>

<%@ page import="com.example.dto.CartItem" %>

<html>

<head>

<title>Shopping Cart</title>

</head>

<body>

<h2>Your Shopping Cart</h2>

<table border="1">

<tr>

<th>Product ID</th>

<th>Name</th>

<th>Quantity</th>

<th>Price</th>

<th>Total</th>

<th>Actions</th>

</tr>

<%

List<CartItem> cartItems = (List<CartItem>) request.getAttribute("cartItems");

if (cartItems != null) {

for (CartItem item : cartItems) {

%>

<tr>

<td><%= item.getProductID() %></td>

<td><%= item.getProductName() %></td>

<td><%= item.getQuantity() %></td>

<td><%= item.getPrice() %></td>

<td><%= item.getTotal() %></td>

<td>

<form action="MainController" method="post" style="display:inline;">

<input type="hidden" name="action" value="UpdateCart">

<input type="hidden" name="productID" value="<%= item.getProductID() %>">

<input type="number" name="quantity" value="<%= item.getQuantity() %>" min="1">

<input type="submit" value="Update">

</form>

<form action="MainController" method="post" style="display:inline;">

<input type="hidden" name="action" value="RemoveFromCart">

<input type="hidden" name="productID" value="<%= item.getProductID() %>">

<input type="submit" value="Remove">

</form>

</td>

</tr>

<%

}

}

%>

</table>

<a href="productList.jsp">Continue Shopping</a>

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

<input type="hidden" name="action" value="Checkout">

<input type="submit" value="Checkout">

</form>

</body>

</html>

**8. checkout.jsp**

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

<%@ page import="java.util.List" %>

<%@ page import="com.example.dto.CartItem" %>

<html>

<head>

<title>Checkout</title>

</head>

<body>

<h2>Checkout</h2>

<table border="1">

<tr>

<th>Product ID</th>

<th>Name</th>

<th>Quantity</th>

<th>Price</th>

<th>Total</th>

</tr>

<%

List<CartItem> cartItems = (List<CartItem>) request.getAttribute("cartItems");

if (cartItems != null) {

for (CartItem item : cartItems) {

%>

<tr>

<td><%= item.getProductID() %></td>

<td><%= item.getProductName() %></td>

<td><%= item.getQuantity() %></td>

<td><%= item.getPrice() %></td>

<td><%= item.getTotal() %></td>

</tr>

<%

}

}

%>

</table>

<h3>Total Amount: <%= request.getAttribute("totalAmount") %></h3>

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

<input type="hidden" name="action" value="PlaceOrder">

<input type="submit" value="Confirm Order">

</form>

</body>

</html>