

## UNIVERSITI MALAYSIA TERENGGANU

## **CSM3023 WEB BASED APPLICATION DEVELOPMENT**

# BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

## LAB 8

## **SEMESTER II 2023/2024**

# Prepared for:

SIR MOHD ARIZAL SHAMSIL BIN MAT RIFIN

Prepared by:

MUHAMMAD ZAHIER BIN RAZMI

(S67943)

# Task 1: Using JSP Page to Access a Simple MySQL Database

#### Code:

## Employee table

```
CREATE TABLE 'employee' (
  'id' int NOT NULL AUTO_INCREMENT,
  'name' varchar(60) DEFAULT NULL,
  'email' varchar(50) DEFAULT NULL,
  'position' varchar(15) DEFAULT NULL,
  PRIMARY KEY ('id')
)
```

## Employee.java

```
package em.model;
* @author Zahier
public class Employee {
   private int id;
   protected String email, name, position;
   public Employee() {
   public Employee(String name, String email, String position) {
       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 String getPosition() {
       return position;
```

```
public void setPosition(String position) {
    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;
}
```

# EmployeeDAO.java

```
package em.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 em.model.Employee;
 * @author Zahier
public class EmployeeDAO {
     Connection connection = null;
private String url = "jdbc:mysql://localhost:3306/lab8_task1";
private String username = "root";
     private String password = "admin";
    private static final String INSERT_EMPLOYEE = "INSERT INTO employee (name, email, position) VALUES (?, ?, ?)";
private static final String SELECT_ALL_EMPLOYEES = "SELECT * FROM employee";
private static final String SELECT_EMPLOYEE_BY_ID = "SELECT * FROM employee WHERE id = ?";
private static final String UPDATE_EMPLOYEE = "UPDATE employee SET name = ?, email = ?, position = ? WHERE id = ?";
private static final String DELETE_EMPLOYEE = "DELETE FROM employee WHERE id = ?";
     public EmployeeDAO() {
     public Connection getConnection() {
          Connection connection = null;
                    Class.forName("com.mysql.cj.jdbc.Driver");
                    connection = DriverManager.getConnection(url, username, password);
               catch (SQLException e) {
                   e.printStackTrace();
               catch (ClassNotFoundException e) {
                   e.printStackTrace();
          return connection;
     public void insertEmployee(Employee employee) {
                Connection connection = getConnection();
               PreparedStatement preparedStatement = connection.prepareStatement(INSERT_EMPLOYEE);
               preparedStatement.setString(1, employee.getName());
               preparedStatement.setString(2, employee.getEmail());
               preparedStatement.setString(3,
               employee.getPosition());
               preparedStatement.executeUpdate();
          catch (SQLException e) {
               e.printStackTrace();
```

```
public List<Employee> selectAllEmployees() {
    List<Employee> employees = new ArrayList<>();
    try {
        Connection connection = getConnection();
        PreparedStatement preparedStatement = connection.prepareStatement(SELECT_ALL_EMPLOYEES);
        ResultSet resultSet = preparedStatement.executeQuery();
        while ( resultSet.next() ) {
            Employee employee = new Employee();
            employee.setId(Integer.parseInt(resultSet.getString("id")));
             employee.setName(resultSet.getString("name"));
             employee.setEmail(resultSet.getString("email"));
             employee.setPosition(resultSet.getString("position"));
             employees.add(employee);
    catch (SQLException e) {
        e.printStackTrace();
return employees;
public Employee selectEmployeeByID(int id) {
    Employee employee = null;
        Connection connection = getConnection();
        PreparedStatement preparedStatement = connection.prepareStatement(SELECT_EMPLOYEE_BY_ID);
        preparedStatement.setInt(1, id);
        ResultSet resultSet = preparedStatement.executeQuery();
        while ( resultSet.next() ) {
            String name = resultSet.getString("name");
String email = resultSet.getString("email");
            String position =
            resultSet.getString("position");
            employee = new Employee(id, name, email, position);
    catch (SQLException e) {
        e.printStackTrace();
return employee;
public boolean updateEmployee (Employee employee) {
    boolean rowUpdated = false;
        Connection connection = getConnection();
        PreparedStatement preparedStatement = connection.prepareStatement(UPDATE_EMPLOYEE);
        preparedStatement.setString(1, employee.getName());
        preparedStatement.setString(2, employee.getEmail());
preparedStatement.setString(3, employee.getPosition());
        preparedStatement.setInt(4, employee.getId());
        rowUpdated = preparedStatement.executeUpdate() > 0;
    catch (SQLException e) {
       e.printStackTrace();
return rowUpdated;
public boolean deleteEmployee(int id) {
   boolean rowDeleted = false;
```

```
Connection connection = getConnection();
        PreparedStatement preparedStatement = connection.prepareStatement(DELETE_EMPLOYEE);
        preparedStatement.setInt(1, id);
         rowDeleted = preparedStatement.executeUpdate() > 0;
    catch (SQLException e) {
        e.printStackTrace();
return rowDeleted;
public 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

```
package em.controller;
import jakarta.servlet.RequestDispatcher;
import java.io.IOException;
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;
import em.dao.EmployeeDAO;
import em.model.Employee;
* @author Zahier
public class EmployeeServlet extends HttpServlet {
   private EmployeeDAO employeeDAO;
   public void init() {
   employeeDAO = new EmployeeDAO();
   protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
       String action = request.getServletPath();
        try {
           switch (action) {
                case "/new":
                   showNewForm(request, response);
```

```
case "/insert":
                  insertEmployee(request, response);
              case "/delete":
                  deleteEmployee(request, response);
                  break;
              case "/edit":
                   showEditForm(request, response);
                  break:
              case "/update":
                  updateEmployee(request, response);
                  break;
              default:
                  listEmployee(request, response);
     catch (SQLException e) (
         throw new ServletException(e);
private void listEmployee(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
     List<Employee> employees = employeeDAO.selectAllEmployees();
     request.setAttribute("employees", employees);
RequestDispatcher requestDispatcher = request.getRequestDispatcher("employeeList.jsp");
     requestDispatcher.forward(request, response);
private void showNewForm(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException (
    RequestDispatcher requestDispatcher = request.getRequestDispatcher("employeeForm.jsp");
    requestDispatcher.forward(request, response);
private void showEditForm(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
     int id = Integer.parseInt(request.getParameter("id"));
    Employee employee = employeeDAO.selectEmployeeByID(id);
    RequestDispatcher requestDispatcher = request.getRequestDispatcher("employeeForm.jsp");
request.setAttribute("employee", employee);
    requestDispatcher.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 employee = new Employee(name, email, position);
    employeeDAO.insertEmployee(employee);
response.sendRedirect("list");
private void updateEmployee(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
    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
int id = Integer.parseInt(request.getParameter("id"));
employeeDAO.deleteEmployee(id);
@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 "Short description"
```

## Index.jsp

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

</pre>
```

## employeeList.jsp

</html>

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 <html>
                     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
                     <title>Employee Management Application</title>
                     </
                             background-color: tomato;
                  }
           </style>
           </head>
                                <nav class="navbar navbar-expand-md navbar-dark">
                                         <a class="navbar-brand">Employee Management App</a>
                                         </div>

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

                                         </header>
                               <div class="row">
                               <div class="container">
<h3 class="text-center">List of Employees</h3>
                               <div class="container text-left">
    <a href="%% request.getContextPath() %>/new" class="btn btn-success">Add New Employee</a>
                               </div>
                               ID
                                                             Name
                                                              E-Mail
                                                              Position
                                                             Actions
                                         </thead>
                                                    <c:forEach var="employee" items="${employees}">
                                                            ctd>c:out value="$(employee.id)" />

<c:out value="$(employee.name)" />

<c:out value="$(employee.email)" />

<c:out value="$(employee.position)" />

'\[
\text{A href="edit?id=<c:out value='$(employee.id)' />">\[
\text{i class="fa-solid fa-pen-tolsquare"></i>
\text{i>}\/a>\forall fabsp;\forall fabsp
                                                              </c:forEach>
                                         </tbody
                               </div>
         </body>
```

## employeeForm.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 <html>
          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
                  citile/Employee Management Application//title/Employee Management Application
//title/Employee Management Application

//title/Employee Management Application

//title/Employee Management Application

//title/Employee Management Application

//title/Employee Management Application

//title/Employee Management Application

//title/Employee Mana
           <style>
                   nav (
          </head>
          <body>
                            <nav class="navbar navbar-expand-md navbar-dark">

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

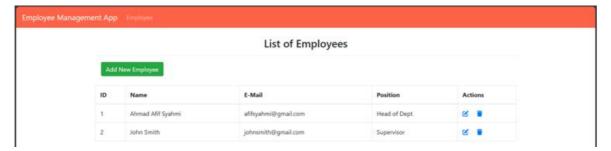
                           </1175
                   </header>
                   <br>
                           <div class="container col-md-5">
<div class="card">
<div class="card-body">
                                    </c:if>
                                     <c:if test="${employee == null}">
                                              <form action="insert" method="post">
                                     </c:if>
                                             <c:if test="${employee != null}">Edit Employee</c:if>
                                             <c:if test="${employee == null}">Add New Employee</c:if>
                                    <c:if test="${employee != null}">
                                                                           "hidden" name="id" value="<c:out value='$(employee.id)' />"></c:if>
                                             <input type
                                                     <label for="name">Name</label>
<input type="text" name="name"</pre>
                                                                                                                      me" id="name" class="form-control" value="<c:out value='${employee.name}' />" required>
                                             </fieldset>
                                             <fieldset class="form-group">
     <label for="email">E-Mail</label>
                                                      <input type="text" name="email" id="email" class="form-control" value="<g:out value="${employee.email}' />">
                                             <fieldset class="form-group">
    <label for="position">Position</label>
                                                      <input type="text" class="form-control" value="<c:out value='${employee.position}' />" readonly>
<input list="positionList" name="position" id="position" class="form-control">
                                                      <datalist id="positionList">
<option value="Manager"></option>
                                                              <coption value="Head of Dept"></option>
<option value="Supervisor"></option>
<option value="Director"></option>
                                                      </datalist>
                                     <button type="submit" class="btn btnDsuccess">Save</button>
                           </form>
                  </div>
                  </div>
                  </div>
          </body>
</html>
```

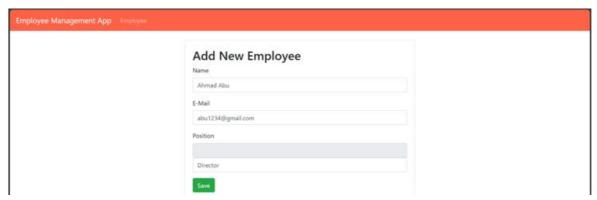
### Error.jsp

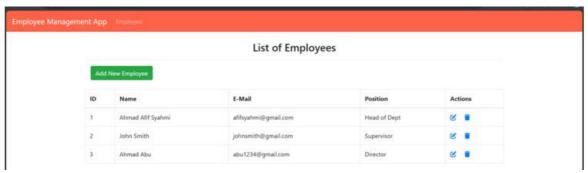
# Output:

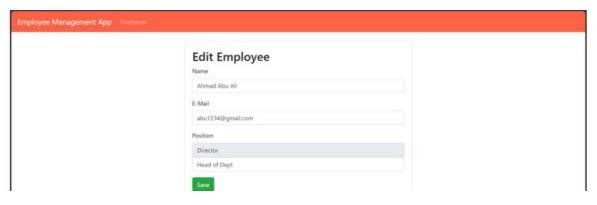
# Application MVC System for Employee Management

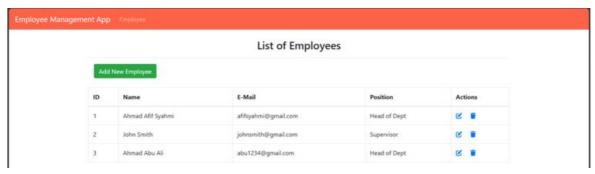
- All Employees List
   Add A New Employee
   Edit Employee

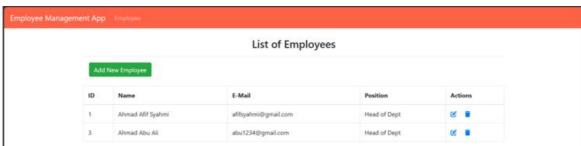












# **Car Shop Application**

## Code:

#### Car table

```
CREATE TABLE 'car' (
'id' int NOT NULL AUTO_INCREMENT,
'brand' varchar(15) DEFAULT NULL,
'model' varchar(30) DEFAULT NULL,
'cylinder' int DEFAULT NULL,
'price' double DEFAULT NULL,
PRIMARY KEY ('id')
)
```

# Car.java

```
package com.cs.model;

/**

* * @author Zahier

*/
public class Car {

    private int id, cylinder;
    private String brand, model;
    private double price;

    public Car() {

    }

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

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

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

    public int getId() {
        return id;
    }
}
```

```
public void setId(int id) {
    this.id = id;
}

public int getCylinder() {
    return cylinder;
}

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

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 double getPrice() {
    return price;
}

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

## CarDAO.java

```
package com.cs.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.cs.model.Car;
* @author Zahier
public class CarDAO {
    Connection connection = null;
private String url = "jdbc:mysql://localhost:3306/CarShopApplication";
    private String username = "root";
    private String password = "admin";
    private static final String INSERT_CAR = "INSERT INTO car (brand, model, cylinder, price) VALUES (2, 2, 2, 2)";
    private static final String SELECT ALL CARS = "SELECT * FROM car";
    private static final String SELECT CAR BY ID = "SELECT * FROM car WHERE id = ?";

private static final String UPDATE CAR = "UPDATE car SET brand = ?, model = ?, cylinder = ?, price = ? WHERE id = ?";
    private static final String DELETE_CAR = "DELETE FROM car WHERE id = ?";
    public Connection getConnection() {
        Connection connection = null;
            try {
                Class.forName("com.mysgl.cj.jdbc.Driver");
                connection = DriverManager.getConnection(url, username, password);
            catch (SQLException e) {
                e.printStackTrace();
            catch (ClassNotFoundException e) {
               e.printStackTrace();
       return connection;
   public void insertCar(Car car) {
           Connection connection = getConnection();
            PreparedStatement preparedStatement = connection.prepareStatement(INSERT CAR);
           preparedStatement.setString(1, car.getBrand());
            preparedStatement.setString(2, car.getModel());
            preparedStatement.setInt(3, car.getCylinder());
           preparedStatement.setDouble(4, car.getPrice());
           preparedStatement.executeUpdate();
       catch (SQLException e) {
           e.printStackTrace();
   public List<Car> selectAllCars() {
       List<Car> cars = new ArrayList<>();
           try {
                Connection connection = getConnection();
                PreparedStatement preparedStatement = connection.prepareStatement(SELECT ALL CARS);
                ResultSet resultSet = preparedStatement.executeQuery();
            while ( resultSet.next() ) {
                Car car = new Car();
                car.setId(Integer.parseInt(resultSet.getString("id")));
```

```
car.setBrand(resultSet.getString("brand"));
            car.setModel(resultSet.getString("model"));
            car.setCylinder(resultSet.getInt("cylinder"));
            car.setPrice(resultSet.getDouble("price"));
            cars.add(car);
    catch (SQLException e) {
       e.printStackTrace();
    return cars;
public Car selectCarByID(int id) {
   Car car = null;
       try {
            Connection connection = getConnection();
            PreparedStatement preparedStatement = connection.prepareStatement(SELECT CAR BY ID);
            preparedStatement.setInt(1, id);
            ResultSet resultSet = preparedStatement.executeQuery();
        while ( resultSet.next() ) {
           String brand = resultSet.getString("brand");
            String model = resultSet.getString("model");
            int cylinder = resultSet.getInt("cylinder");
double price = resultSet.getDouble("price");
            car = new Car(id, brand, model, cylinder, price);
        catch (SQLException e) {
           e.printStackTrace();
    return car;
public boolean updateCar(Car car) {
    boolean rowUpdated = false;
        Connection connection = getConnection();
        PreparedStatement preparedStatement = connection.prepareStatement(UPDATE CAR);
       preparedStatement.setString(1, car.getBrand());
        preparedStatement.setString(2, car.getModel());
       preparedStatement.setInt(3, car.getCylinder());
        preparedStatement.setDouble(4, car.getPrice());
       preparedStatement.setInt(5, car.getId());
       rowUpdated = preparedStatement.executeUpdate() > 0;
    catch (SQLException e) {
       e.printStackTrace();
return rowUpdated;
public boolean deleteCar(int id) {
   boolean rowDeleted = false;
        Connection connection = getConnection();
        PreparedStatement preparedStatement = connection.prepareStatement(DELETE_CAR);
       preparedStatement.setInt(1, id);
       rowDeleted = preparedStatement.executeUpdate() > 0;
    catch (SQLException e) {
       e.printStackTrace();
return rowDeleted;
```

# CarServlet.java

```
package com.cs.controller;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.SQLException;
import java.util.List;
import com.cs.model.Car;
import com.cs.dao.CarDAO;
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.annotation.WebServlet;
* @author Zahier
*/
@WebServlet("/")
public class CarServlet extends HttpServlet {
    private CarDAO carDAO;
    public void init() (
        carDAO = new CarDAO();
     protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
          String action = request.getServletPath();
              switch (action) {
  case "/new":
     showNewForm(request, response);
                   case "/insert":
                      insertCar(request, response);
break;
                   case "/delete":
                       deleteCar(request, response);
                   case "/edit":
                      showEditForm(request, response);
break;
                   case "/update":
                       updateCar(request, response);
                       break;
                   default:
                       listCar(request, response);
          catch (SQLException e) {
              throw new ServletException(e);
    private void showNewForm(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
    RequestDispatcher requestDispatcher = request.getRequestDispatcher("carForm.jsp");
         requestDispatcher.forward(request, response);
    private void showEditForm(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
   int id = Integer.parseInt(request.getParameter("id"));
         Car car = carDAO.selectCarBvID(id);
```

```
RequestDispatcher requestDispatcher = request.getRequestDispatcher("carForm.jsp");
      request.setAttribute("car", car);
      requestDispatcher.forward(request, response);
private void insertCar(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
      String brand = request.getParameter("brand");
String model = request.getParameter("model");
      int cylinder = Integer.parseInt(request.getParameter("cylinder"));
double price = Double.parseDouble(request.getParameter("price"));
      Car car = new Car(brand, model, cylinder, price);
     carDAO.insertCar(car);
response.sendRedirect("list");
private void listCar(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
   List<Car> cars = carDAO.selectAllCars();
   request.setAttribute("cars", cars);
   RequestDispatcher requestDispatcher = request.getRequestDispatcher("carList.jsp");
   requestDispatcher.forward(request, response);
private void updateCar(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
     vate void updateGar(HttpServletRequest request, HttpServletResponse
int id = Integer.pareeInt(request.getParameter("id"));
String brand = request.getParameter("brand");
String model = request.getParameter("model");
int cylinder = Integer.parseInt(request.getParameter("cylinder"));
double price = Double.parseDouble(request.getParameter("price"));
      Car car = new Car(id, brand, model, cylinder, price);
      carDAO.updateCar(car);
      response.sendRedirect("list");
private void deleteCar(HttpServletRequest request, HttpServletResponse response) throws SQLException, IOException, ServletException {
     int id = Integer.parseInt(request.getParameter("id"));
      carDAO.deleteCar(id);
      response.sendRedirect("list");
@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);
Moverride
public String getServletInfo() {
     return "Short description";
```

#### Index.jsp

## CarList.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<*dtaglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" \
<%dtaglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" \
</pre>
  <head>
     </head>
  <body>
     </button>
     <div class="collapse navbar-collapse" id="

          <a class="nav-link" href="list">Cars</a>
          </div>
     </nav>
     <a href="new"><button class="btn btn-primary">Add new car</button></a>

     <thead class="table-dark text-center">
                  ID
                  Brand
                  Model
                 Sylinders/th
                  Action
               </thead>
               <c:forEach var="car" items="${cars}">
                    <c:out value="${car.id}" />
                    </c:forEach>

<
</html>
```

## CarForm.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<*detaglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" $>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" $>
  <head>
     <body>
     </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          </div>
     </nav>
     <div class="card-body">
  <c:if test="$(car != null)">
             <form action="update" method="post">
<h2>Edit Car</h2>
                <input type="hidden" name="id" value="<c:out value='${car.id}' />">
           <c:if test="${car == null}">
            <form action="insert" method="post">
<h2>Add New Car</h2>
```

```
</div>
                                           </div>
                                          <div class="form-group">
     <label for="cylinder">Cylinder</label>
                                                     <input type="number" class="form-control" name="cylinder" id="cylinder" min="0" value="<g:out value='$(car.cylinder)' />" requi
                                           <div class="form-group"
                                           \text{\text{Cinput type="number" class="form-control" name="price" id="price" min="0" step="0.01" value="\text{\text{c:out value='\text{car.price\}' /\rightarrow\" re
\text{\text{\text{div}}}</ti>
                                           <div class="d-flex justify-conte
<c:if test="${car != null}">
                                                                chutton type="reset" class="btn">Reset</button>
<br/>

                                                     </form>
                                </div
                      </div>
         </div>
  </html>
```

# Error.jsp

## Output:

# Application MVC System for Car Shop All car list Add a new car



