Universiti Malaysia Terengganu

Faculty of Computer Sciences and Mathematics

Web Based Application Development

CSM3023

Lab Report 8

Prepared for:

Sir. Mohd Arizal Shamsil bin Mat Rifin

Prepared by:

Ahmad Afif Syahmi bin Ahmad Rozali (S65526)

5th June 2024

Bachelor of Computer Science (Mobile Computing) with Honors

Semester II 2023/2024

Employee Management Application

- Codes:
  - 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 Ahmad Afif Syahmi bin Ahmad Rozali
 */
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 Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class EmployeeDAO {

    Connection connection = null;
    private String url =
"jdbc:mysql://localhost:3306/csm3023lab8";
    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;

        try {
            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) {

        try {
            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;

        try {
            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;
        try {
            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;

        try {
            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 Ahmad Afif Syahmi bin Ahmad Rozali
 */

@WebServlet("/")

public class EmployeeServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
     * 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;

    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);
                    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);
            }
        }
        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 response)
            throws SQLException, IOException, ServletException {

        int id = Integer.parseInt(request.getParameter("id"));
        employeeDAO.deleteEmployee(id);
        response.sendRedirect("list");

    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</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 doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * 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 {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

- index.jsp

```
<%--
    Document   : index
    Created on : 5 Jun 2024, 3:44:24 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@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://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/boot
strap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N
" crossorigin="anonymous">
    </head>
    <body>
```

```
        <h1>Application MVC System for Employee Management</h1>
        <br>
        <ul>
            <li><a href="list">All Employees List</a></li>
            <li><a href="new">Add A New Employee</a></li>
            <li><a href="list">Edit Employee</a></li>
        </ul>
    </body>
</html>
```

- employeeList.jsp

```
<%--
    Document   : employeeList
    Created on : 5 Jun 2024, 3:44:07 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="jakarta.tags.core" %>
<!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://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/boot
strap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N
" crossorigin="anonymous">
        <script src="https://kit.fontawesome.com/aa5de83a5b.js"
crossorigin="anonymous"></script>
        <style>
            nav {
                background-color: tomato;
            }
        </style>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-expand-md navbar-dark">
                <div>
                    <a class="navbar-brand">Employee Management
App</a>
                </div>
                <ul class="navbar-nav">
                    <li><a href="<%= request.getContextPath()
%>/list" class="nav-link">Employee</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>E-Mail</th>
                            <th>Position</th>
                            <th>Actions</th>
                        </tr>
                    </thead>
                    <tbody>
                        <c:forEach var="employee"
items="${employees}">
                            <tr>
                                <td><c:out
value="${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}' />"><i class="fa-solid fa-pen-to-
square"></i></a>&nbsp;&nbsp;&nbsp;&nbsp;
                                    <a href="delete?id=<c:out
value='${employee.id}' />"><i class="fa-solid fa-trash"></i></a>
                                </td>
                            </tr>
                        </c:forEach>
                    </tbody>
                </table>
            </div>
        </div>
    </body>
</html>
```

- employeeForm.jsp

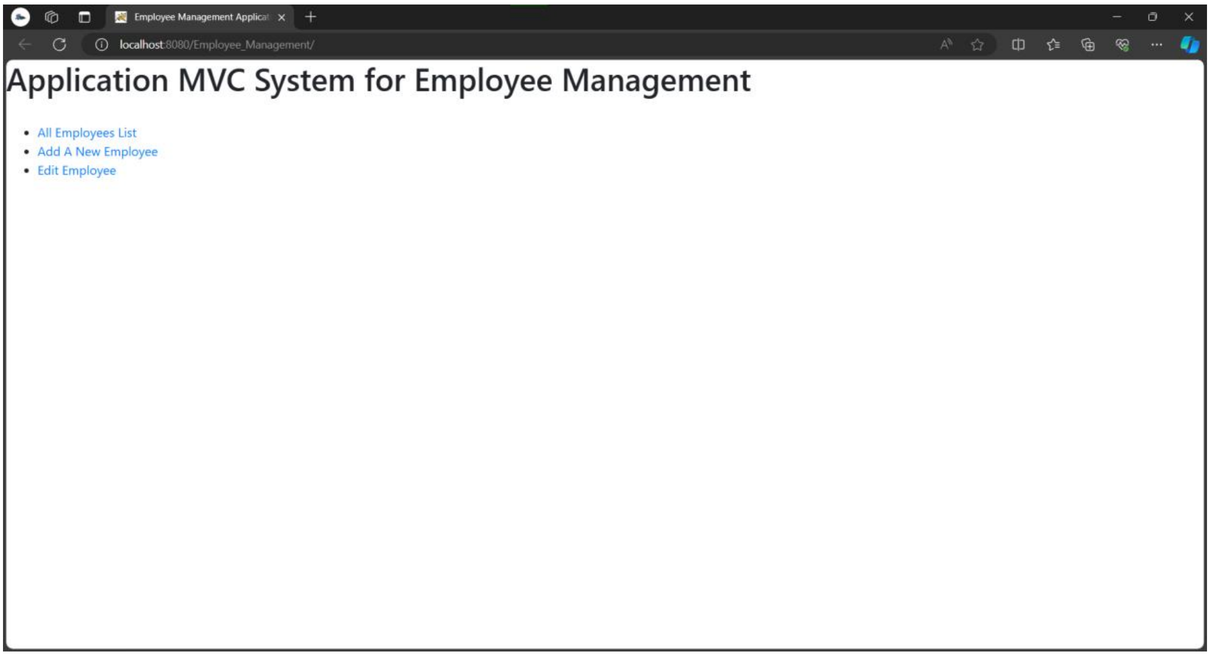
```
<%--
    Document   : employeeForm
    Created on : 5 Jun 2024, 3:43:53 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="jakarta.tags.core" %>
<!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://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/boot
strap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N
" crossorigin="anonymous">
        <style>
            nav {
                background-color: tomato;
            }
        </style>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-expand-md navbar-dark">
                <div>
                    <a class="navbar-brand">Employee Management
App</a>
                </div>
                <ul class="navbar-nav">
                    <li><a href="<%= request.getContextPath()
%>/list" class="nav-link">Employee</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 for="name">Name</label>
                        <input type="text" name="name" id="name"
class="form-control" value="<c:out value='${employee.name}' />"
required>
                    </fieldset>
                    <fieldset class="form-group">
                        <label for="email">E-Mail</label>
                        <input type="text" name="email"
id="email" class="form-control" value="<c:out
value='${employee.email}' />">
                    </fieldset>

                    <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>
                            <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>
```

- error.jsp

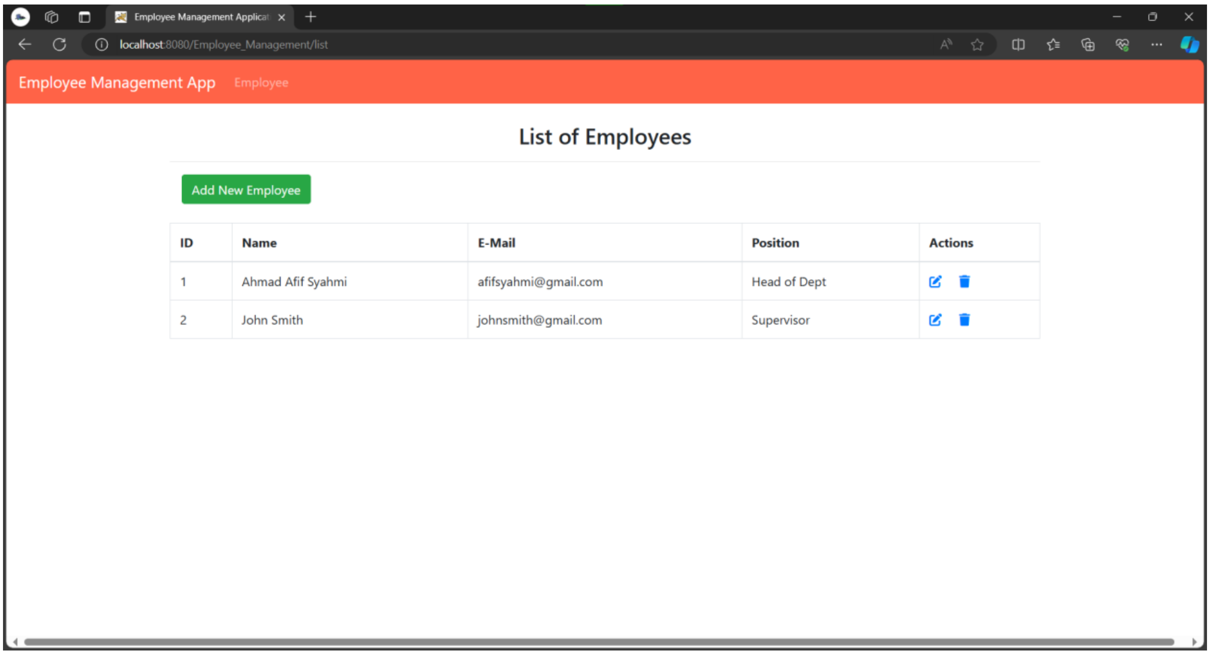
```
<%--
    Document   : error
    Created on : 5 Jun 2024, 3:44:15 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@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>
            <br>
        </center>
    </body>
</html>
```

- Outputs:


Employee Management Applicat
localhost:8080/Employee_Management/

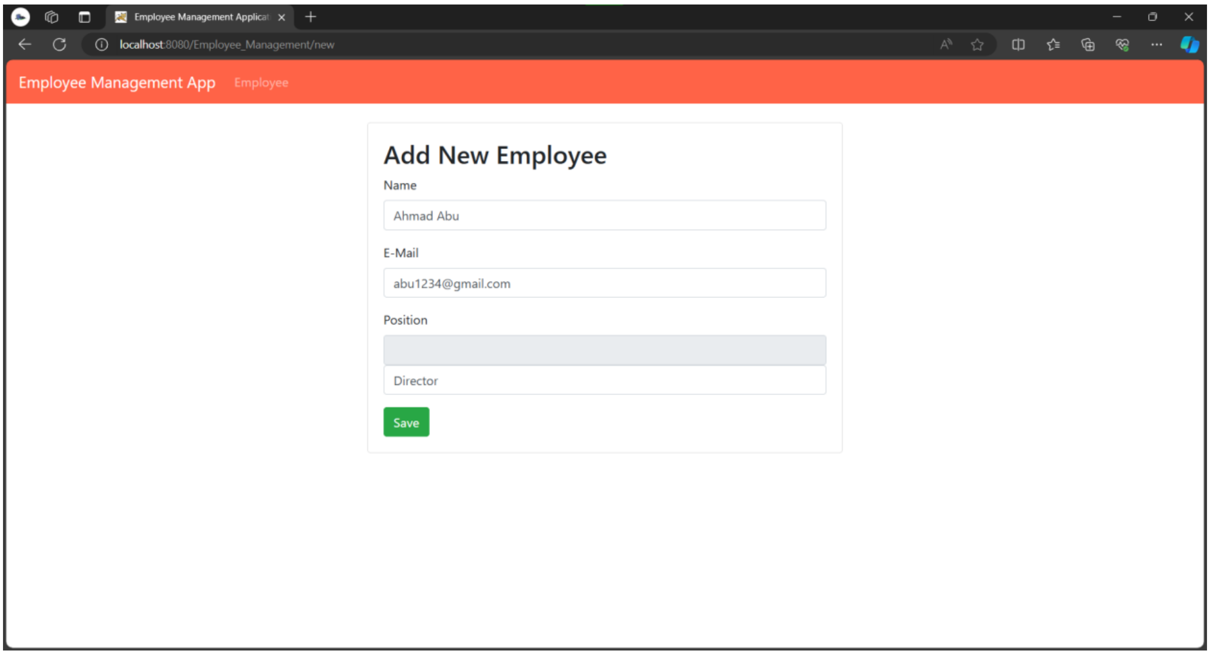
Application MVC System for Employee Management

- All Employees List
- Add A New Employee
- Edit Employee


Employee Management Applicat
localhost:8080/Employee_Management/list
Employee Management App
Employee
Add New Employee

List of Employees

ID	Name	E-Mail	Position	Actions
1	Ahmad Afif Syahmi	afifsyahmi@gmail.com	Head of Dept	✎ 🗑
2	John Smith	johnsmith@gmail.com	Supervisor	✎ 🗑



Employee Management Applicat
localhost:8080/Employee_Management/new
Employee Management App
Employee

Add New Employee

Name

Ahmad Abu

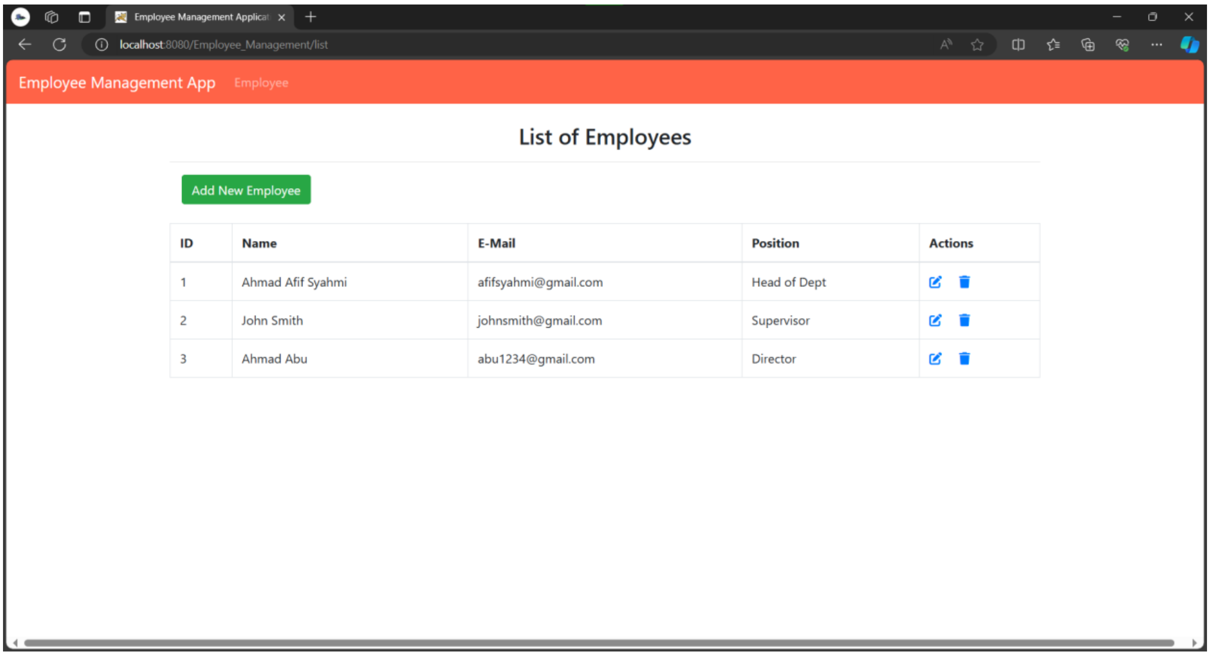
E-Mail

abu1234@gmail.com

Position

Director

Save

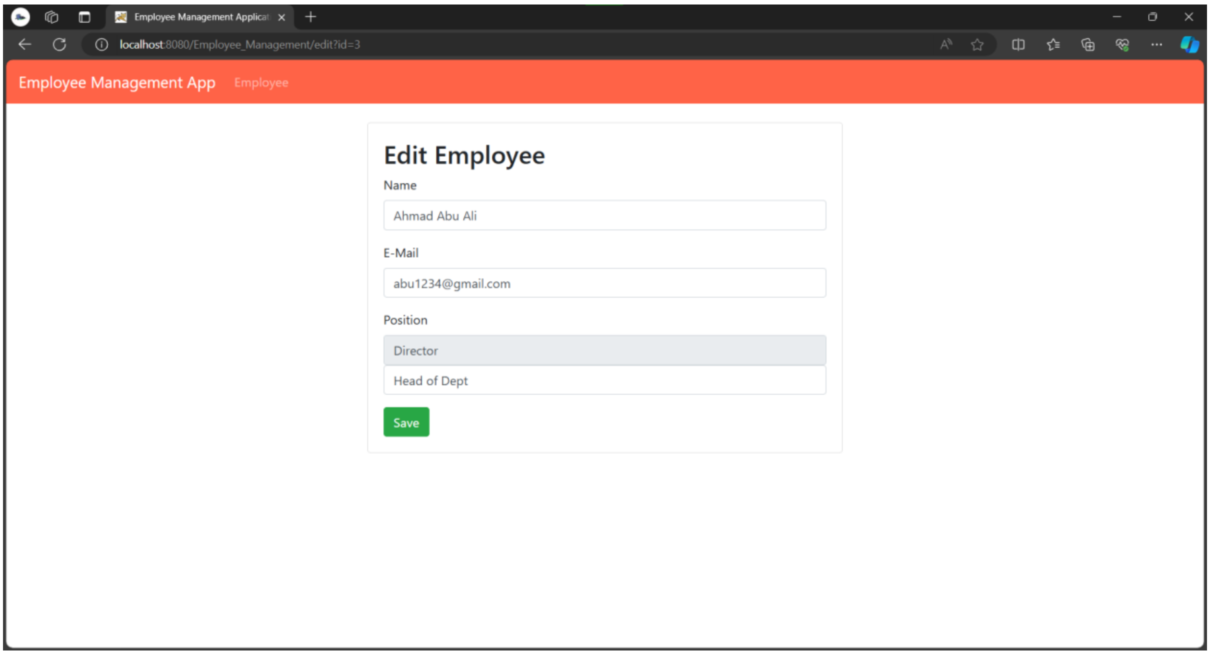


Employee Management Applicat
localhost:8080/Employee_Management/list
Employee Management App
Employee

List of Employees

Add New Employee

ID	Name	E-Mail	Position	Actions
1	Ahmad Afif Syahmi	afifsyahmi@gmail.com	Head of Dept	✎ 🗑
2	John Smith	johnsmith@gmail.com	Supervisor	✎ 🗑
3	Ahmad Abu	abu1234@gmail.com	Director	✎ 🗑


Employee Management Applicat
localhost:8080/Employee_Management/edit?id=3
Employee Management App
Employee

Edit Employee

Name

Ahmad Abu Ali

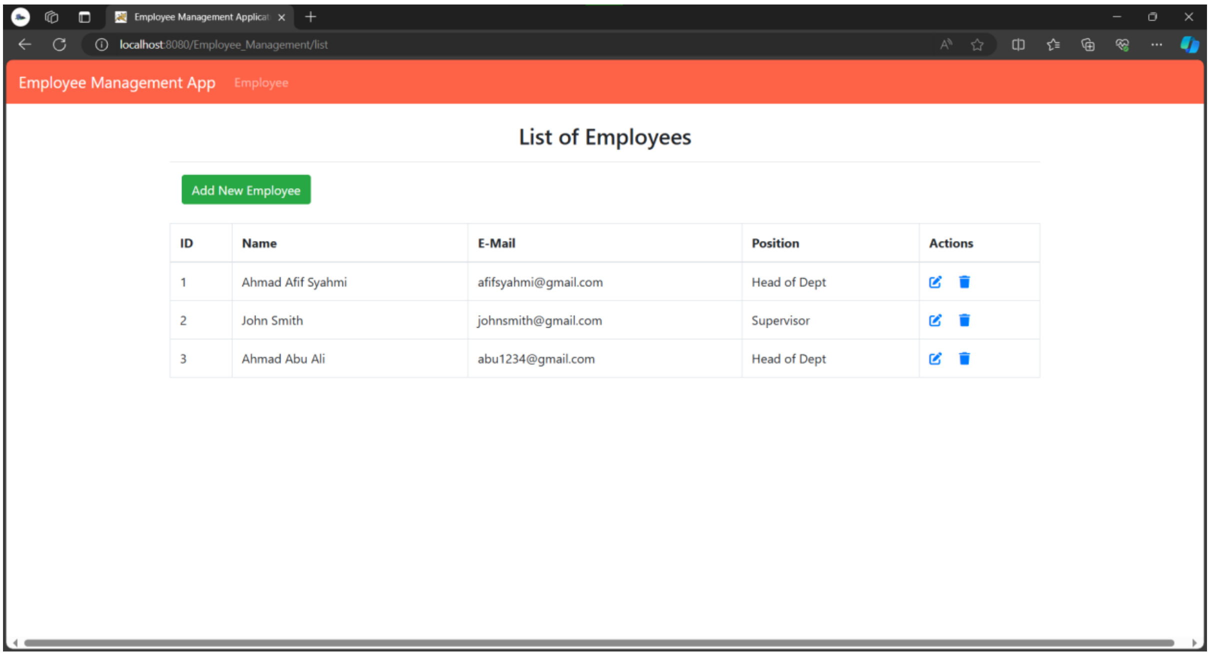
E-Mail

abu1234@gmail.com

Position

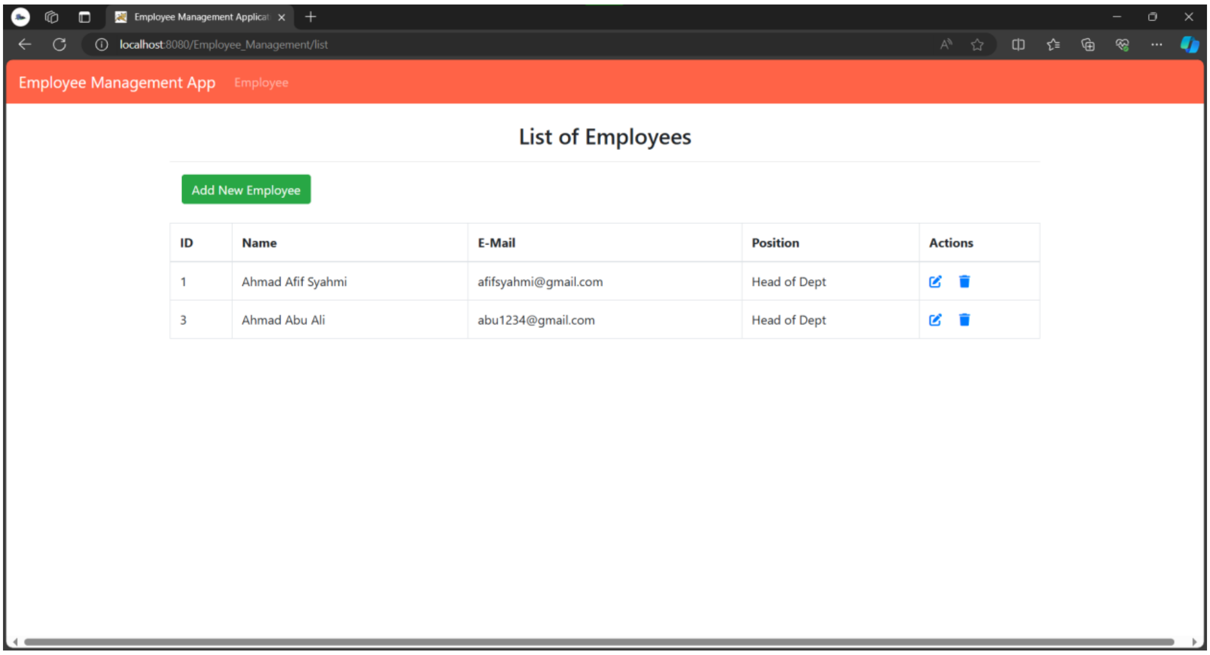
Director

Head of Dept


Employee Management Applicat
localhost:8080/Employee_Management/list
Employee Management App
Employee

List of Employees

ID	Name	E-Mail	Position	Actions
1	Ahmad Afif Syahmi	afifsyahmi@gmail.com	Head of Dept	✎ 🗑
2	John Smith	johnsmith@gmail.com	Supervisor	✎ 🗑
3	Ahmad Abu Ali	abu1234@gmail.com	Head of Dept	✎ 🗑



Employee Management Applicat
localhost:8080/Employee_Management/list
Employee Management App
Employee

List of Employees

ID	Name	E-Mail	Position	Actions
1	Ahmad Afif Syahmi	afifsyahmi@gmail.com	Head of Dept	✎ 🗑
3	Ahmad Abu Ali	abu1234@gmail.com	Head of Dept	✎ 🗑

Car Shop Application

- Codes:
    - userprofile 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 Ahmad Afif Syahmi bin Ahmad Rozali
 */
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.brand = 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 Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class CarDAO {
    Connection connection = null;
    private String url =
"jdbc:mysql://localhost:3306/csm3023lab8";
    private String username = "root";
    private String password = "admin";

    private static final String INSERT_CAR = "INSERT INTO car
(brand, model, cylinder, price) VALUES (?, ?, ?, ?)";
    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.mysql.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) {

        try {
            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;

        try {
            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;

        try {
            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 Ahmad Afif Syahmi bin Ahmad Rozali
 */

@WebServlet("/")

public class CarServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and 
<code>POST</code>
     * 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 CarDAO carDAO;

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

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

        String action  = request.getServletPath();

        try {
            switch (action) {
                case "/new":
                    showNewForm(request, response);
                    break;

                case "/insert":
                    insertCar(request, response);
                    break;

                case "/delete":
                    deleteCar(request, response);
                    break;

                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.selectCarByID(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 {

        int id = Integer.parseInt(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");

    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet 
methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</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 doGet(HttpServletRequest request, 
HttpServletResponse response)
```

```
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * 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 {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

- index.jsp

```
<%--
    Document   : index
    Created on : 14 Jun 2024, 8:38:05 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/boot
strap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N
" crossorigin="anonymous">
        <title>Car Shop | Home</title>
    </head>
    <body>

        <h1>Application MVC System for Car Shop</h1>
        <br>
        <ul>
            <li><a href="list">All car list</a></li>
            <li><a href="new">Add a new car</a></li>
        </ul>

        <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.
min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj
" crossorigin="anonymous"></script>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct
" crossorigin="anonymous"></script>

    </body>
</html>
```

- carList.jsp

```
<%--
    Document   : carList
    Created on : 14 Jun 2024, 8:44:37 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="jakarta.tags.core" %>
<%@taglib prefix="fmt" uri="jakarta.tags.fmt" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/boot
strap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N
" crossorigin="anonymous">
        <script src="https://kit.fontawesome.com/aa5de83a5b.js"
crossorigin="anonymous"></script>
        <title>Car Shop | List</title>
    </head>
    <body>
        <nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
            <a class="navbar-brand" href="index.jsp">Car
Shop</a>

            <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="list">Cars</a>
                    </li>
                </ul>
            </div>
        </nav>
        <main class="container mt-3">
            <h1 class="text-center">List of Cars</h1>
            <hr>
            <a href="new"><button class="btn btn-primary">Add
new car</button></a>

            <table class="table table-bordered mt-3">
                <thead class="table-dark text-center">
                    <tr>
                        <th>ID</th>
                        <th>Brand</th>
                        <th>Model</th>
                        <th>Cylinder</th>
                        <th>Price</th>
                        <th colspan="2">Action</th>
                    </tr>
                </thead>
                <tbody>
                    <c:forEach var="car" items="${cars}">
                        <tr>
                            <td><c:out value="${car.id}" /></td>
                            <td><c:out value="${car.brand}"
/></td>
                            <td><c:out value="${car.model}"
/></td>
                            <td><c:out value="${car.cylinder}"
/></td>
                            <td><fmt:formatNumber type="number"
maxFractionDigits="2" groupingUsed="false" value="${car.price}"
/></td>
                            <td class="text-center"><a
href="edit?id=<c:out value='${car.id}' />"><i class="fa-solid
fa-pen-to-square"></i></a></td>
                            <td class="text-center"><a
href="delete?id=<c:out value='${car.id}' />"><i class="fa-solid
fa-trash"></i></a></td>
                        </tr>
                    </c:forEach>
                </tbody>
            </table>
        </main>

        <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.
min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj
" crossorigin="anonymous"></script>
```

```
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct
" crossorigin="anonymous"></script>
    </body>
</html>
```

- carForm.jsp

```
<%--
    Document   : carForm
    Created on : 14 Jun 2024, 9:23:18 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="jakarta.tags.core" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/boot
strap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N
" crossorigin="anonymous">
        <title>Car Shop | Form</title>
    </head>
    <body>
        <nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
            <a class="navbar-brand" href="index.jsp">Car
Shop</a>

            <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="list">Cars</a>
                    </li>
                </ul>
            </div>
        </nav>
        <main class="container mt-3">
            <div class="row justify-content-center">
                <div class="card col-6">
                    <div class="card-body">
```

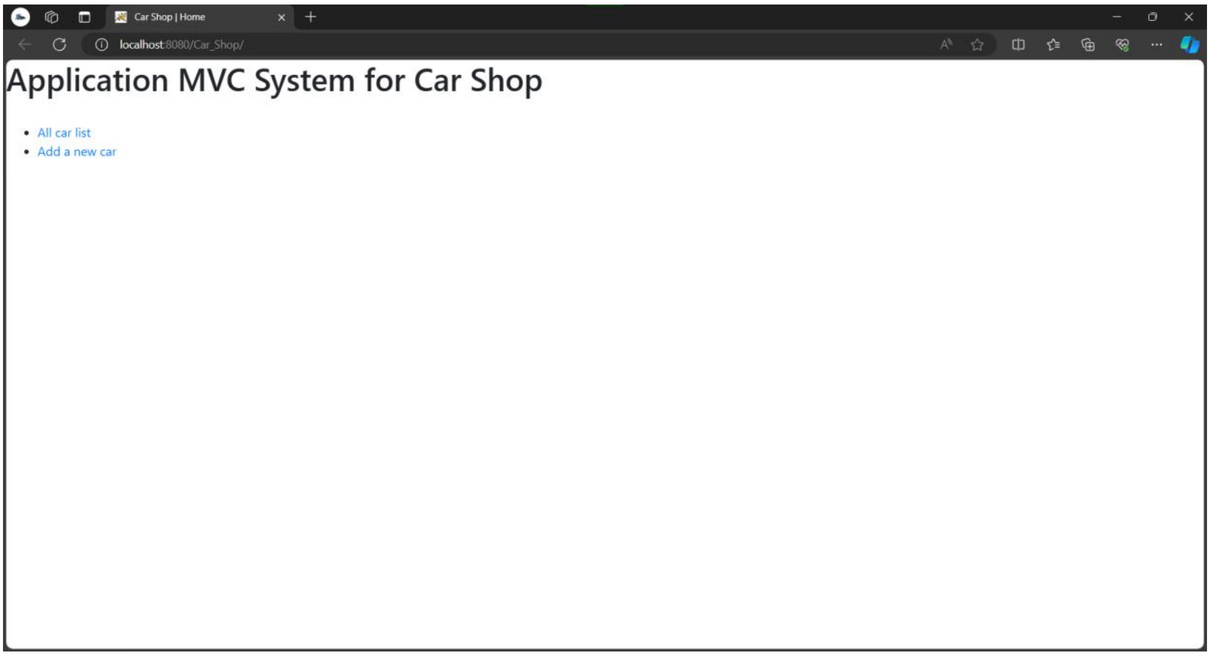
```
                        <c:if test="${car != null}">
                            <form action="update" method="post">
                                <h2>Edit Car</h2>
                                <input type="hidden" name="id"
value="<c:out value='${car.id}' />">
                        </c:if>
                        <c:if test="${car == null}">
                            <form action="insert" method="post">
                                <h2>Add New Car</h2>
                        </c:if>
                            <div class="form-group">
                                <label for="brand">Brand</label>
                                <input type="text" class="form-
control" name="brand" id="brand" maxlength="15" value="<c:out
value='${car.brand}' />" required>
                            </div>
                            <div class="form-group">
                                <label for="model">Model</label>
                                <input type="text" class="form-
control" name="model" id="model" maxlength="30" value="<c:out
value='${car.model}' />" required>
                            </div>
                            <div class="form-group">
                                <label
for="cylinder">Cylinder</label>
                                <input type="number"
class="form-control" name="cylinder" id="cylinder" min="0"
value="<c:out value='${car.cylinder}' />" required>
                            </div>
                            <div class="form-group">
                                <label for="price">Price</label>
                                <input type="number"
class="form-control" name="price" id="price" min="0" step="0.01"
value="<c:out value='${car.price}' />" required>
                            </div>
                            <div class="d-flex justify-content-
end">
                                <c:if test="${car != null}">
                                    <button type="reset"
class="btn">Reset</button>
                                    <button type="submit"
class="btn btn-primary">Save</button>
                                </c:if>
                                <c:if test="${car == null}">
                                    <button type="reset"
class="btn">Clear</button>
                                    <button type="submit"
class="btn btn-primary">Add</button>
                                </c:if>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </main>

        <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.
min.js" integrity="sha384-
```

```
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj
" crossorigin="anonymous"></script>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct
" crossorigin="anonymous"></script>
    </body>
</html>
```

- error.jsp

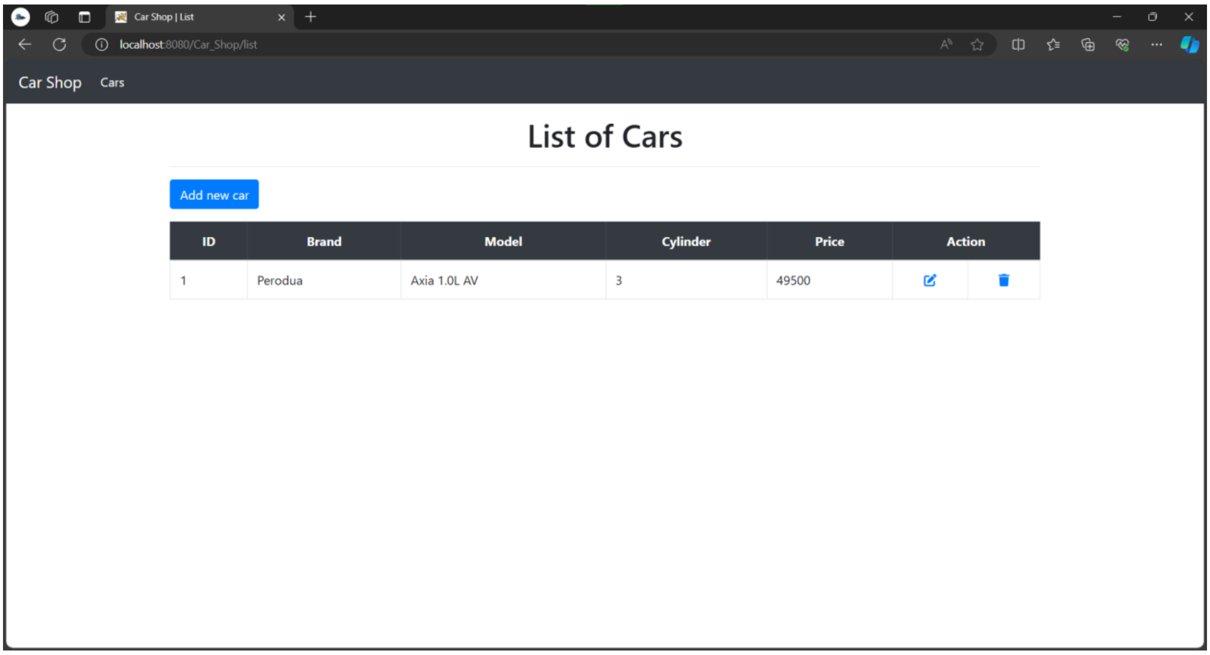
```
<%--
    Document   : error
    Created on : 14 Jun 2024, 10:02:58 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"
isErrorPage="true" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/boot
strap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N
" crossorigin="anonymous">
        <title>Error page</title>
    </head>
    <body>

        <div class="container text-center">
            <h1>Error</h1>
            <h2><%= exception.getMessage() %></h2>
        </div>

        <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.
min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj
" crossorigin="anonymous"></script>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct
" crossorigin="anonymous"></script>

    </body>
</html>
```

- Output:



Application MVC System for Car Shop

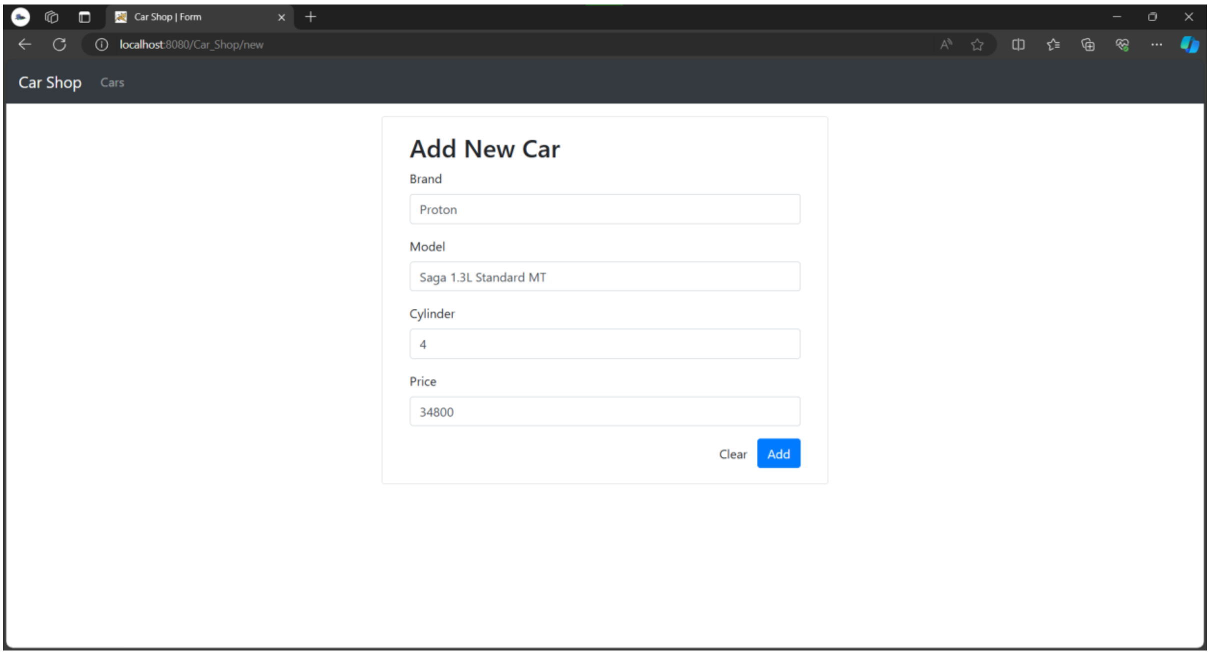
- All car list
- Add a new car



List of Cars

Add new car

ID	Brand	Model	Cylinder	Price	Action	
1	Perodua	Axia 1.0L AV	3	49500	✎	🗑



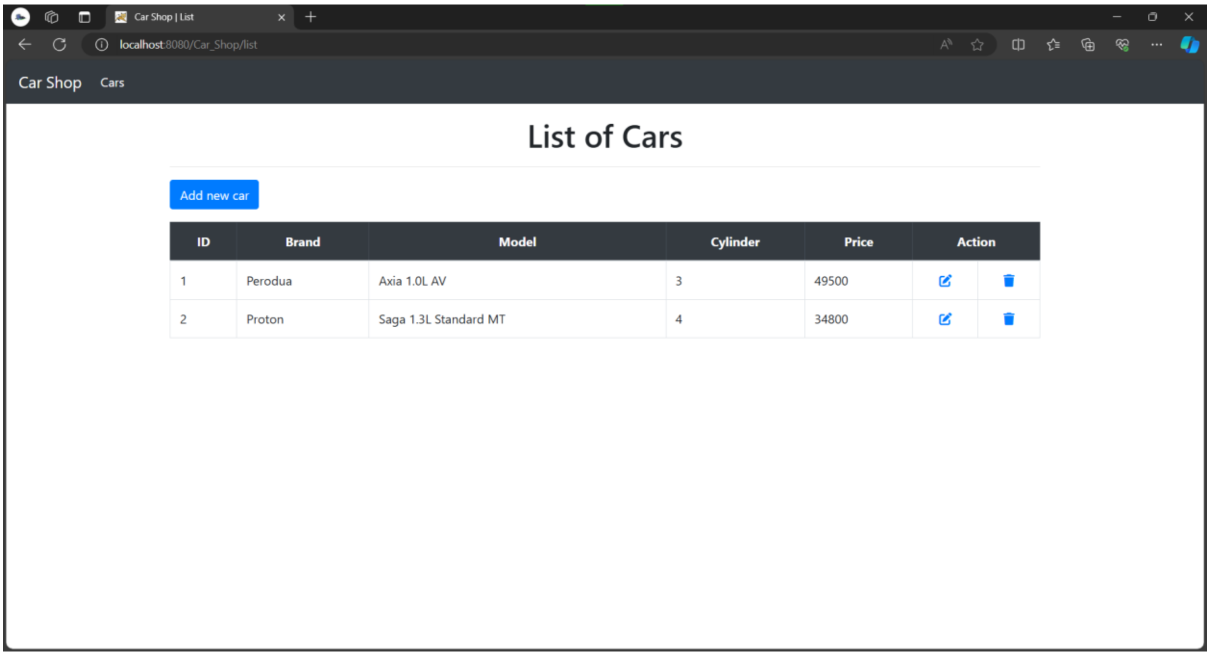
Add New Car

Brand: Proton

Model: Saga 1.3L Standard MT

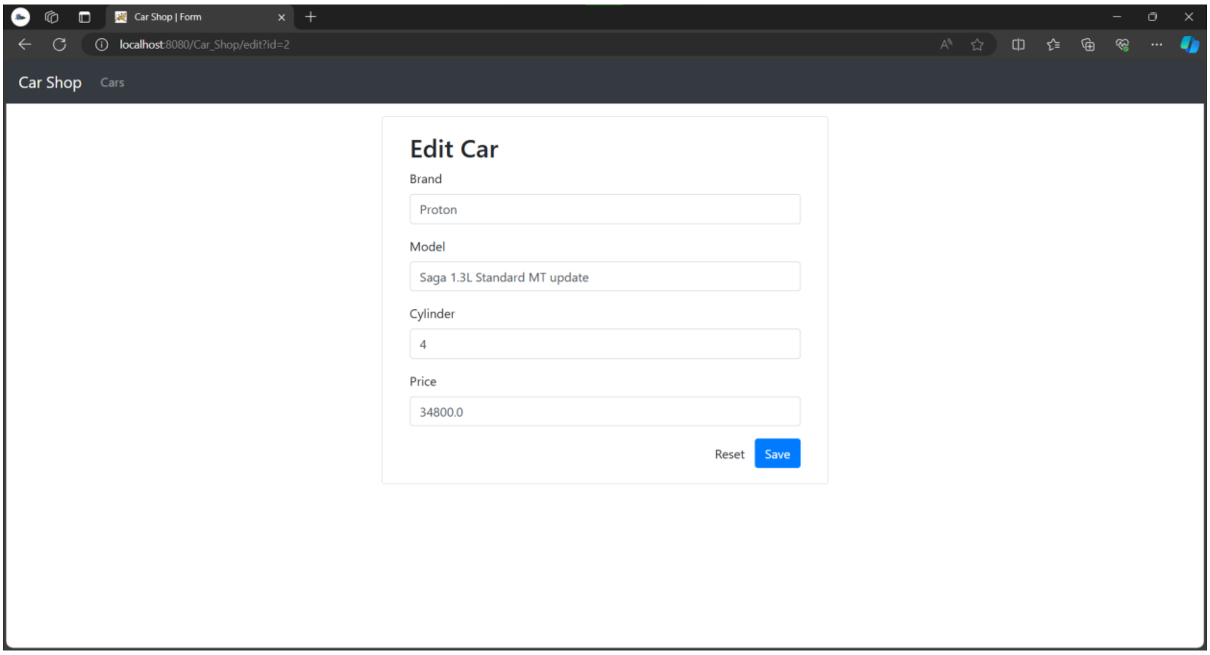
Cylinder: 4

Price: 34800



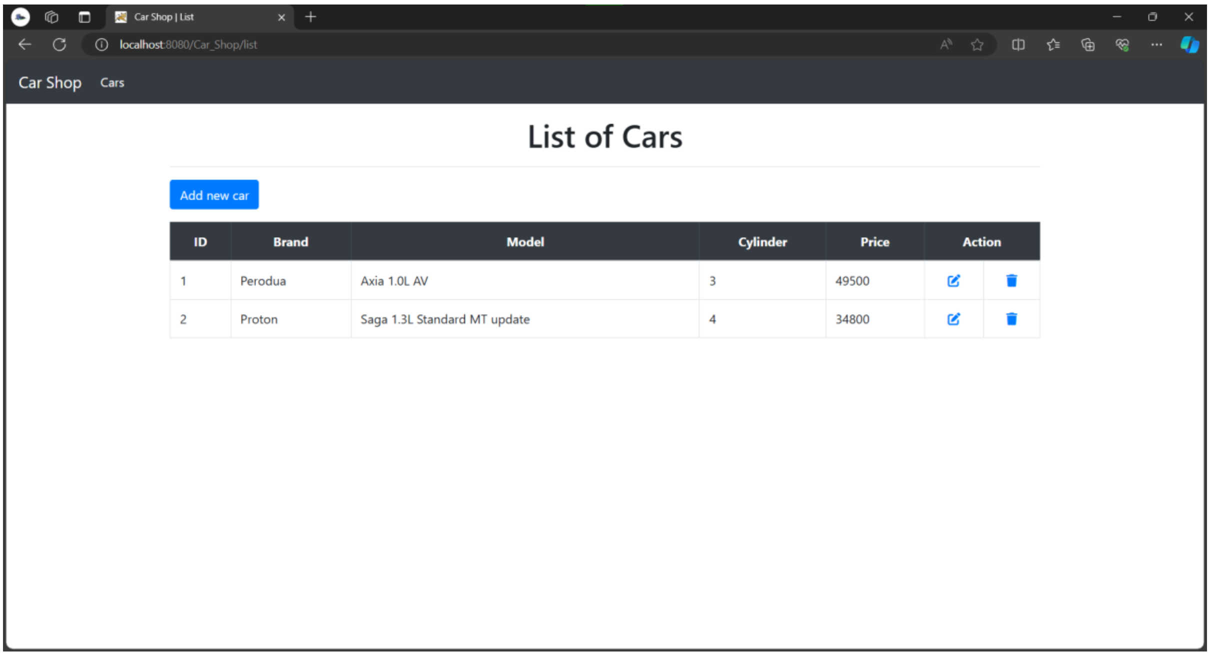
List of Cars

ID	Brand	Model	Cylinder	Price	Action	
1	Perodua	Axia 1.0L AV	3	49500	✎	🗑
2	Proton	Saga 1.3L Standard MT	4	34800	✎	🗑



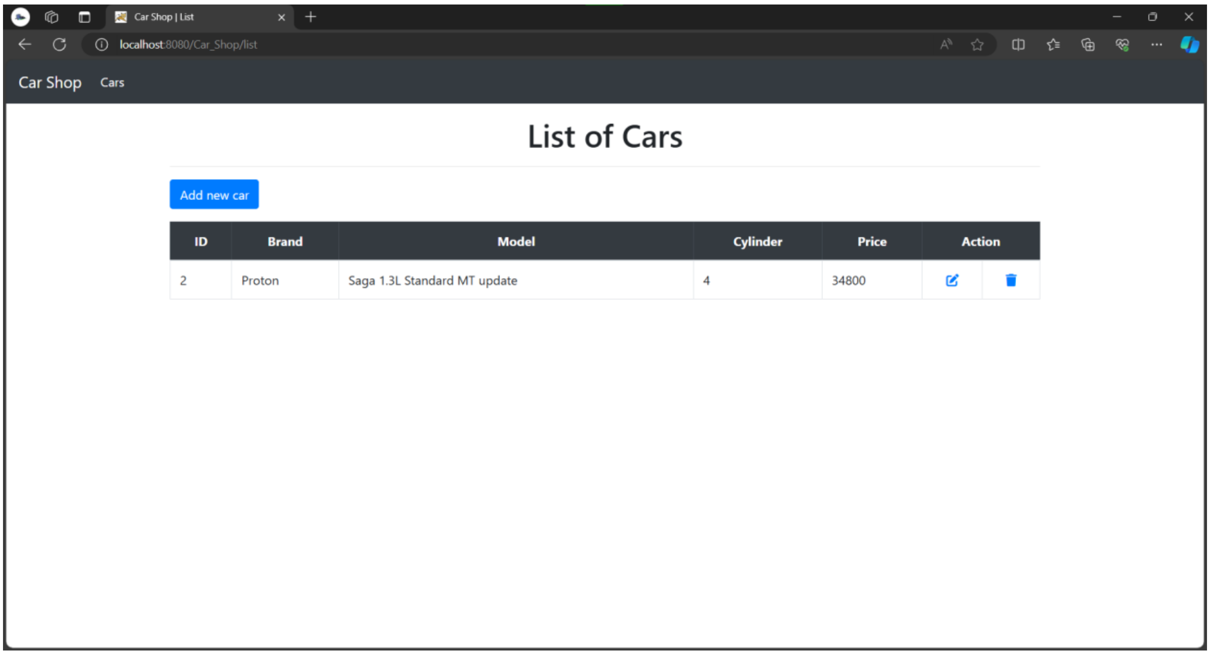
Edit Car

Brand	Proton
Model	Saga 1.3L Standard MT update
Cylinder	4
Price	34800.0



List of Cars

ID	Brand	Model	Cylinder	Price	Action	
1	Perodua	Axia 1.0L AV	3	49500	✎	🗑
2	Proton	Saga 1.3L Standard MT update	4	34800	✎	🗑



List of Cars

Add new car

ID	Brand	Model	Cylinder	Price	Action	
2	Proton	Saga 1.3L Standard MT update	4	34800		