1. AdminValidationServlet

package com.db.controler;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/adminValidationServlet")

public class AdminValidationServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

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

PrintWriter out=response.getWriter();

response.setContentType("text/html");

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

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

if(name.equals("Anil") && pwd.equals("food")){

RequestDispatcher rd=request.getRequestDispatcher("item-list.jsp");

rd.forward(request, response);

}

else{

out.println("<p>Invalid User Please Register First.</p>");

RequestDispatcher rd1=request.getRequestDispatcher("AdminLogin.html");

rd1.forward(request, response);

}

}

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

doGet(request, response);

}

}

2. ItemServlet

package com.db.controler;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.db.utils.ItemDAO;

import demo.Items;

@WebServlet("/")

public class ItemServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private ItemDAO itemDAO;

public void init() {

itemDAO = new ItemDAO();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doGet(request, response);

}

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

String action = request.getServletPath();

try {

switch (action) {

case "/new":

showNewForm(request, response);

break;

case "/insert":

insertItem(request, response);

break;

case "/delete":

deleteItem(request, response);

break;

case "/edit":

showEditForm(request, response);

break;

case "/update":

updateItem(request, response);

break;

default:

listItem(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void listItem(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List < Items > listItem =itemDAO.selectAllItems() ;

request.setAttribute("listItem", listItem);

RequestDispatcher dispatcher = request.getRequestDispatcher("item-list.jsp");

dispatcher.forward(request, response);

}

private void showNewForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("item-form.jsp");

dispatcher.forward(request, response);

}

private void showEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Items existingItem =itemDAO.selectItem(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("item-form.jsp");

request.setAttribute("item", existingItem);

dispatcher.forward(request, response);

}

private void insertItem(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

Items newItem = new Items(id,name, rupee);

itemDAO.insertItem(newItem);

response.sendRedirect("list");

}

private void updateItem(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

Items book = new Items(id, name, rupee);

itemDAO.updateItem(book);

response.sendRedirect("list");

}

private void deleteItem(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

itemDAO.deleteItem(id);

response.sendRedirect("list");

}

}

3. DbConnection:

package com.db.utils;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DbConnection {

/\*

final static String DB\_NAME = "Food\_Boxs";

final static String CONNECTION\_URL = "jdbc:mysql://localhost:3306/" + DB\_NAME;

final static String USERNAME = "root";

final static String PASSWORD = "P12p11@0467";

\*/

public static Connection init() {

Connection con=null;

try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Food\_Boxs","root","P12p11@0467");

}

catch(ClassNotFoundException | SQLException e){

e.printStackTrace();

}//catch

return con;

}

}

4.ItemsDAO:

package com.db.utils;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import demo.Items;

public class ItemDAO {

private static final String INSERT\_ITEMS\_SQL = "INSERT INTO items VALUES(?, ?, ?)";

private static final String SELECT\_ITEMS\_BY\_ID = "select id,item\_name,price\_rupee from items where id =?";

private static final String SELECT\_ALL\_ITEMS = "select \* from items";

private static final String DELETE\_ITEMS\_SQL = "delete from items where id = ?";

private static final String UPDATE\_ITEMS\_SQL = "update items set item\_name = ?,price\_rupee= ? where id = ?";

public void insertItem(Items item) throws SQLException {

// try-with-resource statement will auto close the connection.

try (Connection connection = DbConnection.init(); PreparedStatement preparedStatement = connection.prepareStatement(INSERT\_ITEMS\_SQL)) {

preparedStatement.setInt(1, item.getId());

preparedStatement.setString(2, item.getName());

preparedStatement.setInt(3, item.getPrice());

preparedStatement.executeUpdate();

} catch (SQLException e) {

printSQLException(e);

}

}

public Items selectItem(int id) {

Items item = null;

// Step 1: Establishing a Connection

try (Connection connection = DbConnection.init();

// Step 2:Create a statement using connection object

PreparedStatement preparedStatement = connection.prepareStatement(SELECT\_ITEMS\_BY\_ID);) {

preparedStatement.setInt(1, id);

System.out.println(preparedStatement);

// Step 3: Execute the query or update query

ResultSet rs = preparedStatement.executeQuery();

// Step 4: Process the ResultSet object.

while (rs.next()) {

String name = rs.getString("item\_name");

int rupee = rs.getInt("price\_rupee");

item = new Items(id, name,rupee);

}

} catch (SQLException e) {

printSQLException(e);

}

return item;

}

public List < Items > selectAllItems() {

// using try-with-resources to avoid closing resources (boiler plate code)

List < Items > items = new ArrayList < > ();

// Step 1: Establishing a Connection

try (Connection connection = DbConnection.init();

// Step 2:Create a statement using connection object

PreparedStatement preparedStatement = connection.prepareStatement(SELECT\_ALL\_ITEMS);) {

// Step 3: Execute the query or update query

ResultSet rs = preparedStatement.executeQuery();

// Step 4: Process the ResultSet object.

while (rs.next()) {

int id = rs.getInt("id");

String name = rs.getString("item\_name");

int rupee = rs.getInt("price\_rupee");

items.add(new Items(id, name, rupee));

}

} catch (SQLException e) {

printSQLException(e);

}

return items;

}

public boolean deleteItem(int id) throws SQLException {

boolean rowDeleted;

try (Connection connection = DbConnection.init(); PreparedStatement statement = connection.prepareStatement(DELETE\_ITEMS\_SQL);) {

statement.setInt(1, id);

rowDeleted = statement.executeUpdate() > 0;

}

return rowDeleted;

}

public boolean updateItem(Items item) throws SQLException {

boolean rowUpdated;

try (Connection connection = DbConnection.init(); PreparedStatement statement = connection.prepareStatement(UPDATE\_ITEMS\_SQL);) {

statement.setString(1,item.getName());

statement.setInt(2,item.getPrice());

statement.setInt(3, item.getId());

rowUpdated = statement.executeUpdate() > 0;

}

return rowUpdated;

}

private void printSQLException(SQLException ex) {

for (Throwable e: ex) {

if (e instanceof SQLException) {

e.printStackTrace(System.err);

System.err.println("SQLState: " + ((SQLException) e).getSQLState());

System.err.println("Error Code: " + ((SQLException) e).getErrorCode());

System.err.println("Message: " + e.getMessage());

Throwable t = ex.getCause();

while (t != null) {

System.out.println("Cause: " + t);

t = t.getCause();

}

}

}

}

}

5.Items:

**package** demo;

**public** **class** Items {

**protected** **int** id;

**protected** String name;

**protected** **int** price;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getPrice() {

**return** price;

}

**public** **void** setPrice(**int** price) {

**this**.price = price;

}

**public** Items(**int** id, String name, **int** price) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.price = price;

}

**public** Items() {

**super**();

}

}

6. MenuItems:

**package** demo;

**import** java.util.ArrayList;

**public** **class** MenuItems {

**private** **int** price;

String item\_name;

MenuItems(**int** price,String item\_name)

{

**this**.price = price;

**this**.item\_name = item\_name;

}

**public** **void** Add\_items(**int** price,String name){

}

}

7.adminLogin:

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin login page</title>

<link rel=*"stylesheet"* href=*"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"*>

<script src=*'https://kit.fontawesome.com/a076d05399.js'*></script>

<link rel=*"stylesheet"* href=*"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"* integrity=*"sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"* crossorigin=*"anonymous"*>

</head>

<body>

<header >

<nav class=*"navbar navbar-expand-md navbar-dark"* style="background-color: *tomato*">

<div>

<a href=*""* class=*"navbar-brand"*>Food Box</a>

</div>

<ul class=*"navbar-nav"*>

<li><a href=*"#"* class=*"nav-link"*>Locations</a></li>

<li><a href=*"#"* class=*"nav-link"*>Our chefs</a></li>

<li><a href=*"#"* class=*"nav-link"*>About us</a></li>

</ul>

</nav>

</header>

<div class=*"container col-md-20"*>

<h4>Admin Login:</h4>

<div class=*"card"*>

<div class=*"card-body"*>

<form action=*"adminValidationServlet"* method=*"get"*>

<fieldset class=*"form-group"*>

<label><i class=*"fa fa-user"*></i>&nbsp&nbspUser Name :</label>

<input type=*"text"* class=*"form-control"* name=*"uname"* placeholder=*"Enter your User Name"* value=*""* required=*"required"*>

</fieldset>

<fieldset class=*"form-group"*>

<label><i class=*"fas fa-unlock-alt"*></i>&nbsp&nbspPassword :</label>

<input type=*"password"* class=*"form-control"* name=*"pwd"* placeholder=*"Enter the passowrd"* value=*""* required=*"required"*>

</fieldset>

<button type=*"submit"* class=*"btn btn-success"*>Sign In</button>

</form>

</div>

</div>

</div>

</body>

<

</body>

</html>

8.check:

<%@page import=*"com.db.utils.DbConnection"*%>

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

<%

String user = request.getParameter("uname");

String pwd = request.getParameter("pass");

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

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

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

**try** {

Connection con = DbConnection.init();

Statement st = con.createStatement();

String q;

ResultSet rs;

**int** i = st.executeUpdate("insert into members(first\_name, last\_name, email, uname, password) values ('" + fname + "','" + lname + "','" + email + "','" + user + "','" + pwd + "')");

**if** (i > 0) {

session.setAttribute("userid", user);

response.sendRedirect("login.jsp");

} **else** {

response.sendRedirect("index.jsp");

}

} **catch** (SQLException se) {

//Handle errors for JDBC

se.printStackTrace();

} **catch** (Exception e) {

//Handle errors for Class.forName

e.printStackTrace();

}

%>

9.error:

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"* isErrorPage=*"true"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Error</title>

</head>

<body>

<center>

<h1>Error</h1>

<h2><%=exception.getMessage() %><br/> </h2>

</center>

</body>

</html>

10.index:

<%@page contentType=*"text/html"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Food Shop</title>

<link rel=*"stylesheet"* href=*"w3.css"* type=*"text/css"*>

<link rel=*"stylesheet"* href=*"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"*>

<script src=*'https://kit.fontawesome.com/a076d05399.js'*></script>

<link rel=*"stylesheet"* href=*"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"* integrity=*"sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"* crossorigin=*"anonymous"*>

</head>

<style>

**body** {

background-image: *url("https://png.pngtree.com/thumb\_back/fh260/back\_our/20190621/ourmid/pngtree-light-fresh-and-fresh-literary-food-banner-background-image\_184091.jpg")*;

background-repeat: *no-repeat*;

height: *1000px*;

background-position: *center*;

background-repeat: *no-repeat*;

background-size: *cover*;

position: *relative*;

}

</style>

<body>

<header >

<nav class=*"navbar navbar-expand-md navbar-dark"* style="background-color: *tomato*">

<div>

<a href=*""* class=*"navbar-brand"*>Food Box</a>

</div>

<ul class=*"navbar-nav"*>

<li><a href=*"#"* class=*"nav-link"*>Locations</a></li>

<li><a href=*"#"* class=*"nav-link"*>Our chefs</a></li>

<li><a href=*"#"* class=*"nav-link"*>About</a></li>

<li><a href=*"register.jsp"* class=*"nav-link"*>Registration Page</a></li>

<li><a href=*"adminLogin.jsp"* class=*"nav-link"*>Admin</a></li>

</ul>

</nav>

</header>

<br><br>

<div class=*"container"*>

<div class=*"card"* style="width: *18rem*;">

<div class=*"card-body"*>

<form method=*"post"* action=*"login.jsp"*>

<fieldset class=*"form-group"*>

<label><i class=*"fa fa-user"*></i>&nbsp&nbspUser Name :</label>

<input type=*"text"* class=*"form-control"* name=*"uname"* placeholder=*"Enter your User Name"* value=*""* required=*"required"*>

</fieldset>

<fieldset class=*"form-group"*>

<label><i class=*"fas fa-unlock-alt"*></i>&nbsp&nbspPassword :</label>

<input type=*"password"* class=*"form-control"* name=*"pass"* placeholder=*"Enter the passowrd"* value=*""* required=*"required"*>

</fieldset>

<button type=*"submit"* class=*"btn btn-success"*>Sign In</button>

<fieldset class=*"form-group"*>

<label>Create an account :</label>

<a href=*"register.jsp"*><input type=*"button"* class=*"btn btn-success"* value=*"Sign Up"*></a>

</fieldset></div>

</form>

</div>

</div>

</div>

<div >

<div style="text-align:*center*"> <br><h4></h4></div>

</div>

</body>

</html>