

**CSE 3023**

WEB-BASED APPLICATION DEVELOPMENT

**LAB WEEK 8**

MVC EXAMPLE WITH SERVLETS AND JSP

**LAB DEMONSTRATOR**

SIR ARIZAL SHAMSIL

**PREPARED BY**

|  |  |
| --- | --- |
| **NAME** | **MATRIC NO.** |
| NUR HIDAYATUL AIN BINTI HALUWI | S62367 |

FACULTY OF OCEAN ENGINEERING TECHNOLOGY AND INFORMATICS

UNIVERSITI MALAYSIA TERENGGANU

# **Creating MVC Database Web Application in JSP and Servlets – For Create, Read, Update & Delete**

Running the program and perform CRUD process.

## Step 1: Creating table employees in Company database schema

Graphical user interface, text, application, Word

Description automatically generated

## Step 2: Create new web application project, named as Employee\_Management.

Graphical user interface, application

Description automatically generated

## Step 3: Create three Java class that representing:

* EmployeeDAO.java (act as a Data Access Object (DAO) and to open /close database connection)
* Employee.java (act as a JavaBeans to represent business object), and
* EmployeeServlet.java (act to perform CRUD process)

Graphical user interface, text, application

Description automatically generated

## Step 4: Create files

Graphical user interface, text

Description automatically generated with medium confidence

## Step 5: Add libraries

Graphical user interface, text, application

Description automatically generated

Final schema of the project:

A picture containing text

Description automatically generated

## Step 6: Running the program and try CRUD process:

1. Run index.jsp page.

Graphical user interface, text, application

Description automatically generated

1. Click List All Employee List button to show all records.

Graphical user interface, text, application

Description automatically generated

1. Click Add New Employee button to create new record.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

1. Click hyperlink Edit to do edit/update an existing record.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

The changes are updated for user with ID = 7.

1. Click hyperlink Delete to do delete an existing record.

Graphical user interface, table

Description automatically generated

The record with ID = 9 is no longer in the list after being deleted.

## Source Codes

1. EmployeeDAO.java

package com.DAO;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import com.Model.Employee;

/\*\*

\*

\* @author

\*/

public class EmployeeDAO {

private String jdbcURL = "jdbc:mysql://localhost:3306/company";

private String jdbcUsername = "root";

private String jdbcPassword = "admin";

private static final String INSERT\_EMPLOYEES\_SQL = "INSERT INTO employees (name, email, position) VALUES (?, ?, ?);";

private static final String SELECT\_EMPLOYEE\_BY\_ID = "SELECT id, name, email, position FROM employees where id=?";

private static final String SELECT\_ALL\_EMPLOYEES = "SELECT \* FROM employees";

private static final String DELETE\_EMPLOYEES\_SQL = "DELETE FROM employees where id=?";

private static final String UPDATE\_EMPLOYEES\_SQL = "UPDATE employees set name=?, email=?, position=? where id=?";

public EmployeeDAO(){}

protected Connection getConnection(){

Connection connection = null;

try{

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

connection = DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);

} catch (SQLException e){

e.printStackTrace();

} catch (ClassNotFoundException e){

e.printStackTrace();

}

return connection;

}

public void insertEmployee(Employee employee) throws SQLException{

System.out.println(INSERT\_EMPLOYEES\_SQL);

try (Connection connection = getConnection();

PreparedStatement preparedStatement =

connection.prepareStatement(INSERT\_EMPLOYEES\_SQL)){

preparedStatement.setString(1, employee.getName());

preparedStatement.setString(2, employee.getEmail());

preparedStatement.setString(3, employee.getPosition());

System.out.println(preparedStatement);

preparedStatement.executeUpdate();

} catch (SQLException e){

printSQLException(e);

}

}

public Employee selectEmployee(int id){

Employee employee = null;

try(Connection connection = getConnection();

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

preparedStatement.setInt(1, id);

System.out.println(preparedStatement);

ResultSet rs = preparedStatement.executeQuery();

while (rs.next()){

String name = rs.getString("name");

String email = rs.getString("email");

String position = rs.getString("position");

employee = new Employee(id, name, email, position);

}

}

catch (SQLException e){

printSQLException(e);

}

return employee;

}

public List <Employee> selectAllEmployees(){

List <Employee> employees = new ArrayList<>();

try (Connection connection = getConnection();

PreparedStatement preparedStatement =

connection.prepareStatement(SELECT\_ALL\_EMPLOYEES);){

System.out.print(preparedStatement);

ResultSet rs = preparedStatement.executeQuery();

while (rs.next()){

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

String name = rs.getString("name");

String email = rs.getString("email");

String position = rs.getString("position");

employees.add(new Employee(id, name, email, position));

}

}catch(SQLException e){

printSQLException(e);

}

return employees;

}

public boolean deleteEmployee (int id) throws SQLException{

boolean rowDeleted;

try(Connection connection = getConnection();

PreparedStatement statement =

connection.prepareStatement(DELETE\_EMPLOYEES\_SQL);){

statement.setInt(1, id);

rowDeleted = statement.executeUpdate() > 0;

}

return rowDeleted;

}

public boolean updateEmployee(Employee employee) throws SQLException{

boolean rowUpdated;

try(Connection connection = getConnection();

PreparedStatement statement =

connection.prepareStatement(UPDATE\_EMPLOYEES\_SQL);){

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

statement.setString(2, employee.getEmail());

statement.setString(3, employee.getPosition());

statement.setInt(4, employee.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();

}

}

}

}

}

/\*\*

\* Handles the HTTP <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

User user = new User();

user.setUserid(request.getParameter("userid"));

user.setFirstname(request.getParameter("firstname"));

user.setLastname(request.getParameter("lastname"));

if (action.equalsIgnoreCase("edit")){

try {

dao.updateUser(user);

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null, ex);

}

}else{

try {

dao.addUser(user);

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null, ex);

}

}

RequestDispatcher view = request.getRequestDispatcher(LIST\_USER);

try {

request.setAttribute("users", dao.getAllUsers());

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null, ex);

}

view.forward(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

1. Employee.java

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.Model;

/\*\*

\*

\* @author

\*/

public class Employee {

private int id;

private String name;

private String email;

private String position;

public Employee(int id, String name, String email, String position) {

super();

this.id = id;

this.name = name;

this.email = email;

this.position = position;

}

public Employee(String name, String email, String position) {

super();

this.name = name;

this.email = email;

this.position = position;

}

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 String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPosition() {

return position;

}

public void setPosition(String position) {

this.position = position;

}

}

1. EmployeeServlet.java

package com.Web;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import com.DAO.EmployeeDAO;

import com.Model.Employee;

/\*\*

\*

\* @author

\*/

@WebServlet("/")

public class EmployeeServlet extends HttpServlet {

private EmployeeDAO employeeDAO;

public void init(){

employeeDAO = new EmployeeDAO();

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doGet(request, response);

}

@Override

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":

insertEmployee(request, response);

break;

case "/delete":

deleteEmployee(request, response);

break;

case "/edit":

showEditForm(request, response);

break;

case "/update":

updateEmployee(request, response);

break;

default:

listEmployee(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void listEmployee(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException{

List <Employee> listEmployee = employeeDAO.selectAllEmployees();

request.setAttribute("listEmployee", listEmployee);

RequestDispatcher dispatcher = request.getRequestDispatcher("employeeList.jsp");

dispatcher.forward(request, response);

}

private void showNewForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException{

RequestDispatcher dispatcher = request.getRequestDispatcher("employeeForm.jsp");

dispatcher.forward(request, response);

}

private void showEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException{

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

Employee existingEmployee = employeeDAO.selectEmployee(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("employeeForm.jsp");

request.setAttribute("employee", existingEmployee);

dispatcher.forward(request, response);

}

private void insertEmployee(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException{

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

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

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

Employee newEmployee = new Employee(name, email, position);

employeeDAO.insertEmployee(newEmployee);

response.sendRedirect("list");

}

private void updateEmployee(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException{

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

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

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

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

Employee employee = new Employee(id, name, email, position);

employeeDAO.updateEmployee(employee);

response.sendRedirect("list");

}

private void deleteEmployee(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException{

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

employeeDAO.deleteEmployee(id);

response.sendRedirect("list");

}

}

1. employeeForm.jsp

<%--

Document : employeeForm

--%>

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

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

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

<title>Employee Management Application</title>

<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">Employee Management App</a>

</div>

<ul class="navbar-nav">

<li><a href="<%=request.getContextPath()%>/list" class="nav-link">Employees</a></li>

</ul>

</nav>

</header>

<br>

<div class="container col-md-5">

<div class="card">

<div class="card-body">

<c:if test="${employee != null}">

<form action="update" method="POST">

</c:if>

<c:if test="${employee == null}">

<form action="insert" method="POST">

</c:if>

<h2>

<c:if test="${employee != null}">

Edit Employee

</c:if>

<c:if test="${employee == null}">

Add New Employee

</c:if>

</h2>

<c:if test="${employee != null}">

<input type="hidden" name="id" value="<c:out value='${employee.id}'/>">

</c:if>

<fieldset class="form-group">

<label>Employee Name</label>

<input type="text" value="<c:out value='${employee.name}'/>"

class="form-control" name="name" required="required">

</fieldset>

<fieldset class="form-group">

<label>Employee Email</label>

<input type="text" value="<c:out value='${employee.email}'/>"

class="form-control" name="email">

</fieldset>

<fieldset class="form-group">

<label>Employee Position</label>

<input type="text" value="<c:out value='${employee.position}'/>" class="form-control" readonly>

<input list="positionList" id="position" class="form-control" name="position">

<datalist id="positionList">

<option value="Manager"></option>

<option value="Head of Dept"></option>

<option value="Supervisor"></option>

<option value="Director"></option>

</datalist>

</fieldset>

<button type="submit" class="btn btn-success">Save</button>

</form>

</div>

</div>

</div>

</body>

</html>

1. employeeList.jsp

<%--

Document : employeeList

--%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

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

<!DOCTYPE html>

<html>

<head>

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

<title>Employee Management Application</title>

<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">Employee Management App</a>

</div>

<ul class="navbar-nav">

<li><a href="<%=request.getContextPath()%>/list">Employees</a></li>

</ul>

</nav>

</header>

<br>

<div class="row">

<div class="container">

<h3 class="text-center">List of Employees</h3>

<hr>

<div class="container text-left">

<a href="<%=request.getContextPath()%>/new" class="btn btn-success">Add New Employee</a>

</div>

<br>

<table class="table table-bordered">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Position</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<c:forEach var="employee" items="${listEmployee}">

<tr>

<td>

<c:out value="${employee.id}" />

</td>

<td>

<c:out value="${employee.name}" />

</td>

<td>

<c:out value="${employee.email}" />

</td>

<td>

<c:out value="${employee.position}" />

</td>

<td>

<a href="edit?id=<c:out value='${employee.id}' />">Edit</a> &nbsp;&nbsp;&nbsp;&nbsp;

<a href="delete?id=<c:out value='${employee.id}' />">Delete</a>

</td>

</tr>

</c:forEach>

</tbody>

</table>

</div>

</div>

</body>

</html>

1. error.jsp

<%--

Document : error

--%>

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

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

<html>

<head>

<title>Error Page</title>

</head>

<body>

<h1>Error</h1>

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

</body>

</html>

1. index.jsp

<%--

Document : index

--%>

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

<!DOCTYPE html>

<html>

<head>

<title>User Management Application</title>

<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>

<h1>Application MVC system for Employee Management</h1>

<ul>

<li><a href="http://localhost:8080/Employee\_Management/list">All Employee List</a></li>

<li><a href="http://localhost:8080/Employee\_Management/new">Add a New Employee </a></li>

<li><a href="http://localhost:8080/Employee\_Management/list">Edit Employee </a></li>

</ul>

</body>

</html>