



CSM3023 WEB BASED APPLICATION DEVELOPMENT (K1)

**BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH
HONORS**

SEMESTER II 2023/2024

LAB 8 – An MVC Example with Servlets and JSP

Prepared for:

DR. MOHAMAD NOR HASSAN

Prepared by:

MUHAMMAD IZZUL WAFIY BIN IZAM (S65466)

Coding:

Employee.java

```
package com.Model;

/**
 *
 * @author USER
 */
public class Employee {
    protected int id;
    protected String name;
    protected String email;
    protected String position;

    public Employee() {}

    public Employee(String name, String email, String position) {
        super();
        this.name = name;
        this.email = email;
        this.position = position;
    }

    public Employee(int id, String name, String email, String position) {
        this.id = id;
        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;
    }
}
```

EmployeeDAO.java

```
package com.DAO;

/**
 * @author USER
 */
import java.sql.*;
import java.sql.SQLException;
import java.util.*;
import com.Model.Employee;

public class EmployeeDAO {
    Connection connection = null;
    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.cj.jdbc.Driver");
            connection = DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);
            System.out.println("Database connected!");
        } 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;
        // Step 1: Establishing a Connection
        try (Connection connection = getConnection();
            // Step 2: Create a statement using connection
            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.println(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();
            }
        }
    }
}
}

```

EmployeeServlet.java

```

import com.DAO.EmployeeDAO;
import com.Model.Employee;
import jakarta.servlet.RequestDispatcher;
import java.io.IOException;
import java.io.PrintWriter;
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 java.sql.SQLException;
import java.util.List;

/**
 *
 * @author Lenovo
 */
@WebServlet("/")
public class EmployeeServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    private EmployeeDAO employeeDAO;

    @Override
    public void init() {
        employeeDAO = new EmployeeDAO();
    }

    

HttpServlet methods. Click on the + sign on the left to edit the code.


}

```

EmployeeForm.jsp

EmployeeList.jsp

```
<!--
Document   : EmployeeList
Created on : 6 Jun 2024, 12:31:25 am
Author    : USER
-->

<${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 System</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQVtLW8v39SpjRb9jT0oq64lcV8gW7s60vcR40XWQ7a9vwbKa94T94" 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>
<hr>

<div class="row">
<div class="container">
<h3 class="container">List of Employees</h3>
<hr>
<div class="container text-left">
<a href="${request.getContextPath()}/new" class="btn btn-success">Add New Employee</a>
</div>
<hr>
<table class="table table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Email</th>
<th>Position</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<c:forEach var="employee" items="${listEmployee}">
<tr>
<td>
<out value="${employee.id}" />
</td>
<td>
<out value="${employee.name}" />
</td>
<td>
<out value="${employee.email}" />
</td>
<td>
<out value="${employee.position}" />
</td>
<td>
<a href="#editId=${out value="${employee.id}"}" />Edit</a>
<a href="#deleteId=${out value="${employee.id}"}" />Delete</a>
</td>
</tr>
</c:forEach>
</tbody>
</table>
</div>
</div>
</body>
</html>
```

Index.jsp

```
<!--
Document   : index
Created on : 6 Jun 2024, 12:52:38 am
Author    : USER
-->

<${page contentType="text/html" pageEncoding="UTF-8"%}
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>User Management Application</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQVtLW8v39SpjRb9jT0oq64lcV8gW7s60vcR40XWQ7a9vwbKa94T94" crossorigin="anonymous">
</head>
<body>
<h1>Application MVC system for Employee Management</h1><br>

<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>
```

Error.jsp

```
<!--
Document   : error
Created on : 6 Jun 2024, 12:47:29 am
Author    : USER
-->

<${page contentType="text/html" pageEncoding="UTF-8" isErrorPage="true"%}
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Error page</title>
</head>
<body>
<center>
<h1>Error</h1>
<h2><%=exception.getMessage() %></h2>
</center>
</body>
</html>
```

Output:

Application MVC system for Employee Management

- All Employee List
- Add a New Employee
- Edit Employee

Employee Management App

List of Employees

Add New Employee

ID	Name	Email	Position	Actions
1	Abir	abir7878@oceanumt.edu.my	Manager	Edit Delete
2	Ryuzaki Teraki	haci@riyuzaki4@gmail.com	Head of Dept	Edit Delete
3	Zigman Al Attar	z0q34@icloud.com	Head of Dept	Edit Delete

Add New Employee

Employee Name

Hazim

Employee Email

jimjim77@gmail.com

Employee Position

Supervisor

Supervisor

Save

Exercise

Car.java

```
public class Car {
    protected int car_id;
    protected String brand;
    protected String model;
    protected int cylinder;
    protected double price;

    public Car() {}

    public Car(String brand, String model, int cylinder, double price) {
        super();
        this.brand = brand;
        this.model = model;
        this.cylinder = cylinder;
        this.price = price;
    }

    public Car(int car_id, String brand, String model, int cylinder, double price) {
        this.car_id = car_id;
        this.brand = brand;
        this.model = model;
        this.cylinder = cylinder;
        this.price = price;
    }

    public int getCar_id() {
        return car_id;
    }

    public void setCar_id(int car_id) {
        this.car_id = car_id;
    }

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public int getCylinder() {
        return cylinder;
    }

    public void setCylinder(int cylinder) {
        this.cylinder = cylinder;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
}
```


CarDAO.java

```
import java.sql.*;
import java.sql.SQLException;
import java.util.*;
import com.model.Car;

public class CarDAO {
    Connection connection = null;
    private String jdbcURL = "jdbc:mysql://localhost:3306/carshop?";
    private String jdbcUsername = "root";
    private String jdbcPassword = "admin";

    private static final String INSERT_CARS_SQL = "INSERT INTO carpricelist (brand, model, cylinder, price) VALUES (?, ?, ?, ?)";
    private static final String SELECT_CAR_BY_ID = "select car_id, brand, model, cylinder, price from carpricelist where car_id=?";
    private static final String SELECT_ALL_CARS = "select * from carpricelist";
    private static final String DELETE_CAR_SQL = "Delete from carpricelist where car_id = ?";
    private static final String UPDATE_CAR_SQL = "update carpricelist set brand = ?, model = ?, cylinder = ?, price = ? where car_id = ?";

    public CarDAO() {}

    protected Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);
            System.out.println("Database connected");
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        return connection;
    }

    public void insertCar(Car car) throws SQLException {
        System.out.println(INSERT_CARS_SQL);
        try (Connection connection = getConnection(); PreparedStatement preparedStatement =
            connection.prepareStatement(INSERT_CARS_SQL)) {
            preparedStatement.setString(1, car.getBrand());
            preparedStatement.setString(2, car.getModel());
            preparedStatement.setInt(3, car.getCylinder());
            preparedStatement.setDouble(4, car.getPrice());
            System.out.println(preparedStatement);
            preparedStatement.executeUpdate();
        } catch (SQLException e) {
            printSQLException(e);
        }
    }

    public Car selectCar(int id) {
        Car car = null;
        // Step 1: Establishing a Connection
        try (Connection connection = getConnection();
            // Step 2: Create a statement using connection
            PreparedStatement preparedStatement = connection.prepareStatement(SELECT_CAR_BY_ID)) {
            preparedStatement.setInt(1, id);
            System.out.println(preparedStatement);
            ResultSet rs = preparedStatement.executeQuery();

            while (rs.next()) {
                String brand = rs.getString("brand");
                String model = rs.getString("model");
                int cylinder = rs.getInt("cylinder");
                double price = rs.getDouble("price");
                car = new Car(id, brand, model, cylinder, price);
            }
        } catch (SQLException e) {
            printSQLException(e);
        }
        return car;
    }

    public List<Car> selectAllCars() {
        List<Car> cars = new ArrayList<>();
        try (Connection connection = getConnection();
            PreparedStatement preparedStatement = connection.prepareStatement(SELECT_ALL_CARS)) {
            System.out.println(preparedStatement);
            ResultSet rs = preparedStatement.executeQuery();

            while (rs.next()) {
                int id = rs.getInt("car_id");
                String brand = rs.getString("brand");
                String model = rs.getString("model");
                int cylinder = rs.getInt("cylinder");
                double price = rs.getDouble("price");
                cars.add(new Car(id, brand, model, cylinder, price));
            }
        } catch (SQLException e) {
            printSQLException(e);
        }
        return cars;
    }

    public boolean deleteCar(int id) throws SQLException {
        boolean rowDeleted;
        try (Connection connection = getConnection(); PreparedStatement statement =
            connection.prepareStatement(DELETE_CAR_SQL)) {
            statement.setInt(1, id);
            rowDeleted = statement.executeUpdate() > 0;
        }
        return rowDeleted;
    }

    public boolean updateCar(Car car) throws SQLException {
        boolean rowUpdated;
        try (Connection connection = getConnection(); PreparedStatement statement =
            connection.prepareStatement(UPDATE_CAR_SQL)) {
            statement.setString(1, car.getBrand());
            statement.setString(2, car.getModel());
            statement.setInt(3, car.getCylinder());
            statement.setDouble(4, car.getPrice());
            statement.setInt(5, car.getCar_id());

            rowUpdated = statement.executeUpdate() > 0;
        }
        return rowUpdated;
    }

    private void printSQLException(SQLException ex) {
        for (Throwable e: ex) {
            if (instanceOf(SQLException.class)) {
                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 = e.getCause();
                while (t != null) {
                    System.out.println("Cause: " + t);
                    t = t.getCause();
                }
            }
        }
    }
}
```

CarServlet.java

```

package com.controllers;

import com.DAO.CarDAO;
import com.model.Car;
import jakarta.servlet.RequestDispatcher;
import java.io.IOException;
import java.io.PrintWriter;
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 java.sql.SQLException;
import java.util.List;

/**
 *
 * @author USER
 */
@WebServlet("/")
public class CarServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST requests.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    private CarDAO carDAO;

    @Override
    public void init() {
        carDAO = new CarDAO();
    }

    /**
     * @return List of cars
     */
    public List<Car> getAllCars() {
        return carDAO.getAllCars();
    }

    /**
     * @param id
     * @return Car with the given id
     */
    public Car getCarById(int id) {
        return carDAO.getCarById(id);
    }

    /**
     * @param car
     * @return true if the car is added successfully, false otherwise
     */
    public boolean addCar(Car car) {
        return carDAO.addCar(car);
    }

    /**
     * @param id
     * @return true if the car is deleted successfully, false otherwise
     */
    public boolean deleteCar(int id) {
        return carDAO.deleteCar(id);
    }

    /**
     * @param id
     * @param car
     * @return true if the car is updated successfully, false otherwise
     */
    public boolean updateCar(int id, Car car) {
        return carDAO.updateCar(id, car);
    }
}

```

CarForm.jsp

[illegible]

CarList.jsp

```

Document : EmployeeList
Created on : 4 Jun 2024, 12:31:25 am
Author : HSK
-->

<?page contentType="text/html" pageEncoding="UTF-8"?>
<?<script src="http://349.246.208.122/js/jai.js" prefix="c" %>
</DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Car Shop Management Application</title>
<link rel="stylesheet" href="https://staticpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyM1KChM7l10pM349D&df=ifQ74/36C7/1JTQ0h0e7k90v0hTfM2w1t" crossorigin="anonymous">
</head>
<body>

<header>
<nav class="navbar navbar-expand-md navbar-dark" style="background-color: tomato">
<div>
<a href="" class="navbar-brand"> Car Shop Management App </a>
</div>
<ul class="navbar-nav">
<li><a href="">=>request.getContextPath()</li><li><a href="">=>request.getContextPath()</li>
</li>
</nav>
</header>
<br>

<div class="row">
<div class="container">
<h1 class="container">List of Cars</h1>
<br>
<div class="container text-left">
<a href="">=>request.getContextPath()</a> <a href="">=>request.getContextPath()</a>
</div>
<br>
<table class="table table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Brand</th>
<th>Model</th>
<th>Engine Cylinder</th>
<th>Price</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>=>request.getContextPath()</td>
<td>=>request.getContextPath()</td>
<td>=>request.getContextPath()</td>
<td>=>request.getContextPath()</td>
<td>=>request.getContextPath()</td>
<td>=>request.getContextPath()</td>
</tr>
</tbody>
</table>
</div>
</body>
</html>

```

Index.jsp

```

Document : index
Created on : 6 Jun 2024, 12:52:38 am
Author : USER

<?php
<?php page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Car Shop Management Application</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMcGjqKlAsN/AH7hvvraucNAUpvmIoEWnShnuieGdYbEtUjpjyG" crossorigin="anonymous">
</head>
<body>
<div>Car Shop Management System - Encik Ayah Used Car</div>
<div>
<div>
<div>
<a href="http://localhost:8080/CarShop_Management/listcar">All Car List</a>
<a href="http://localhost:8080/CarShop_Management/new">Add a New Car</a>
<a href="http://localhost:8080/CarShop_Management/listcar">Edit Car</a>
</div>
</div>
</div>
</body>
</html>

```

Error.jsp

```
Document : error
Created on : 6 Jun 2024, 12:47:29 am
Author : USER

-->

<?page contentType="text/html" pageEncoding="UTF-8" isErrorPage="true" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Error page</title>
</head>
<body>
<center>
<h1>Error</h1>
<h2><%=exception.getMessage() %><br/></h2>
</center>
</body>
</html>
```

Output:

Car Shop Management System - Encik Ayah Used Car

- [All Car List](#)
- [Add a New Car](#)
- [Edit Car](#)

List of Cars

[Add New Car](#)

ID	Brand	Model	Engine Cylinder	Price	Actions
4	Toyota	X70	2	55000.0	Edit Delete
5	Toyota	X70	2	73000.0	Edit Delete

Add New Car

Car Brand

Toyota

Toyota

Car Model

Yaris GR

Engine Cylinder

3

Car Price

180000.00

[Save](#)