UNIVERSITI MALAYSIA TERENGGANU

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS

CSM3023 – WEB BASED APPLICATION DEVELOPMENT

LAB REPORT 8 – JSP : AN MVC EXAMPLE WITH SERVLETS AND JSP

PREPARED BY:

HARINATUL MUFLIHUN BINTI HASNUL MUNAWAR

(S67604)

PREPARED FOR:

DR. MOHAMAD NOR BIN HASSAN

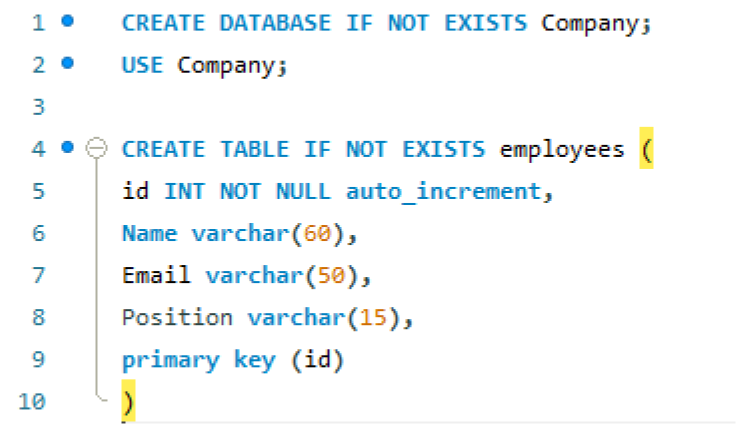
BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING)

WITH HONOURS

SEMESTER II 2023/2024

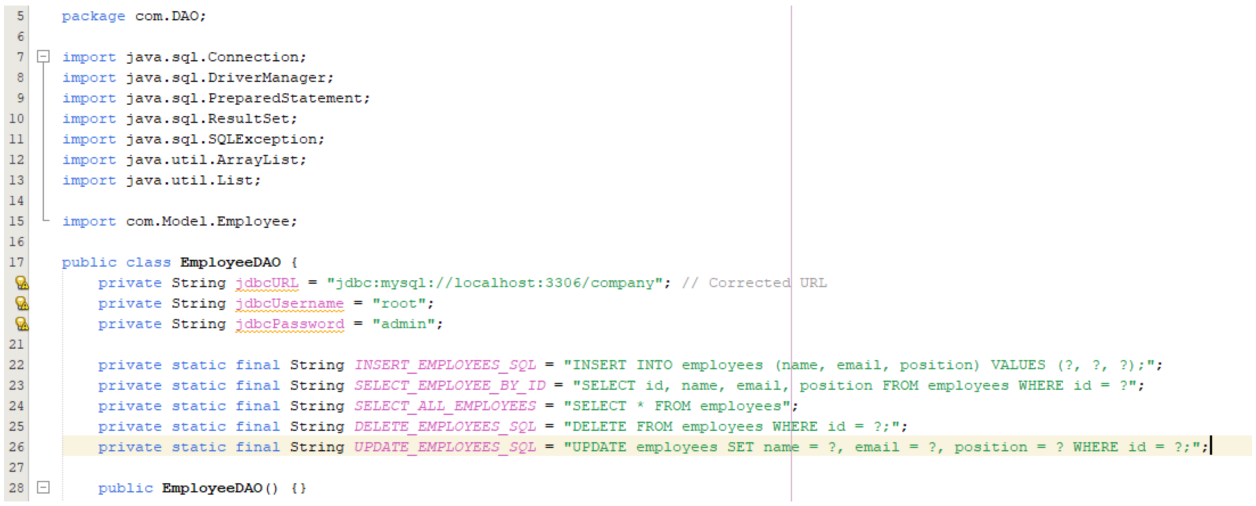
TASK 1 : CREATING MVC DATABASE WEB APPLICATION IN JSP AND SERVLETS – FOR CREATE, READ, UPDATE, DELETE

COMPANY database schema.

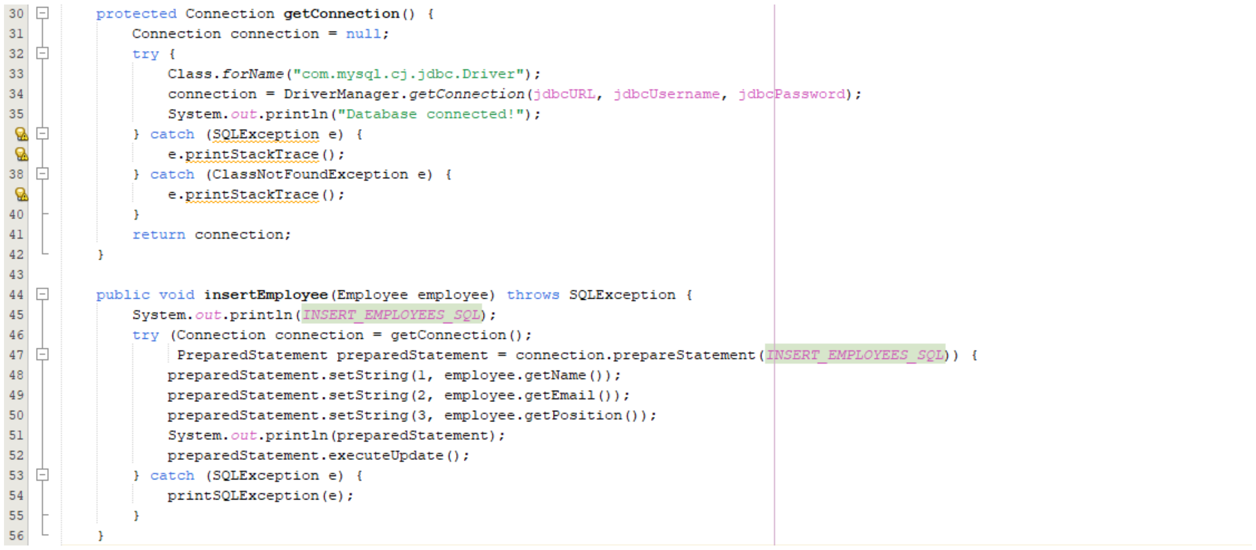


```sql
CREATE DATABASE IF NOT EXISTS Company;
USE Company;

CREATE TABLE IF NOT EXISTS employees (
id INT NOT NULL auto_increment,
Name varchar(60),
Email varchar(50),
Position varchar(15),
primary key (id)
)
```

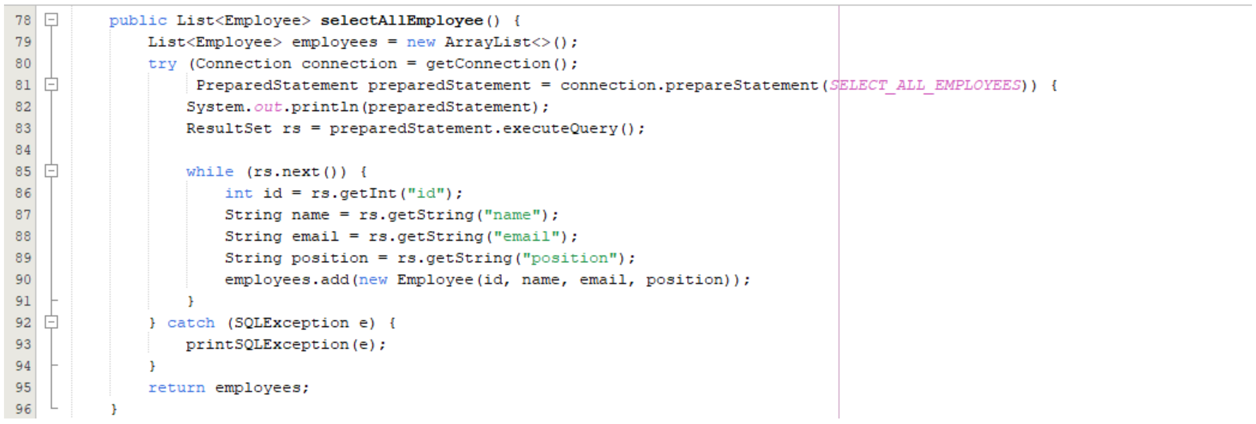
EmployeeDAO.java



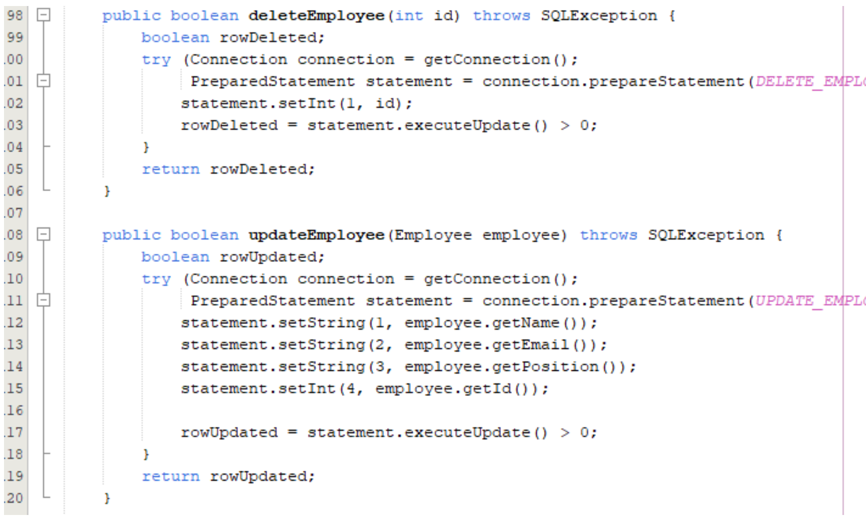
```java
package com.DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.Model.Employee;

public class EmployeeDAO {
    private String jdbcURL = "jdbc:mysql://localhost:3306/company"; // Corrected URL
    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() {}
```



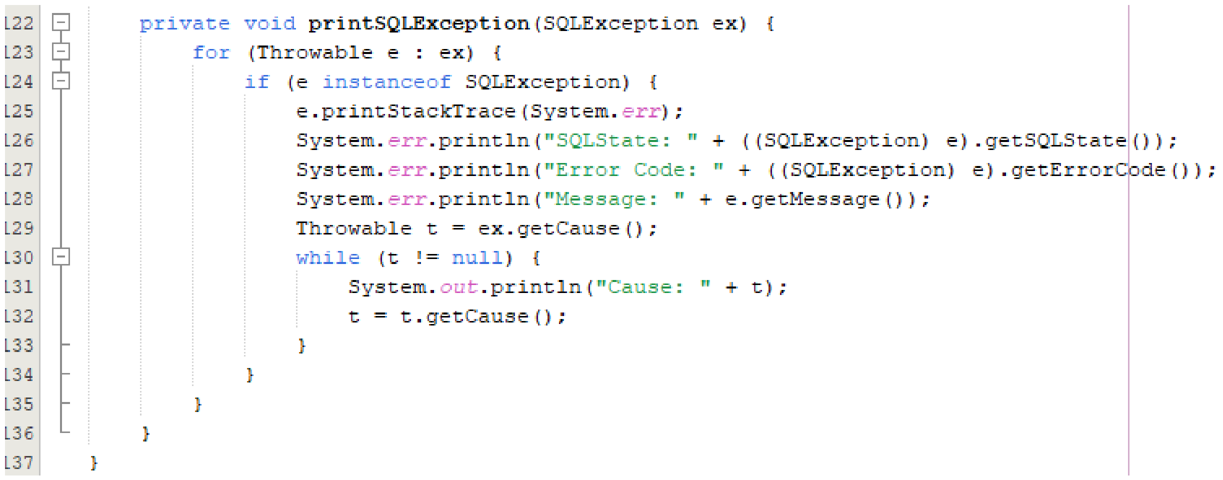
```java
    protected Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);
            System.out.println("Database connected!");
        } 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);
        }
    }
```



```java
    public List<Employee> selectAllEmployee() {
        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;
    }
```

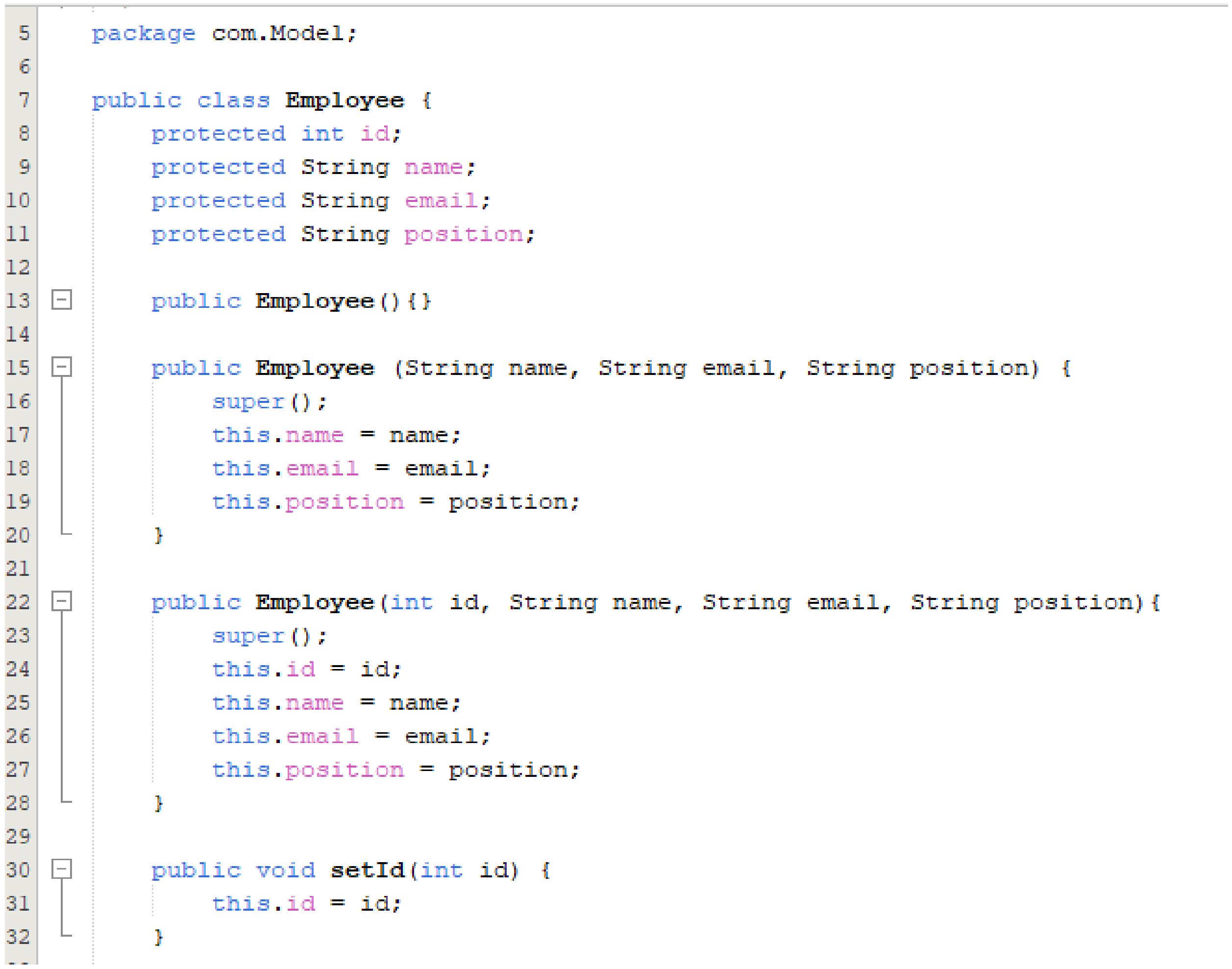


```java
    public boolean deleteEmployee(int id) throws SQLException {
        boolean rowDeleted;
        try (Connection connection = getConnection();
             PreparedStatement statement = connection.prepareStatement(DELETE_EMPLOYEES_SQL)) {
            statement.setInt(1, id);
            rowDeleted = 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;
    }
```

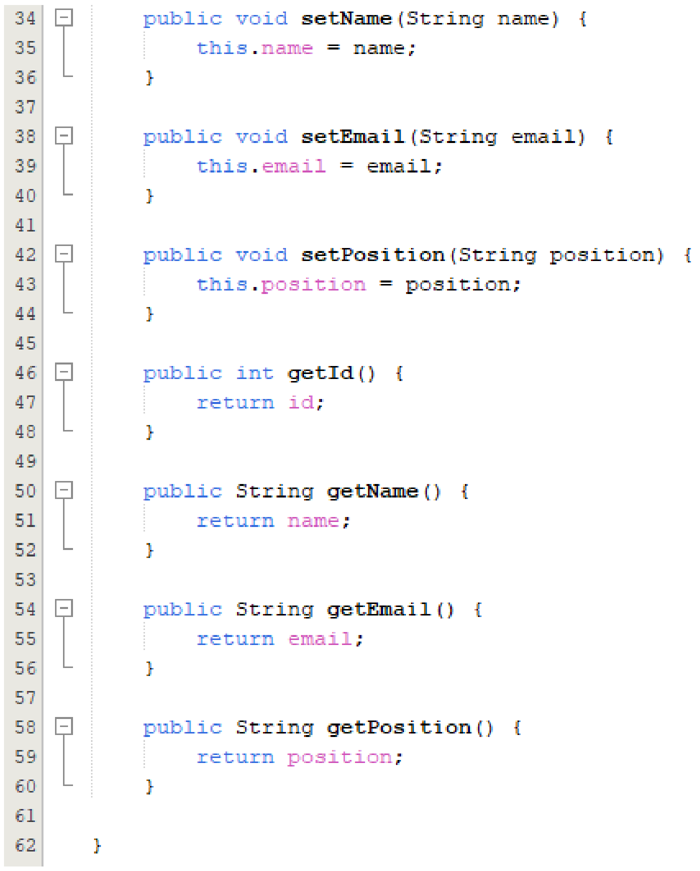


```java
    private void printSQLException(SQLException ex) {
        for (Throwable e : ex) {
            if (e instanceof SQLException) {
                e.printStackTrace(System.err);
                System.err.println("SQLState: " + ((SQLException) e).getSQLState());
                System.err.println("Error 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();
                }
            }
        }
    }
}
```

Employee.java

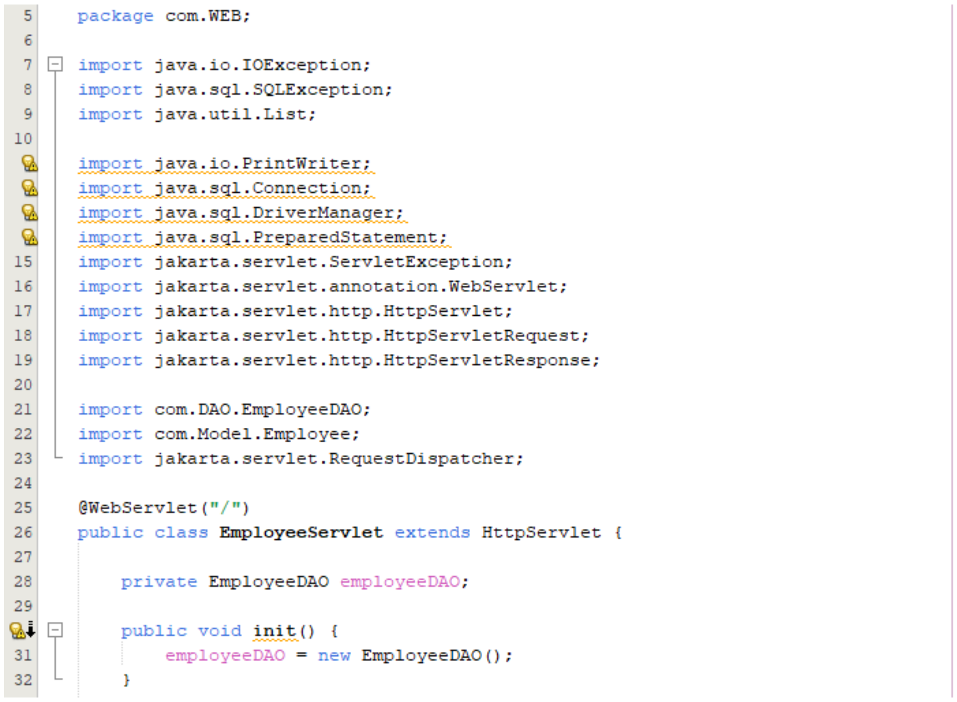


```java
package com.Model;

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){
        super();
        this.id = id;
        this.name = name;
        this.email = email;
        this.position = position;
    }

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

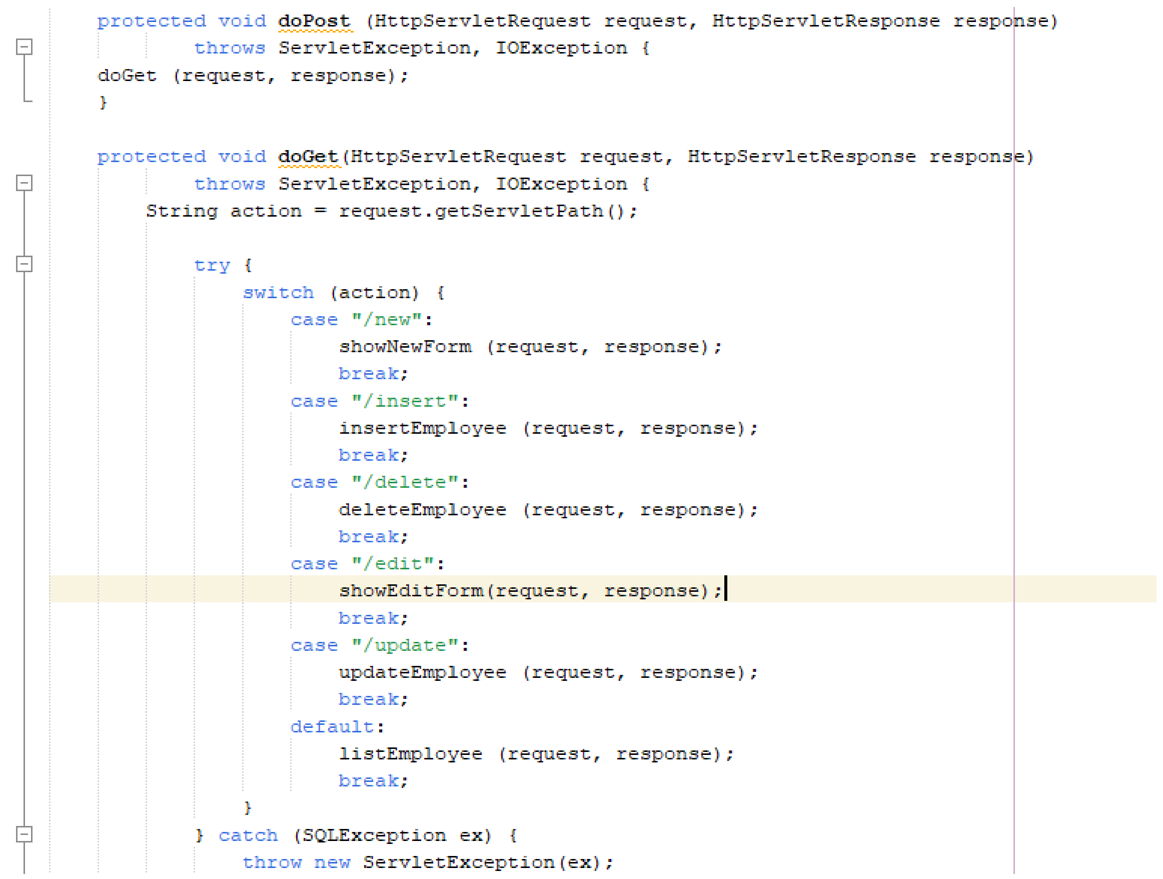


```java
    public void setName(String name) {
        this.name = name;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public void setPosition(String position) {
        this.position = position;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    public String getPosition() {
        return position;
    }

}
```

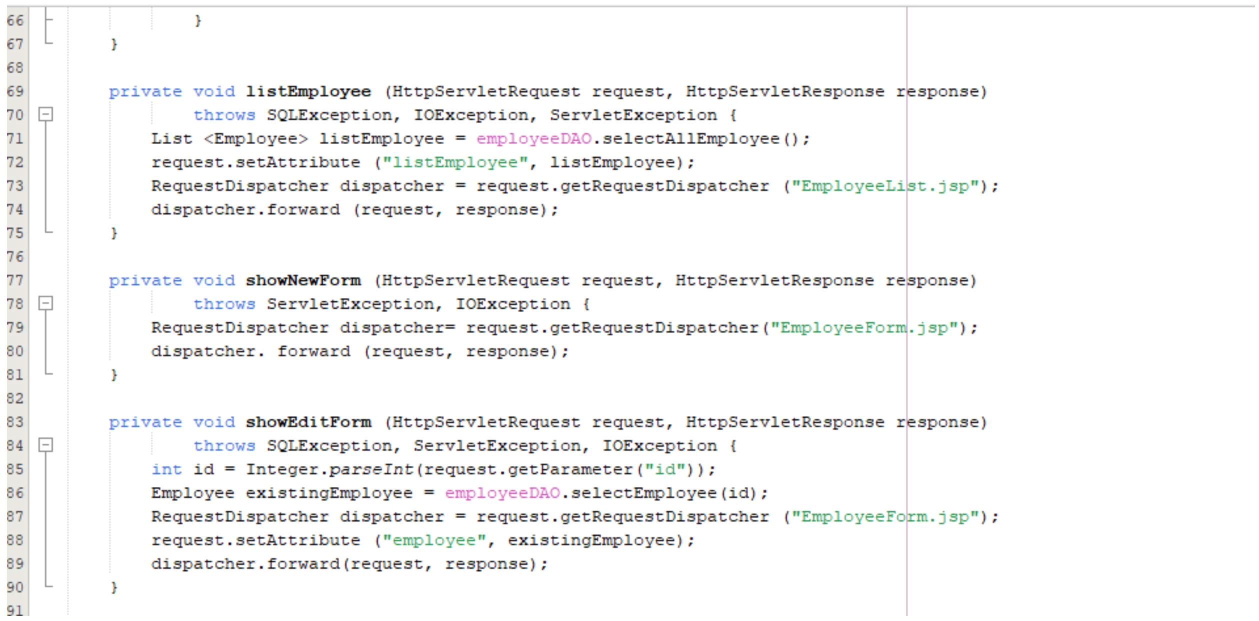
employeeServlet.java



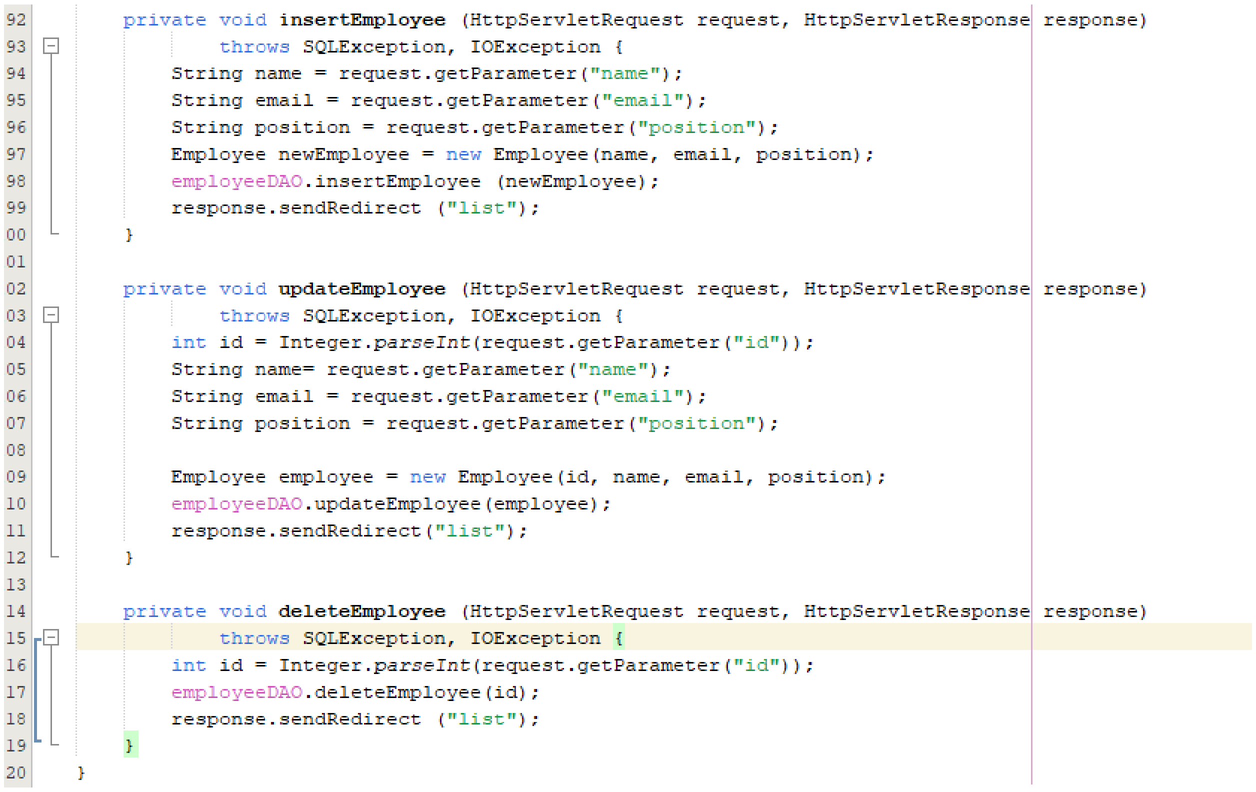
```java
package com.WEB;

import java.io.IOException;
import java.sql.SQLException;
import java.util.List;

import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import com.DAO.EmployeeDAO;
import com.Model.Employee;
import jakarta.servlet.RequestDispatcher;

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

    private EmployeeDAO employeeDAO;

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



```java
protected void doPost (HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
doGet (request, response);
}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    String action = request.getServletPath();

        try {
            switch (action) {
                case "/new":
                    showNewForm (request, response);
                    break;
                case "/insert":
                    insertEmployee (request, response);
                    break;
                case "/delete":
                    deleteEmployee (request, response);
                    break;
                case "/edit":
                    showEditForm(request, response);
                    break;
                case "/update":
                    updateEmployee (request, response);
                    break;
                default:
                    listEmployee (request, response);
                    break;
            }
        } catch (SQLException ex) {
            throw new ServletException(ex);
```

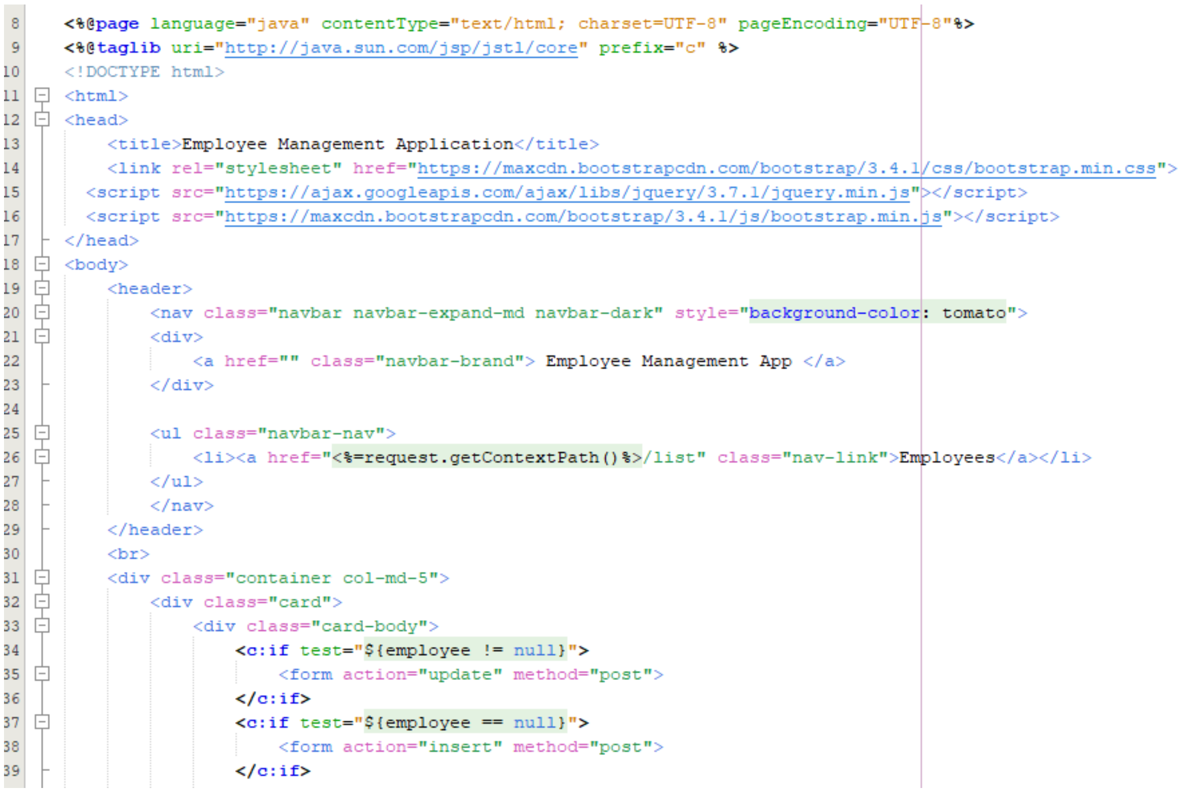


```java
            }
    }

    private void listEmployee (HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException, ServletException {
        List <Employee> listEmployee = employeeDAO.selectAllEmployee();
        request.setAttribute ("listEmployee", listEmployee);
        RequestDispatcher dispatcher = request.getRequestDispatcher ("EmployeeList.jsp");
        dispatcher.forward (request, response);
    }

    private void showNewForm (HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        RequestDispatcher dispatcher= request.getRequestDispatcher("EmployeeForm.jsp");
        dispatcher. forward (request, response);
    }

    private void showEditForm (HttpServletRequest request, HttpServletResponse response)
            throws SQLException, ServletException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
        Employee existingEmployee = employeeDAO.selectEmployee(id);
        RequestDispatcher dispatcher = request.getRequestDispatcher ("EmployeeForm.jsp");
        request.setAttribute ("employee", existingEmployee);
        dispatcher.forward(request, response);
    }

```

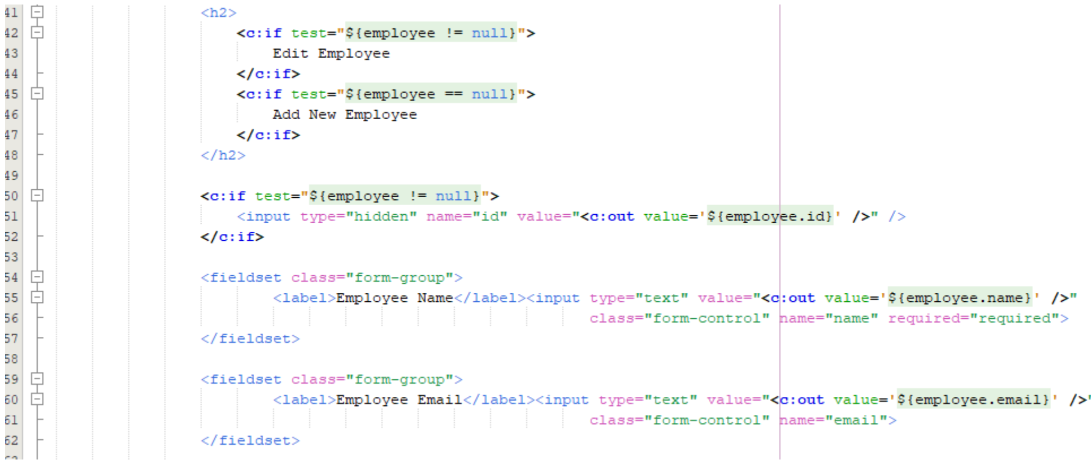


```
    private void insertEmployee (HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String position = request.getParameter("position");
        Employee newEmployee = new Employee(name, email, position);
        employeeDAO.insertEmployee (newEmployee);
        response.sendRedirect ("list");
    }

    private void updateEmployee (HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
        String name= request.getParameter("name");
        String email = request.getParameter("email");
        String position = request.getParameter("position");

        Employee employee = new Employee(id, name, email, position);
        employeeDAO.updateEmployee(employee);
        response.sendRedirect("list");
    }

    private void deleteEmployee (HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
        employeeDAO.deleteEmployee(id);
        response.sendRedirect ("list");
    }
}
```

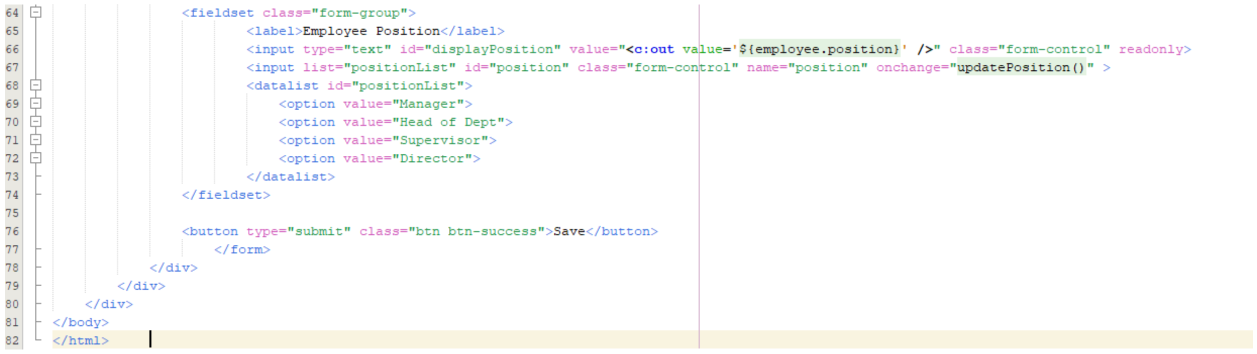
employeeForm.jsp



```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
    <title>Employee Management Application</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-md navbar-dark" style="background-color: tomato">
        <div>
            <a href="" class="navbar-brand"> Employee Management App </a>
        </div>

        <ul class="navbar-nav">
            <li><a href="<%=request.getContextPath()%>/list" class="nav-link">Employees</a></li>
        </ul>
        </nav>
    </header>
    <br>
    <div class="container col-md-5">
        <div class="card">
            <div class="card-body">
                <c:if test="${employee != null}">
                    <form action="update" method="post">
                </c:if>
                <c:if test="${employee == null}">
                    <form action="insert" method="post">
                </c:if>
```

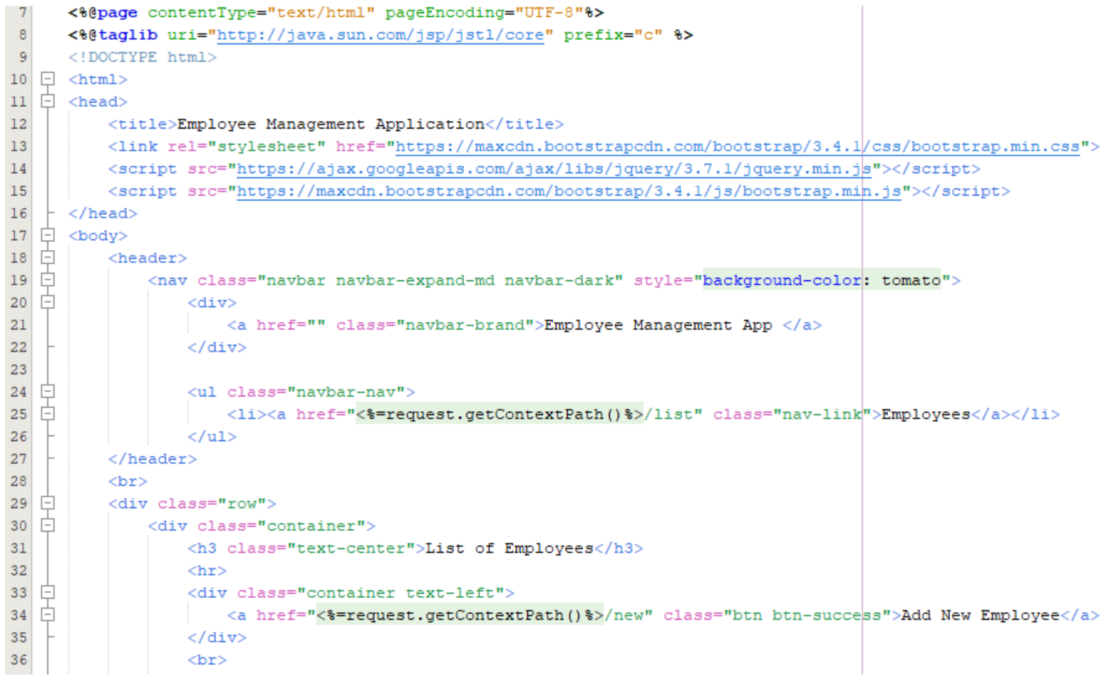


```
            <h2>
                <c:if test="${employee != null}">
                    Edit Employee
                </c:if>
                <c:if test="${employee == null}">
                    Add New Employee
                </c:if>
            </h2>

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

            <fieldset class="form-group">
                    <label>Employee Name</label><input type="text" value="<c:out value='${employee.name}' />"
                                                       class="form-control" name="name" required="required">
            </fieldset>

            <fieldset class="form-group">
                    <label>Employee Email</label><input type="text" value="<c:out value='${employee.email}' />"
                                                        class="form-control" name="email">
            </fieldset>
```

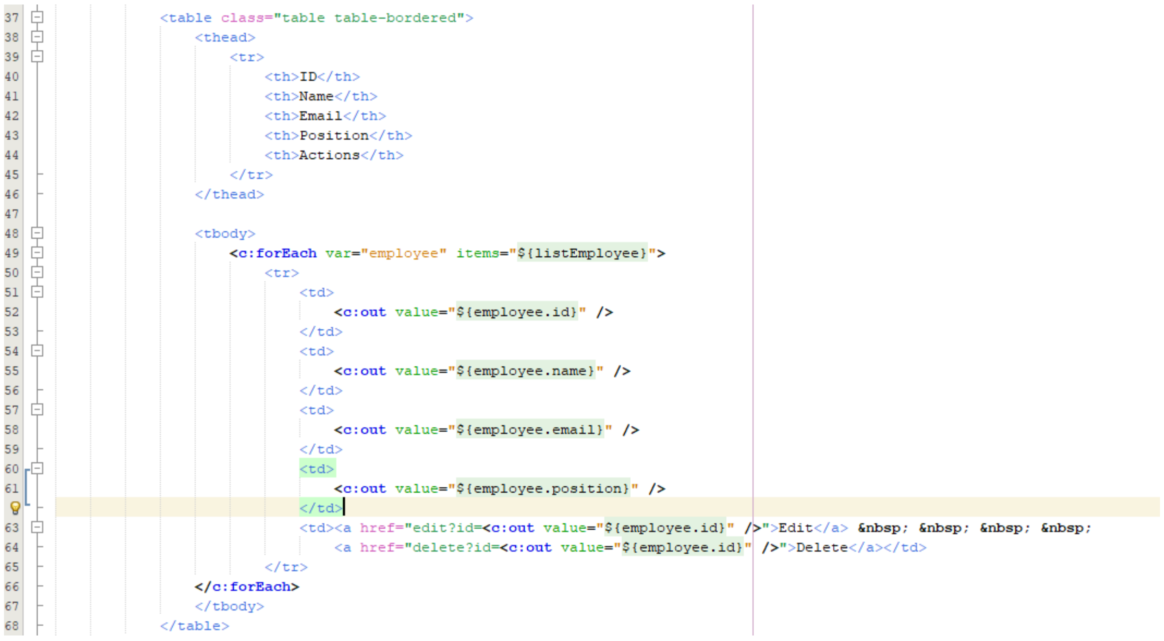


```
                    <fieldset class="form-group">
                            <label>Employee Position</label>
                            <input type="text" id="displayPosition" value="<c:out value='${employee.position}' />" class="form-control" readonly>
                            <input list="positionList" id="position" class="form-control" name="position" onchange="updatePosition()" >
                            <datalist id="positionList">
                                <option value="Manager">
                                <option value="Head of Dept">
                                <option value="Supervisor">
                                <option value="Director">
                            </datalist>
                    </fieldset>

                    <button type="submit" class="btn btn-success">Save</button>
                        </form>
                </div>
            </div>
        </div>
    </body>
</html>
```

employeeList.jsp



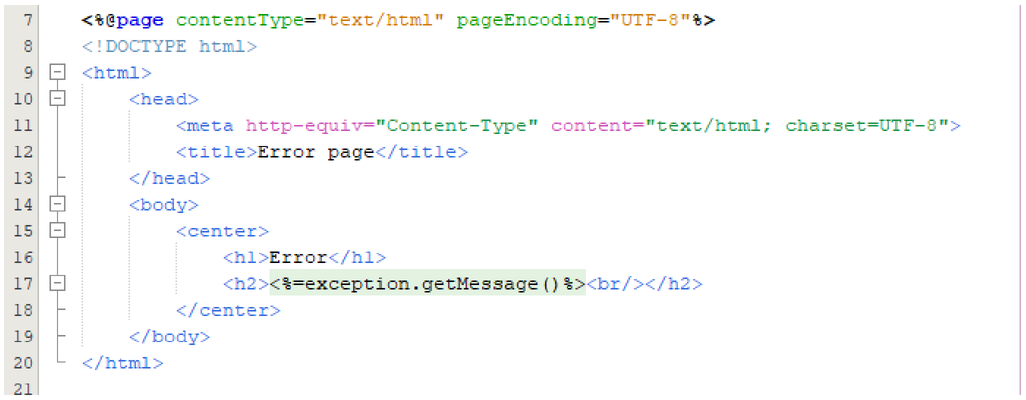
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
    <title>Employee Management Application</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</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>
    </header>
    <br>
    <div class="row">
        <div class="container">
            <h3 class="text-center">List of Employees</h3>
            <hr>
            <div class="container text-left">
                <a href="<%=request.getContextPath()%>/new" class="btn btn-success">Add New Employee</a>
            </div>
            <br>
```



```
<table class="table table-bordered">
    <thead>
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Email</th>
            <th>Position</th>
            <th>Actions</th>
        </tr>
    </thead>

    <tbody>
        <c:forEach var="employee" items="${listEmployee}">
            <tr>
                <td>
                    <c:out value="${employee.id}" />
                </td>
                <td>
                    <c:out value="${employee.name}" />
                </td>
                <td>
                    <c:out value="${employee.email}" />
                </td>
                <td>
                    <c:out value="${employee.position}" />
                </td>
                <td><a href="edit?id=<c:out value="${employee.id}" />">Edit</a> &nbsp; &nbsp; &nbsp; &nbsp;
                    <a href="delete?id=<c:out value="${employee.id}" />">Delete</a></td>
            </tr>
        </c:forEach>
    </tbody>
</table>
```

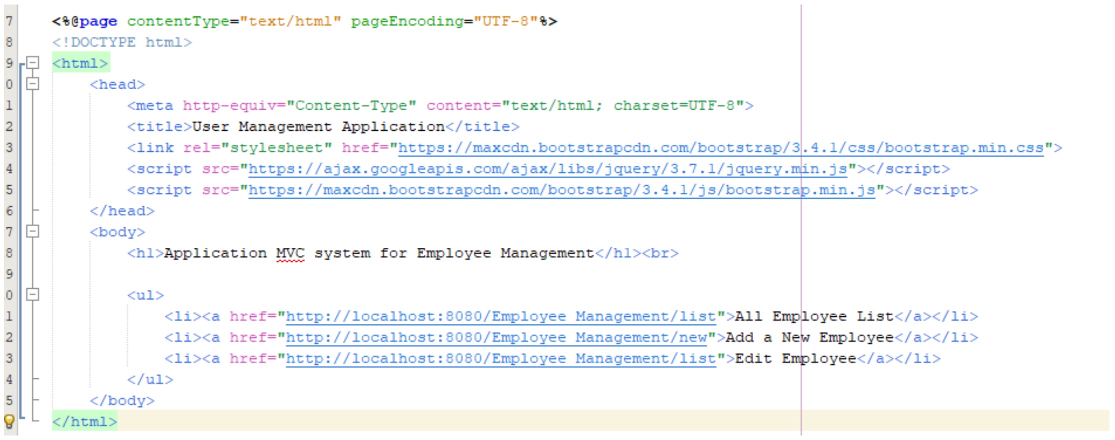
```
        </div>
    </div>
</body>
</html>
```

Error.jsp



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

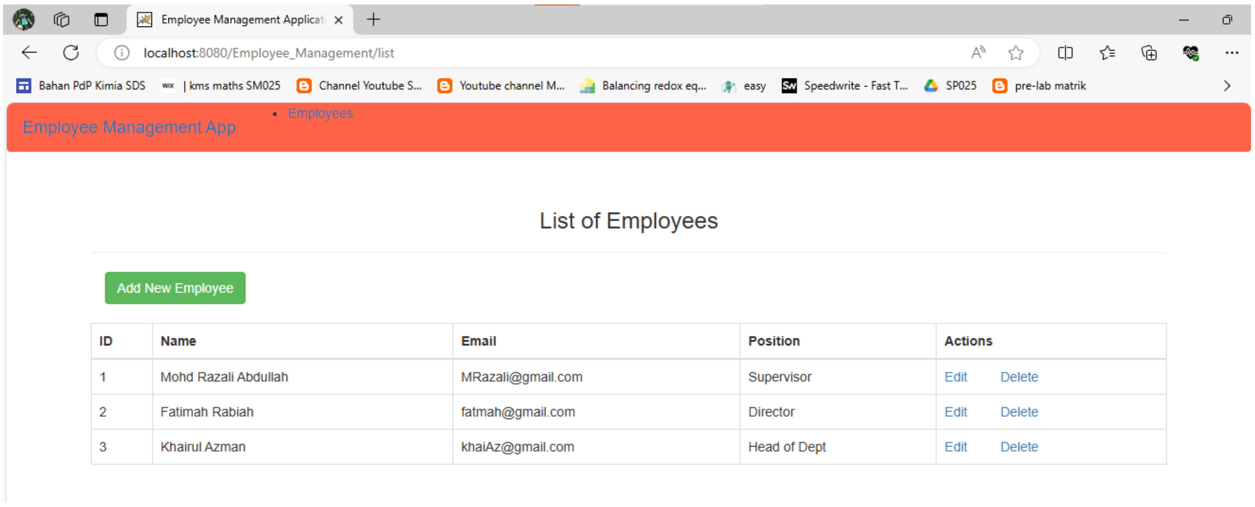
Index.jsp



```
<%@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://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
    </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>
```

The output.

- This is the list data that I insert.



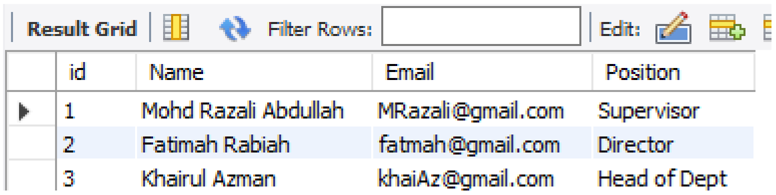
localhost:8080/Employee_Management/list

Employee Management App • Employees

List of Employees

Add New Employee

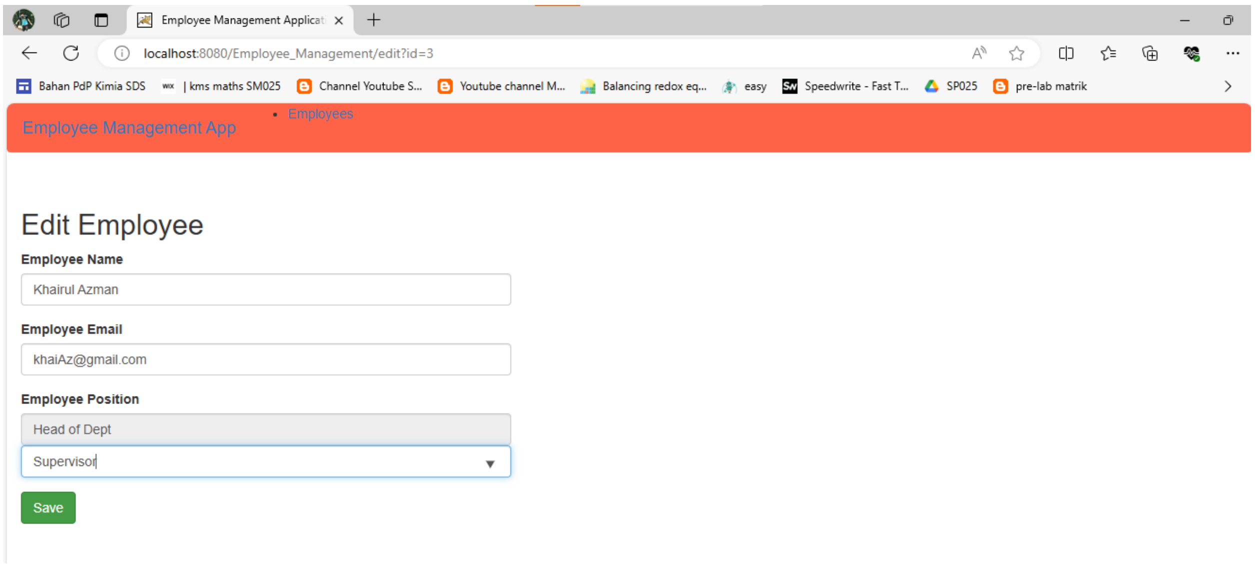
ID	Name	Email	Position	Actions
1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor	Edit Delete
2	Fatimah Rabiah	fatmah@gmail.com	Director	Edit Delete
3	Khairul Azman	khaiAz@gmail.com	Head of Dept	Edit Delete



Result Grid | Filter Rows: | Edit:

id	Name	Email	Position
1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor
2	Fatimah Rabiah	fatmah@gmail.com	Director
3	Khairul Azman	khaiAz@gmail.com	Head of Dept

- The data that I edit



localhost:8080/Employee_Management/edit?id=3

Employee Management App • Employees

Edit Employee

Employee Name: Khairul Azman

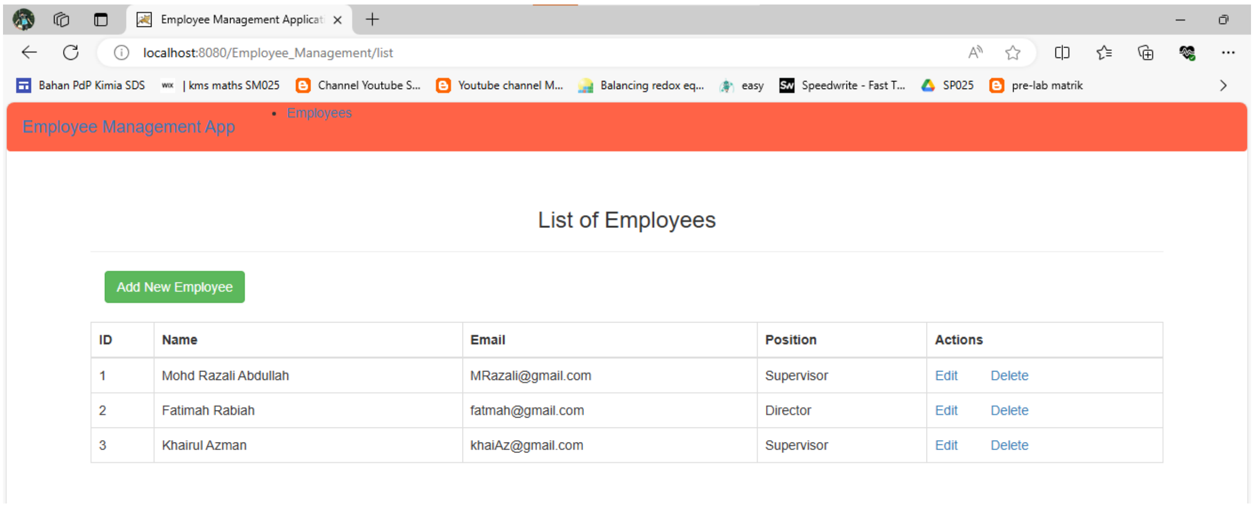
Employee Email: khaiAz@gmail.com

Employee Position: Head of Dept

Supervisor

Save

- The data after edit



localhost:8080/Employee_Management/list

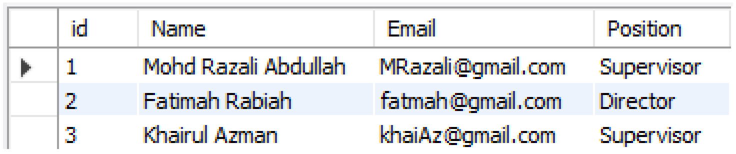
Employee Management App

- Employees

List of Employees

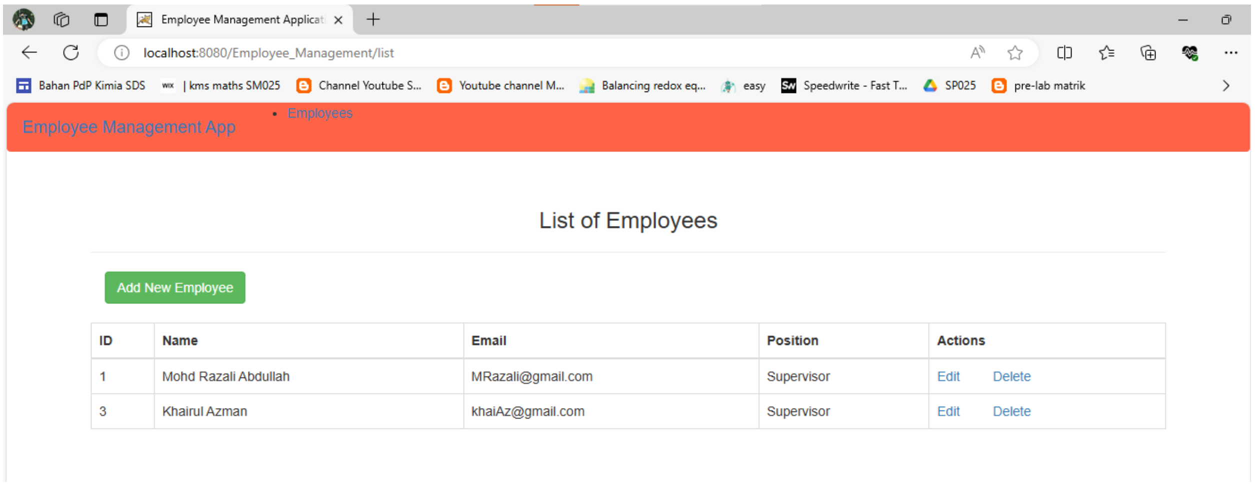
Add New Employee

ID	Name	Email	Position	Actions
1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor	Edit Delete
2	Fatimah Rabiah	fatmah@gmail.com	Director	Edit Delete
3	Khairul Azman	khaiAz@gmail.com	Supervisor	Edit Delete



	id	Name	Email	Position
▶	1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor
	2	Fatimah Rabiah	fatmah@gmail.com	Director
	3	Khairul Azman	khaiAz@gmail.com	Supervisor

- The data when I click delete button.



localhost:8080/Employee_Management/list

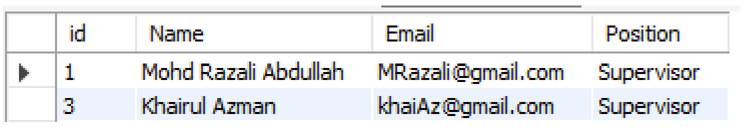
Employee Management App

- Employees

List of Employees

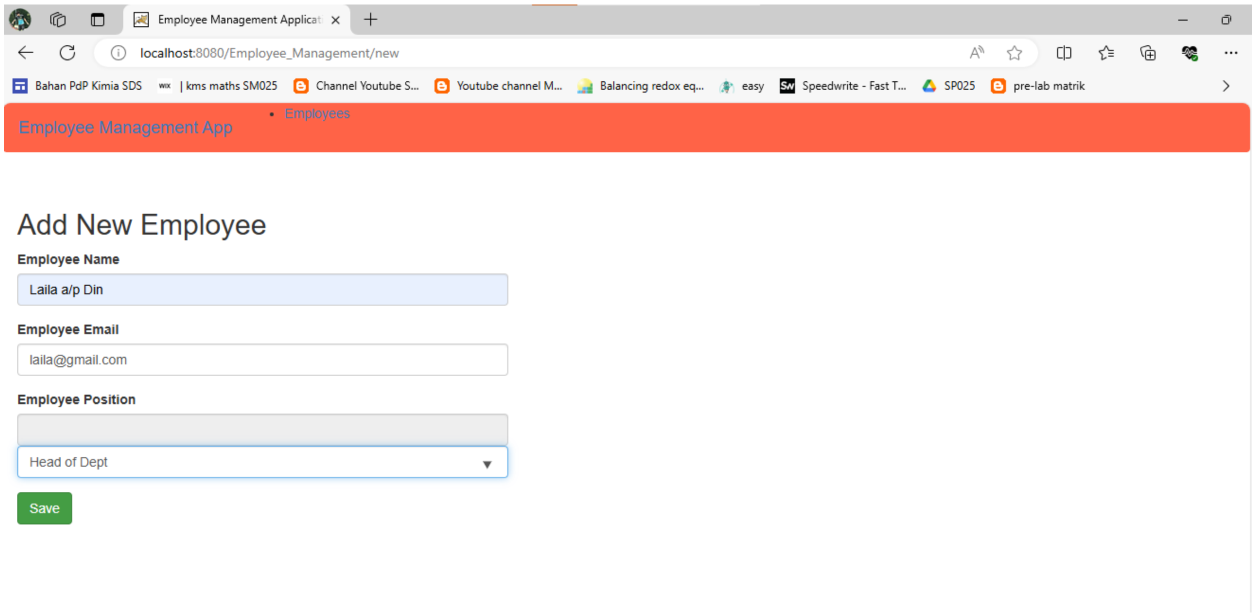
Add New Employee

ID	Name	Email	Position	Actions
1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor	Edit Delete
3	Khairul Azman	khaiAz@gmail.com	Supervisor	Edit Delete



	id	Name	Email	Position
▶	1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor
	3	Khairul Azman	khaiAz@gmail.com	Supervisor

- The output add new employee.



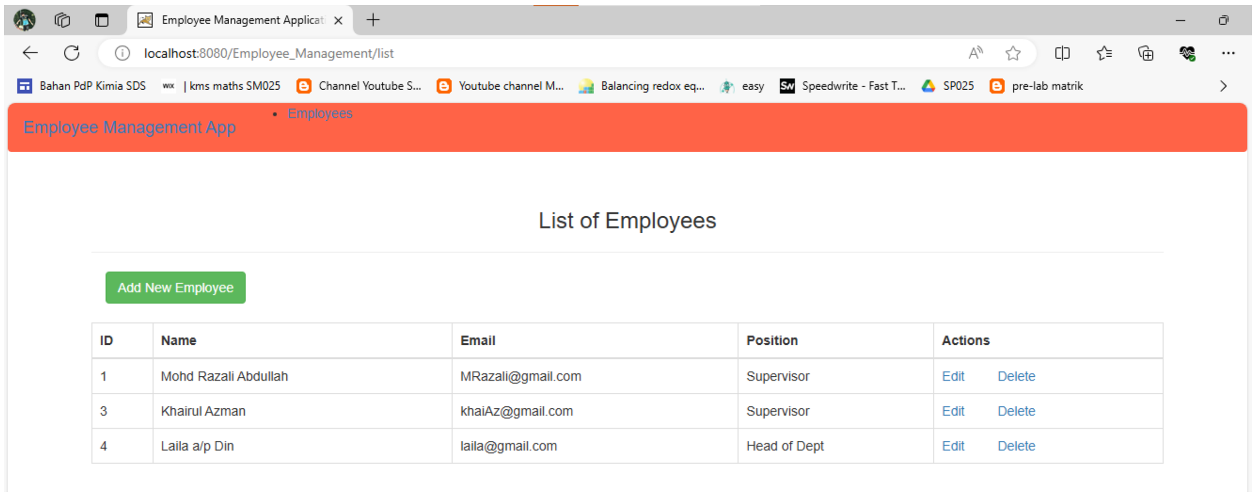
localhost:8080/Employee_Management/new

Employee Management App · Employees

Add New Employee

Employee Name	Laila a/p Din
Employee Email	laila@gmail.com
Employee Position	Head of Dept

Save



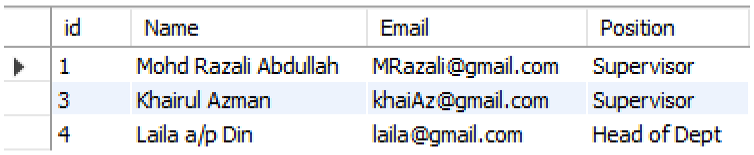
localhost:8080/Employee_Management/list

Employee Management App · Employees

List of Employees

Add New Employee

ID	Name	Email	Position	Actions
1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor	Edit Delete
3	Khairul Azman	khaiAz@gmail.com	Supervisor	Edit Delete
4	Laila a/p Din	laila@gmail.com	Head of Dept	Edit Delete



id	Name	Email	Position
1	Mohd Razali Abdullah	MRazali@gmail.com	Supervisor
3	Khairul Azman	khaiAz@gmail.com	Supervisor
4	Laila a/p Din	laila@gmail.com	Head of Dept

EXERCISE

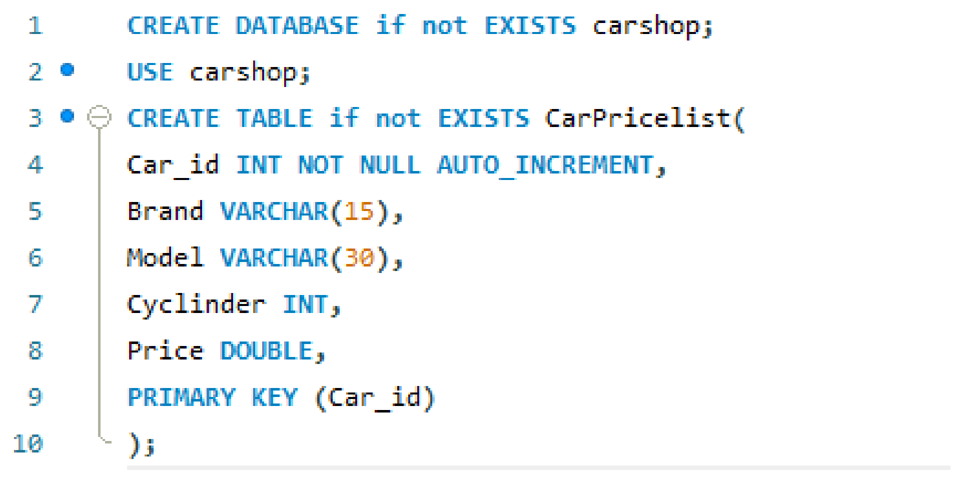
Using this database shema, please create MVC Application [CRUD] for Car Shop, using JSP, Servlet, and MySQL.

```
CREATE DATABASE if not EXISTS carshop;

USE carshop;

CREATE TABLE if not EXISTS CarPricelist(

    Car_id INT NOT NULL AUTO_INCREMENT,

    Brand VARCHAR(15),

    Model VARCHAR(30),

    Cyclinder INT,

    Price DOUBLE,

    PRIMARY KEY (Car_id)

);
```

The application should be able to handle these activities:

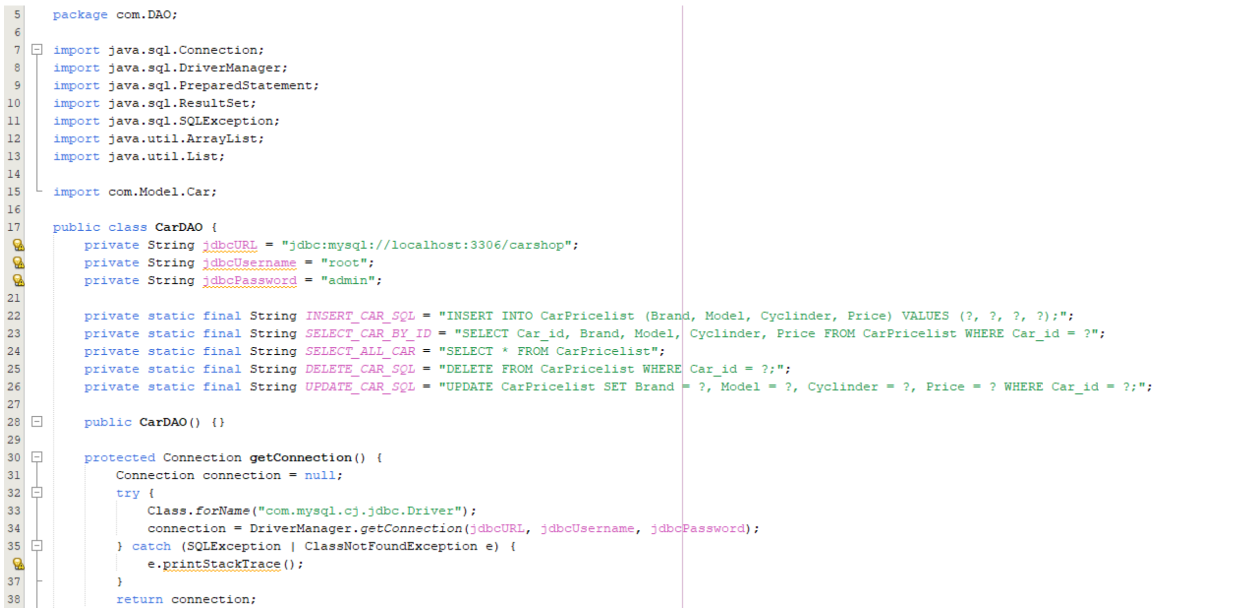
1. View all data
2. Add new data
3. Edit/Update current data
4. Delete the specific data

Carshop scheme.

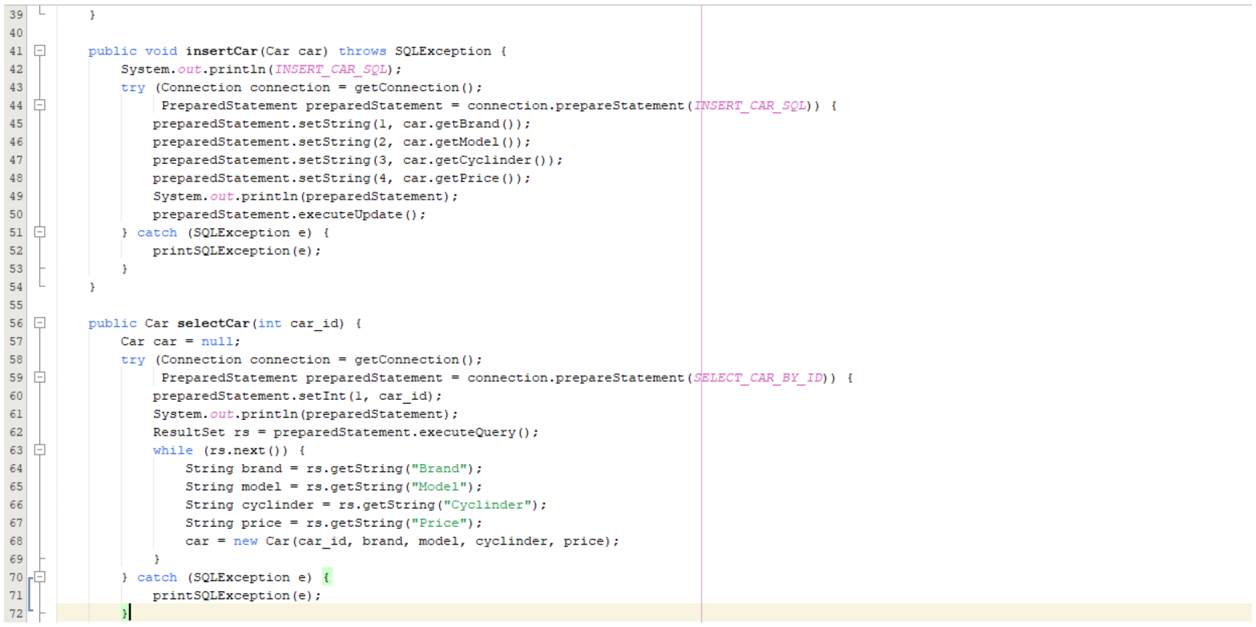


```
CREATE DATABASE if not EXISTS carshop;
USE carshop;
CREATE TABLE if not EXISTS CarPricelist(
Car_id INT NOT NULL AUTO_INCREMENT,
Brand VARCHAR(15),
Model VARCHAR(30),
Cyclinder INT,
Price DOUBLE,
PRIMARY KEY (Car_id)
);
```

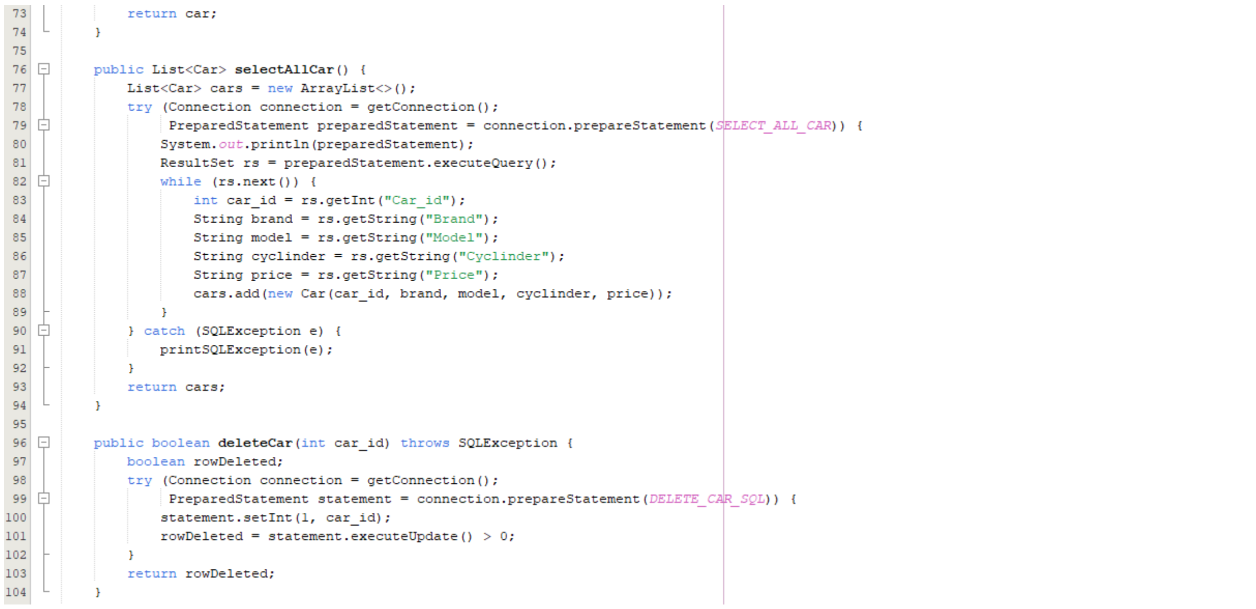
CarDAO.java



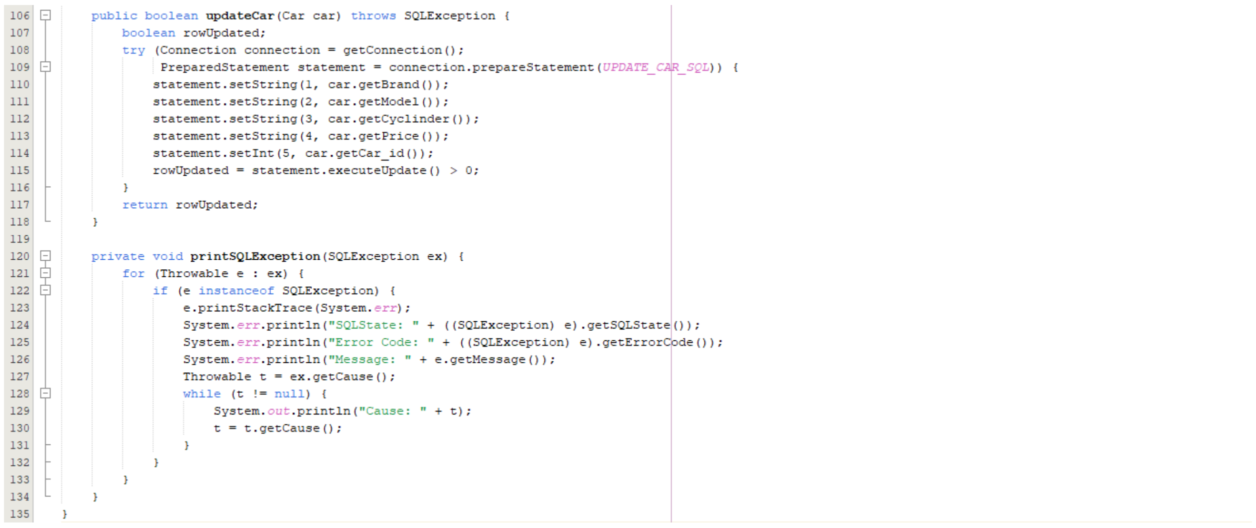
```java
package com.DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.Model.Car;

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

    private static final String INSERT_CAR_SQL = "INSERT INTO CarPricelist (Brand, Model, Cyclinder, Price) VALUES (?, ?, ?, ?);";
    private static final String SELECT_CAR_BY_ID = "SELECT Car_id, Brand, Model, Cyclinder, Price FROM CarPricelist WHERE Car_id = ?";
    private static final String SELECT_ALL_CAR = "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 = ?, Cyclinder = ?, Price = ? WHERE Car_id = ?;";

    public CarDAO() {}

    protected Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        }
        return connection;
```



```java
    }

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

    public Car selectCar(int car_id) {
        Car car = null;
        try (Connection connection = getConnection();
             PreparedStatement preparedStatement = connection.prepareStatement(SELECT_CAR_BY_ID)) {
            preparedStatement.setInt(1, car_id);
            System.out.println(preparedStatement);
            ResultSet rs = preparedStatement.executeQuery();
            while (rs.next()) {
                String brand = rs.getString("Brand");
                String model = rs.getString("Model");
                String cyclinder = rs.getString("Cyclinder");
                String price = rs.getString("Price");
                car = new Car(car_id, brand, model, cyclinder, price);
            }
        } catch (SQLException e) {
            printSQLException(e);
        }
```

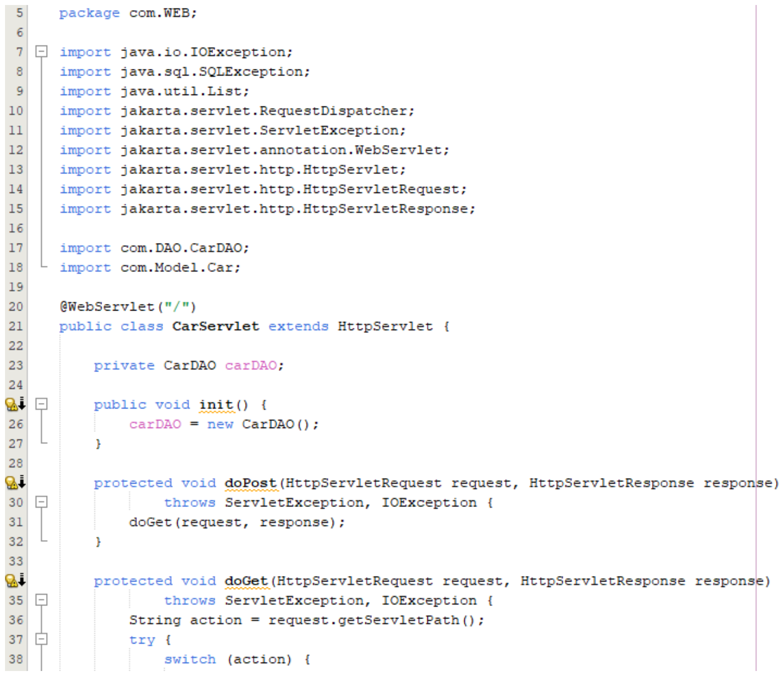


```java
        return car;
    }

    public List<Car> selectAllCar() {
        List<Car> cars = new ArrayList<>();
        try (Connection connection = getConnection();
             PreparedStatement preparedStatement = connection.prepareStatement(SELECT_ALL_CAR)) {
            System.out.println(preparedStatement);
            ResultSet rs = preparedStatement.executeQuery();
            while (rs.next()) {
                int car_id = rs.getInt("Car_id");
                String brand = rs.getString("Brand");
                String model = rs.getString("Model");
                String cyclinder = rs.getString("Cyclinder");
                String price = rs.getString("Price");
                cars.add(new Car(car_id, brand, model, cyclinder, price));
            }
        } catch (SQLException e) {
            printSQLException(e);
        }
        return cars;
    }

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

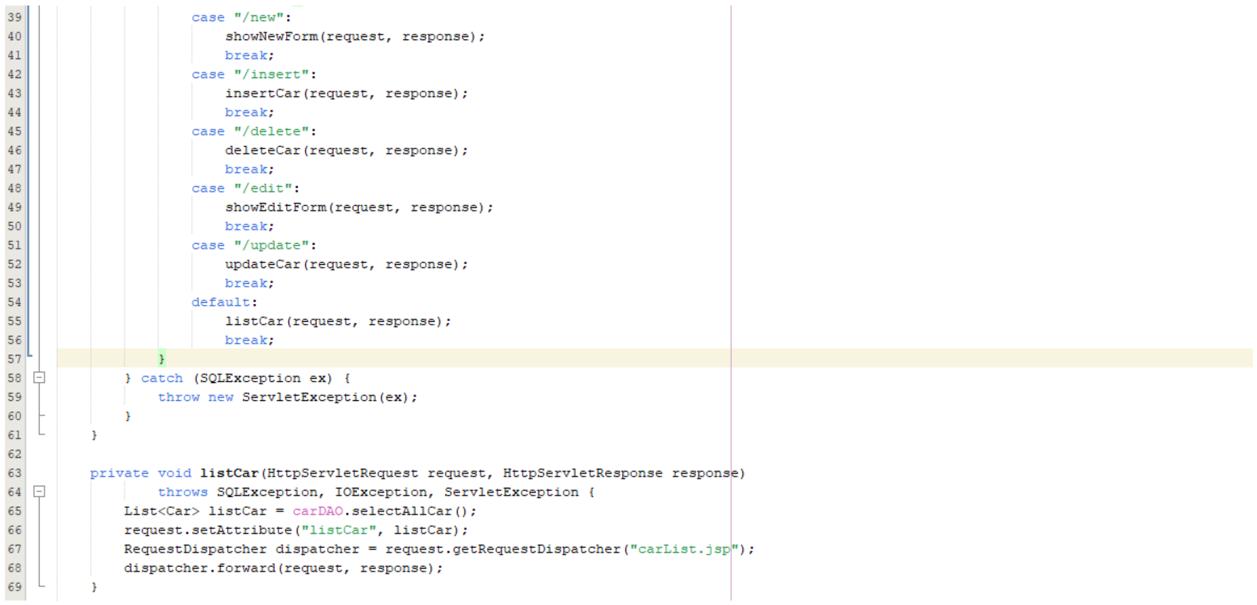


```java
    public boolean updateCar(Car car) throws SQLException {
        boolean rowUpdated;
        try (Connection connection = getConnection();
             PreparedStatement statement = connection.prepareStatement(UPDATE_CAR_SQL)) {
            statement.setString(1, car.getBrand());
            statement.setString(2, car.getModel());
            statement.setString(3, car.getCyclinder());
            statement.setString(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 (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();
                }
            }
        }
    }
}
```

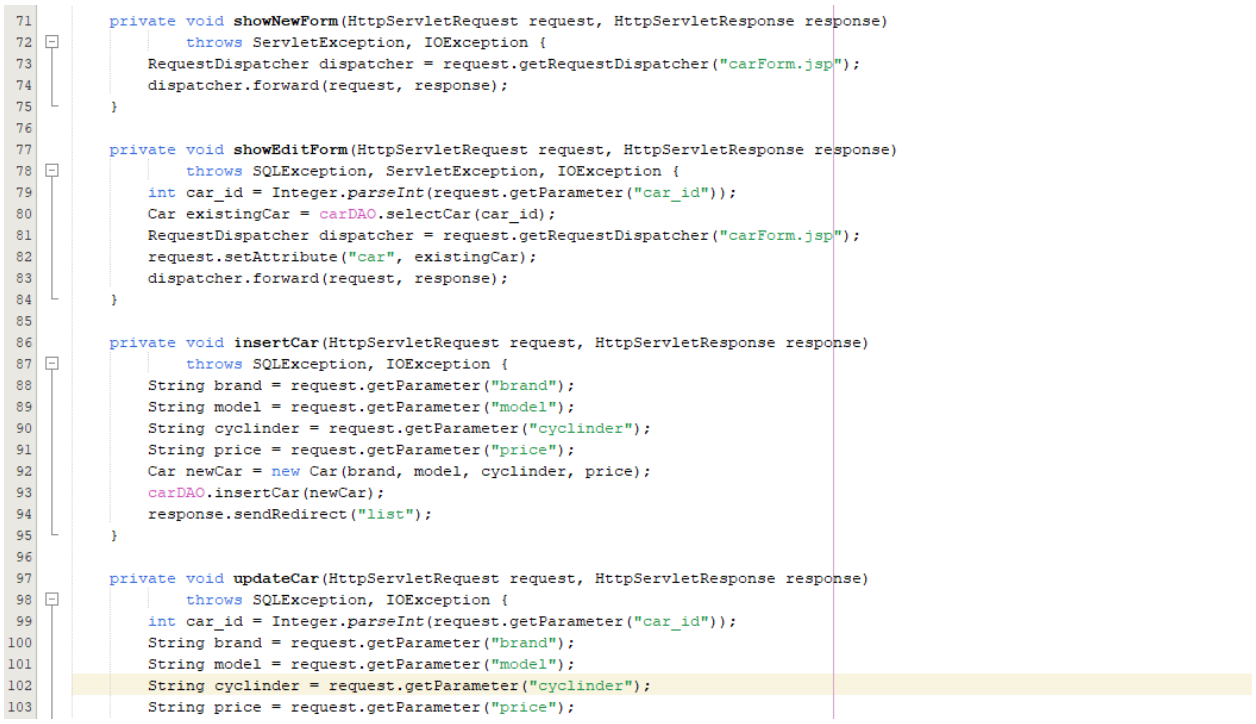
CarServlet.java



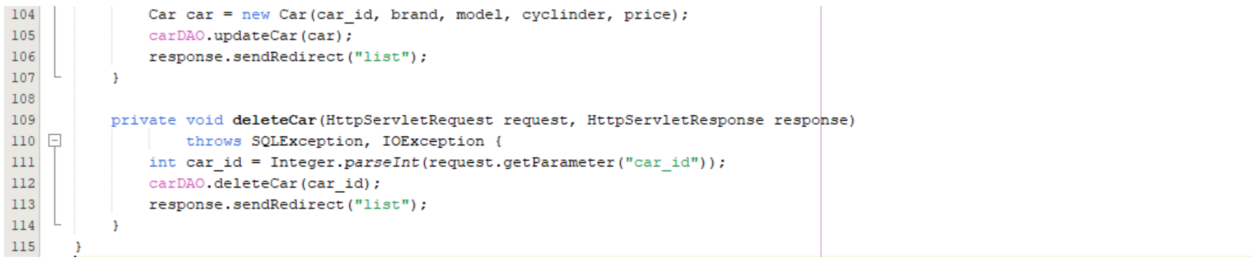
```java
package com.WEB;

import java.io.IOException;
import java.sql.SQLException;
import java.util.List;
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import com.DAO.CarDAO;
import com.Model.Car;

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

    private CarDAO carDAO;

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

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

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String action = request.getServletPath();
        try {
            switch (action) {
```



```java
                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);
                    break;
            }
        } catch (SQLException ex) {
            throw new ServletException(ex);
        }
    }

    private void listCar(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException, ServletException {
        List<Car> listCar = carDAO.selectAllCar();
        request.setAttribute("listCar", listCar);
        RequestDispatcher dispatcher = request.getRequestDispatcher("carList.jsp");
        dispatcher.forward(request, response);
    }
```

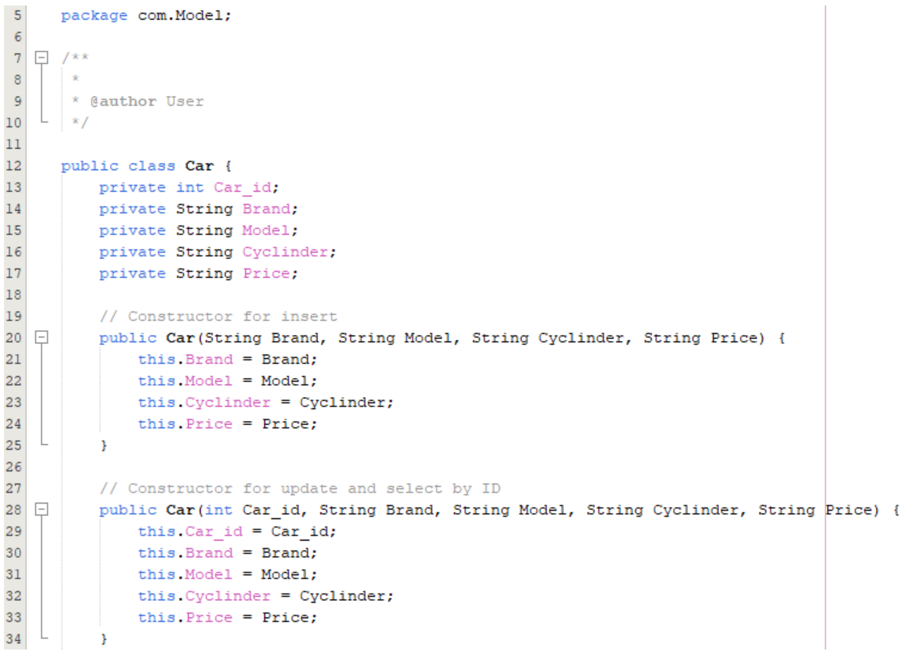


```java
    private void showNewForm(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        RequestDispatcher dispatcher = request.getRequestDispatcher("carForm.jsp");
        dispatcher.forward(request, response);
    }

    private void showEditForm(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, ServletException, IOException {
        int car_id = Integer.parseInt(request.getParameter("car_id"));
        Car existingCar = carDAO.selectCar(car_id);
        RequestDispatcher dispatcher = request.getRequestDispatcher("carForm.jsp");
        request.setAttribute("car", existingCar);
        dispatcher.forward(request, response);
    }

    private void insertCar(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        String brand = request.getParameter("brand");
        String model = request.getParameter("model");
        String cyclinder = request.getParameter("cyclinder");
        String price = request.getParameter("price");
        Car newCar = new Car(brand, model, cyclinder, price);
        carDAO.insertCar(newCar);
        response.sendRedirect("list");
    }

    private void updateCar(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        int car_id = Integer.parseInt(request.getParameter("car_id"));
        String brand = request.getParameter("brand");
        String model = request.getParameter("model");
        String cyclinder = request.getParameter("cyclinder");
        String price = request.getParameter("price");
```

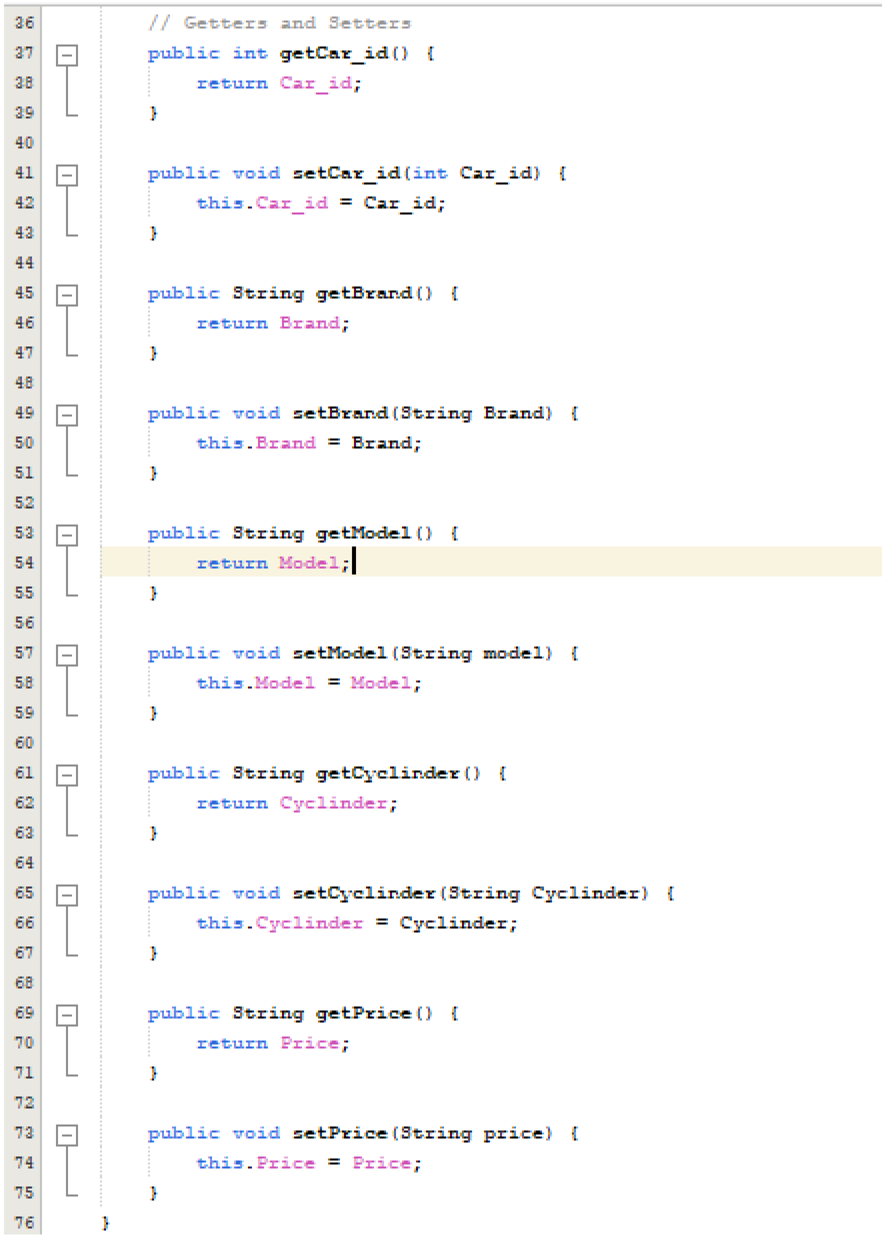


```java
        Car car = new Car(car_id, brand, model, cyclinder, price);
        carDAO.updateCar(car);
        response.sendRedirect("list");
    }

    private void deleteCar(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        int car_id = Integer.parseInt(request.getParameter("car_id"));
        carDAO.deleteCar(car_id);
        response.sendRedirect("list");
    }
}
```

Car,java

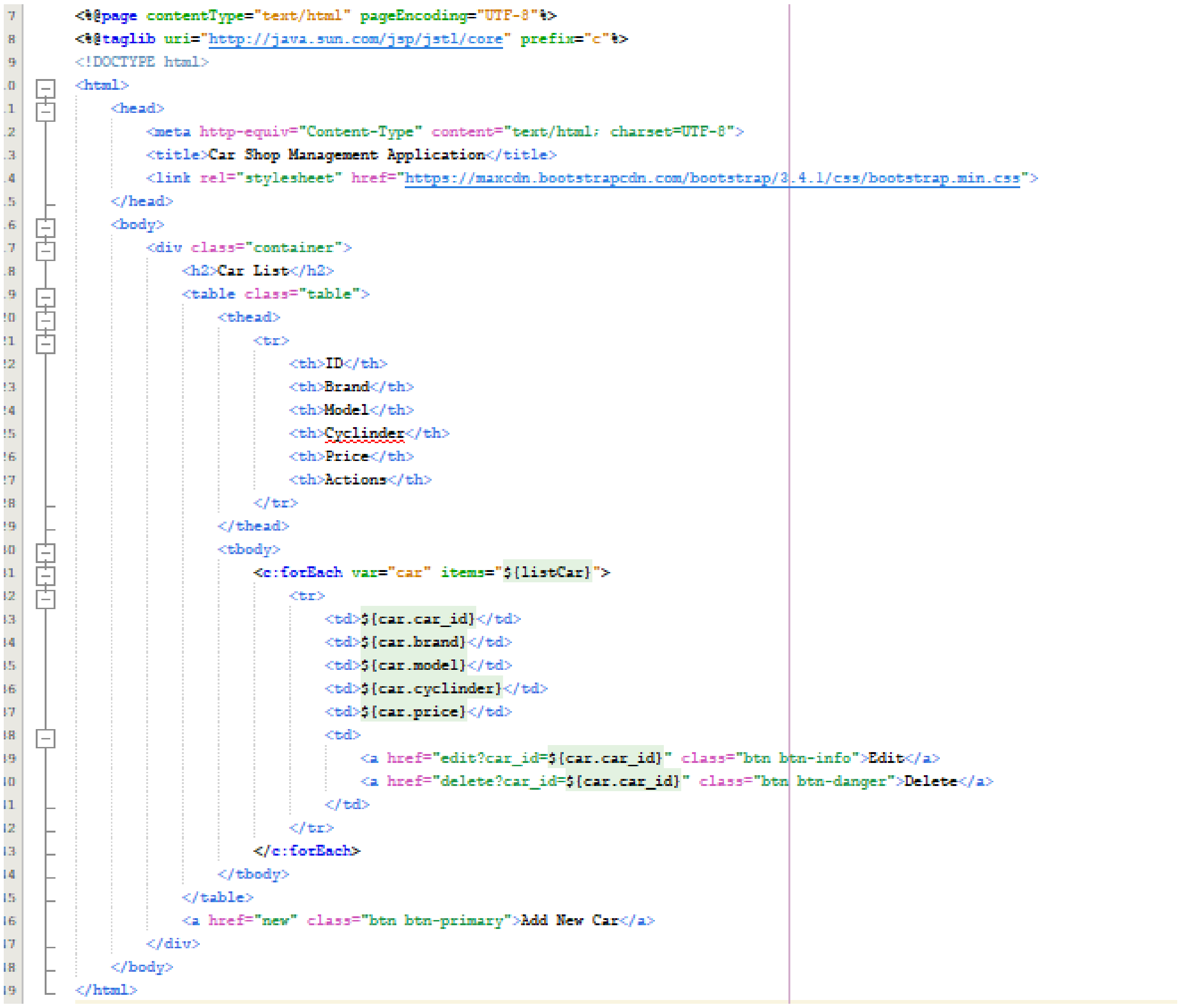


```java
package com.Model;

/**
 *
 * @author User
 */

public class Car {
    private int Car_id;
    private String Brand;
    private String Model;
    private String Cyclinder;
    private String Price;

    // Constructor for insert
    public Car(String Brand, String Model, String Cyclinder, String Price) {
        this.Brand = Brand;
        this.Model = Model;
        this.Cyclinder = Cyclinder;
        this.Price = Price;
    }

    // Constructor for update and select by ID
    public Car(int Car_id, String Brand, String Model, String Cyclinder, String Price) {
        this.Car_id = Car_id;
        this.Brand = Brand;
        this.Model = Model;
        this.Cyclinder = Cyclinder;
        this.Price = Price;
    }
```



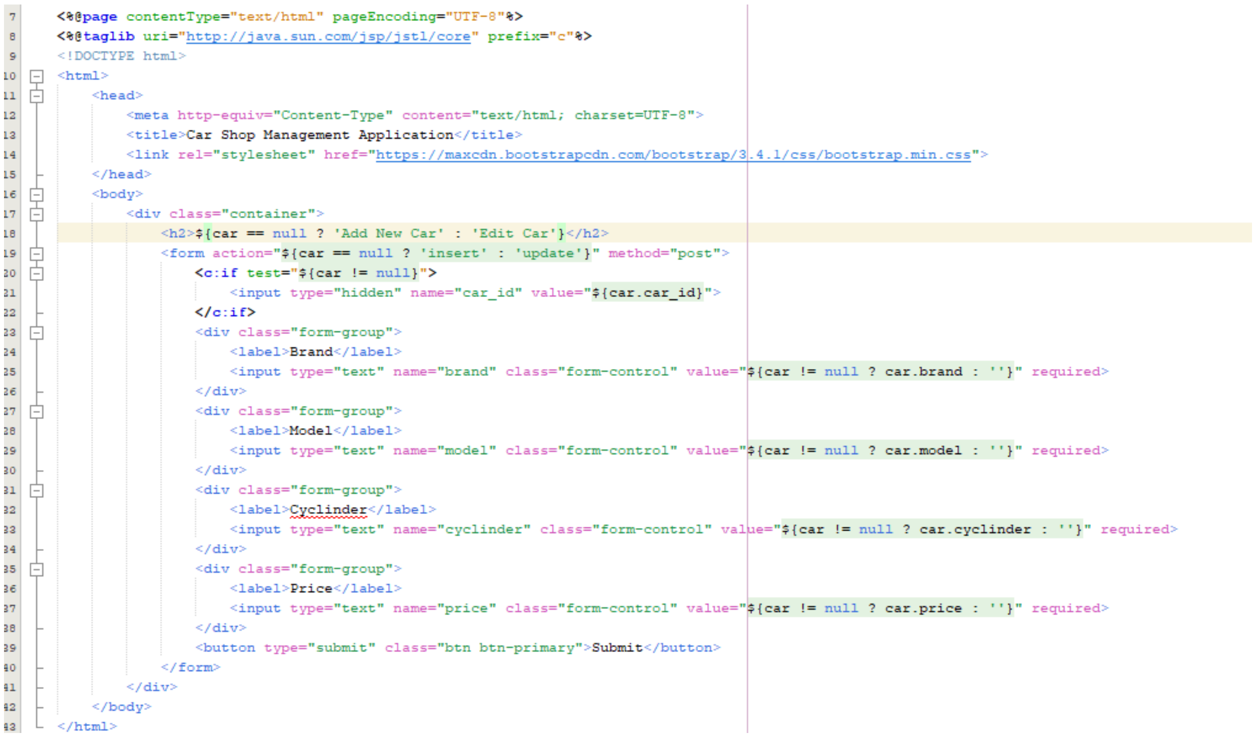
```java
    // Getters and Setters
    public int getCar_id() {
        return Car_id;
    }

    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 String getCyclinder() {
        return Cyclinder;
    }

    public void setCyclinder(String Cyclinder) {
        this.Cyclinder = Cyclinder;
    }

    public String getPrice() {
        return Price;
    }

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

carList.jsp



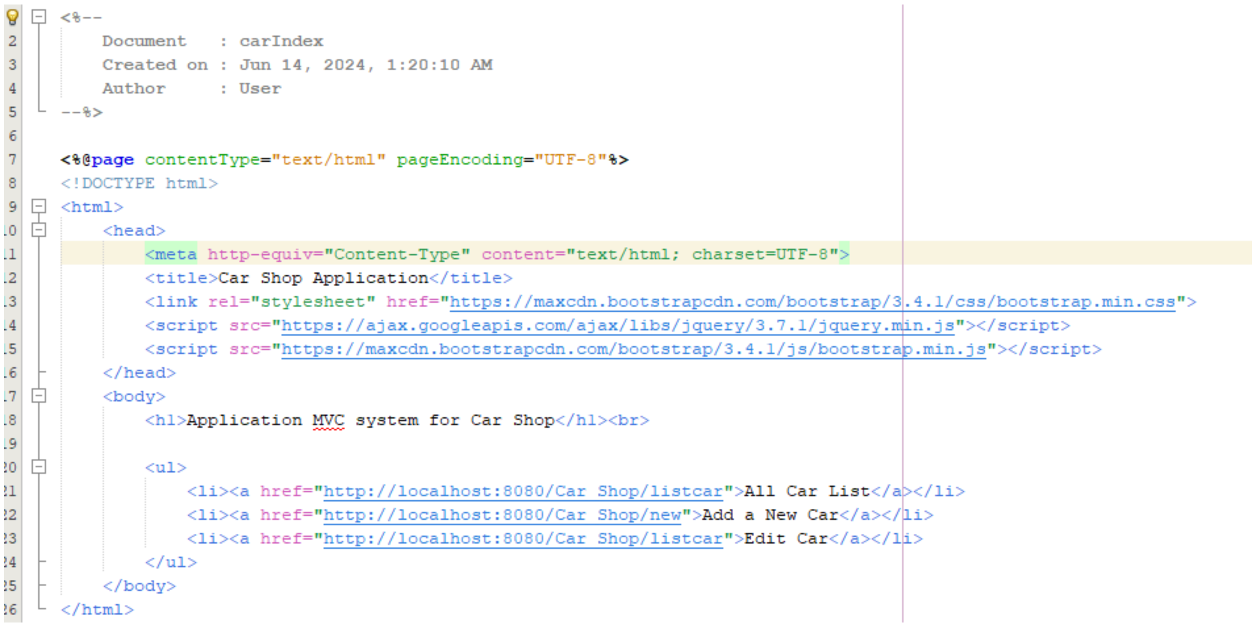
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Car Shop Management Application</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    </head>
    <body>
        <div class="container">
            <h2>Car List</h2>
            <table class="table">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Brand</th>
                        <th>Model</th>
                        <th>Cyclinder</th>
                        <th>Price</th>
                        <th>Actions</th>
                    </tr>
                </thead>
                <tbody>
                    <c:forEach var="car" items="${listCar}">
                        <tr>
                            <td>${car.car_id}</td>
                            <td>${car.brand}</td>
                            <td>${car.model}</td>
                            <td>${car.cyclinder}</td>
                            <td>${car.price}</td>
                            <td>
                                <a href="edit?car_id=${car.car_id}" class="btn btn-info">Edit</a>
                                <a href="delete?car_id=${car.car_id}" class="btn btn-danger">Delete</a>
                            </td>
                        </tr>
                    </c:forEach>
                </tbody>
            </table>
            <a href="new" class="btn btn-primary">Add New Car</a>
        </div>
    </body>
</html>
```

carForm.jsp



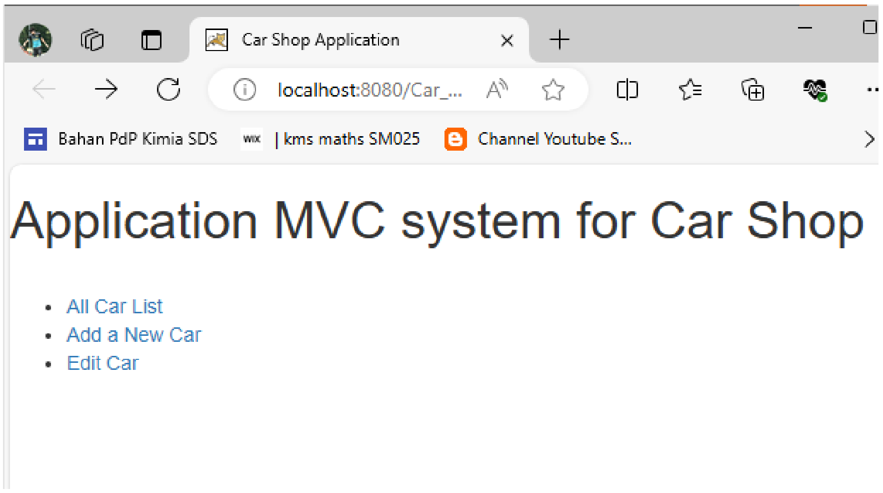
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Car Shop Management Application</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    </head>
    <body>
        <div class="container">
            <h2>${car == null ? 'Add New Car' : 'Edit Car'}</h2>
            <form action="${car == null ? 'insert' : 'update'}" method="post">
                <c:if test="${car != null}">
                    <input type="hidden" name="car_id" value="${car.car_id}">
                </c:if>
                <div class="form-group">
                    <label>Brand</label>
                    <input type="text" name="brand" class="form-control" value="${car != null ? car.brand : ''}" required>
                </div>
                <div class="form-group">
                    <label>Model</label>
                    <input type="text" name="model" class="form-control" value="${car != null ? car.model : ''}" required>
                </div>
                <div class="form-group">
                    <label>Cyclinder</label>
                    <input type="text" name="cyclinder" class="form-control" value="${car != null ? car.cyclinder : ''}" required>
                </div>
                <div class="form-group">
                    <label>Price</label>
                    <input type="text" name="price" class="form-control" value="${car != null ? car.price : ''}" required>
                </div>
                <button type="submit" class="btn btn-primary">Submit</button>
            </form>
        </div>
    </body>
</html>
```

carIndex.jsp



```
<%--
    Document   : carIndex
    Created on : Jun 14, 2024, 1:20:10 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>Car Shop Application</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
    </head>
    <body>
        <h1>Application MVC system for Car Shop</h1><br>

        <ul>
            <li><a href="http://localhost:8080/Car_Shop/listcar">All Car List</a></li>
            <li><a href="http://localhost:8080/Car_Shop/new">Add a New Car</a></li>
            <li><a href="http://localhost:8080/Car_Shop/listcar">Edit Car</a></li>
        </ul>
    </body>
</html>
```

The output.



Car Shop Application

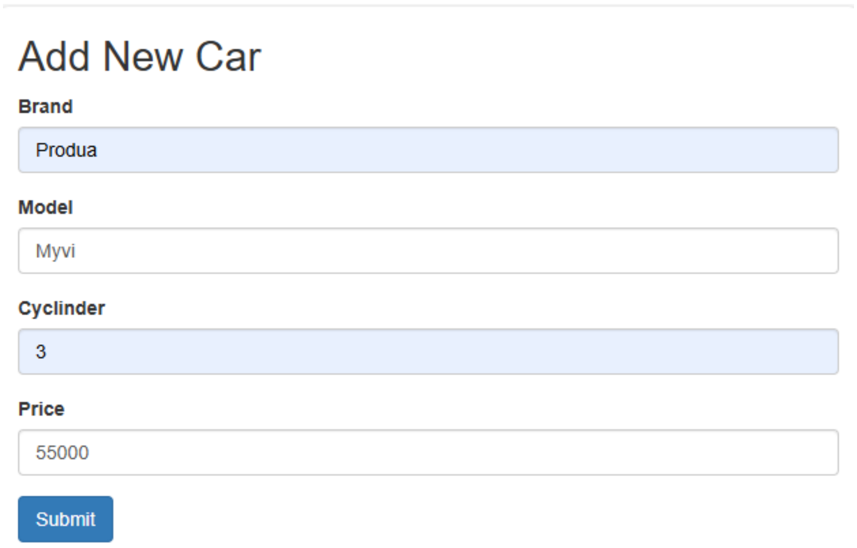
localhost:8080/Car_...

Bahan PdP Kimia SDS | kms maths SM025 Channel Youtube S...

Application MVC system for Car Shop

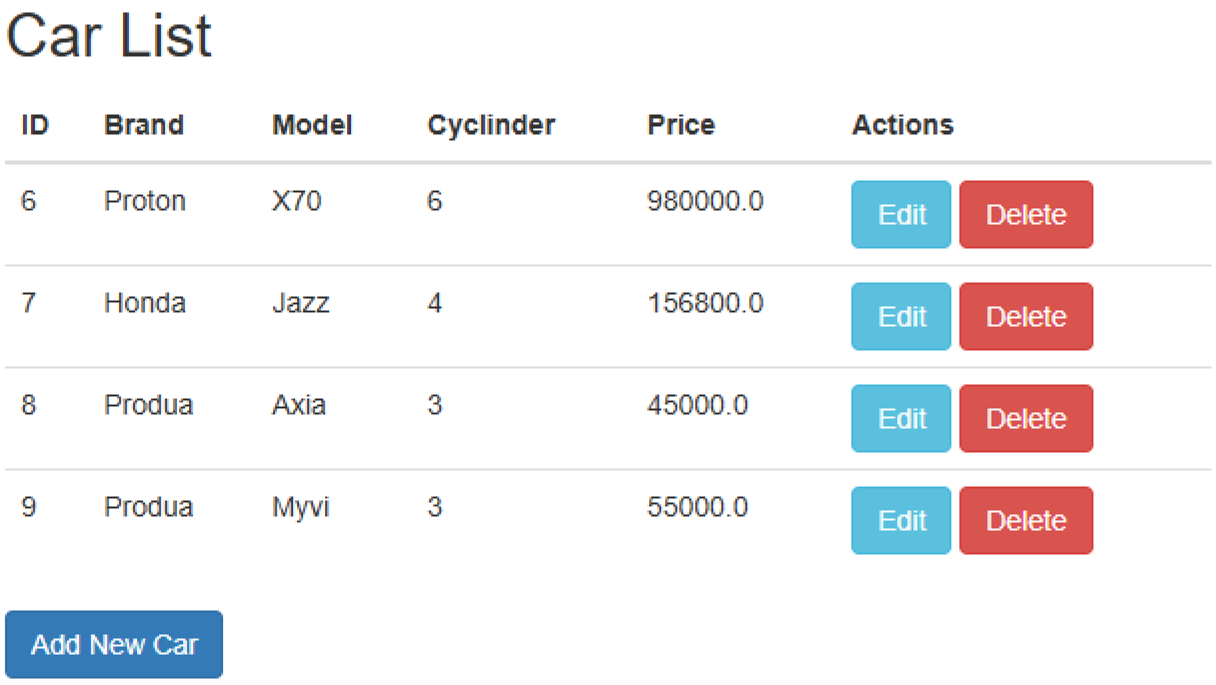
- All Car List
- Add a New Car
- Edit Car

- Add new car



Add New Car

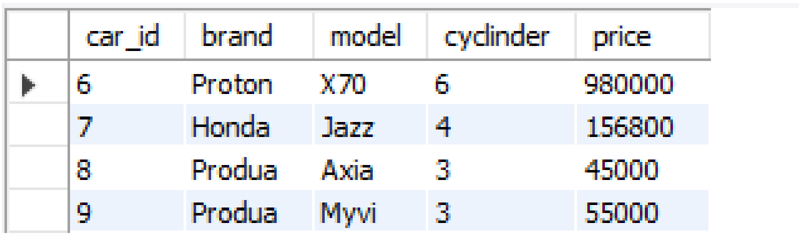
Brand	Produa
Model	Myvi
Cyclinder	3
Price	55000



Car List

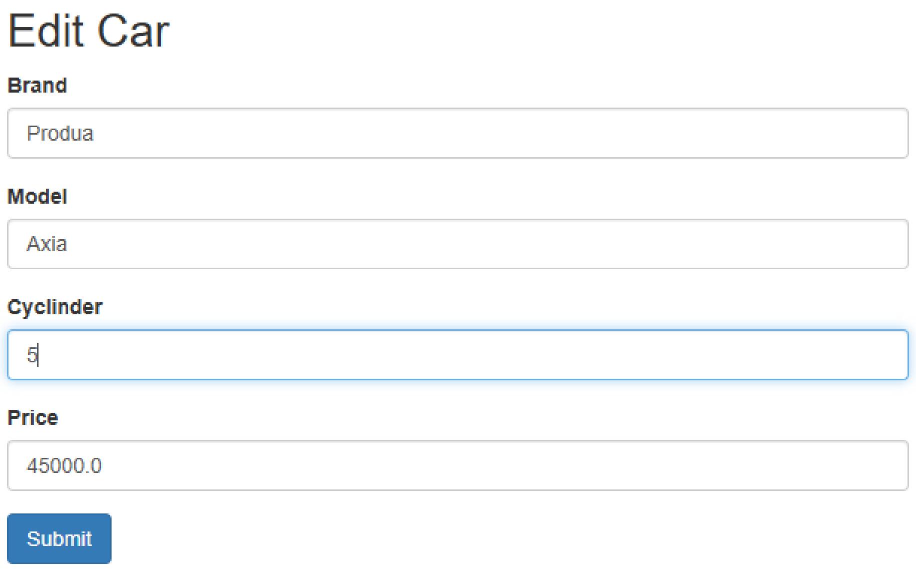
ID	Brand	Model	Cyclinder	Price	Actions
6	Proton	X70	6	980000.0	Edit Delete
7	Honda	Jazz	4	156800.0	Edit Delete
8	Produa	Axia	3	45000.0	Edit Delete
9	Produa	Myvi	3	55000.0	Edit Delete

Add New Car



car_id	brand	model	cyclinder	price
6	Proton	X70	6	980000
7	Honda	Jazz	4	156800
8	Produa	Axia	3	45000
9	Produa	Myvi	3	55000

- Edit car



Edit Car

Brand

Produa

Model

Axia

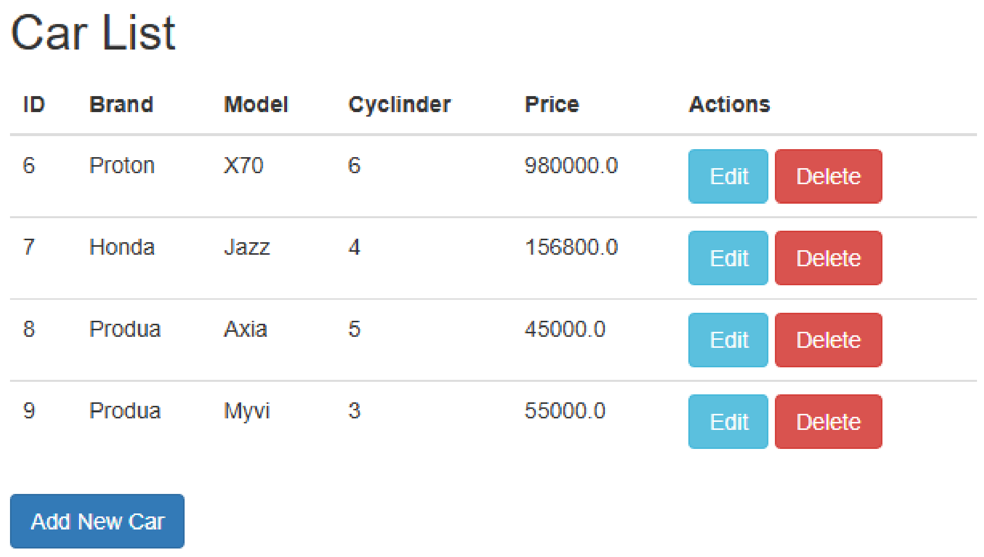
Cyclinder

5

Price

45000.0

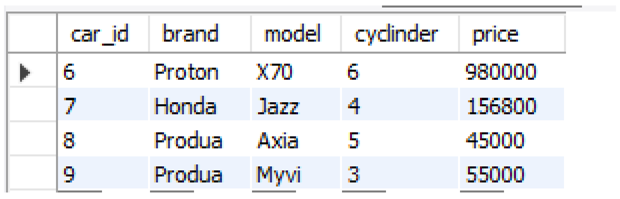
Submit



Car List

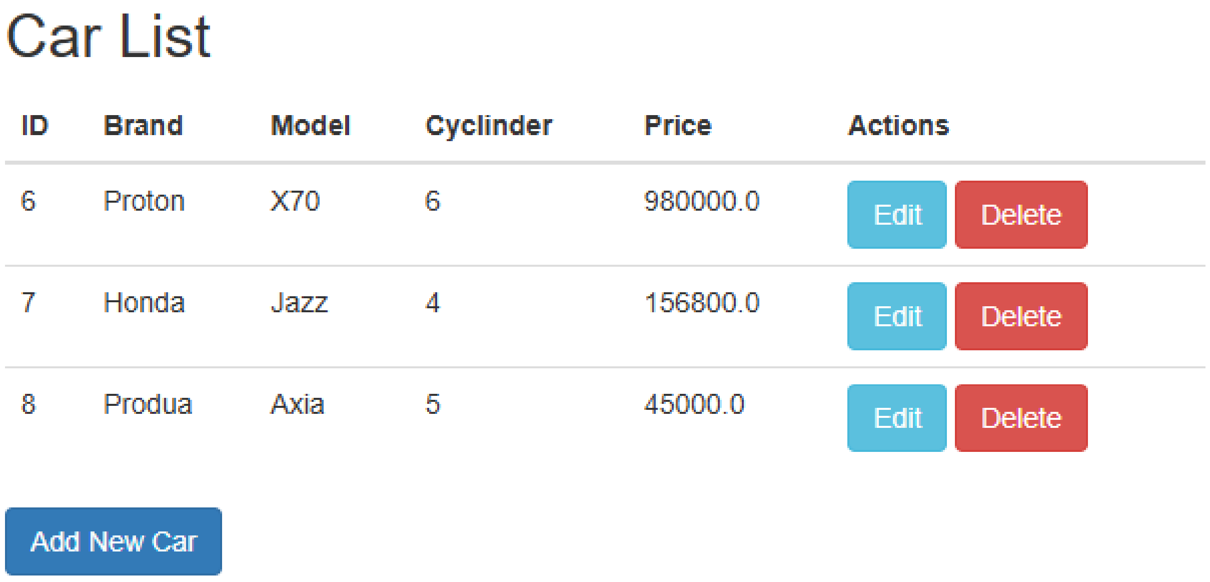
ID	Brand	Model	Cyclinder	Price	Actions
6	Proton	X70	6	980000.0	Edit Delete
7	Honda	Jazz	4	156800.0	Edit Delete
8	Produa	Axia	5	45000.0	Edit Delete
9	Produa	Myvi	3	55000.0	Edit Delete

Add New Car



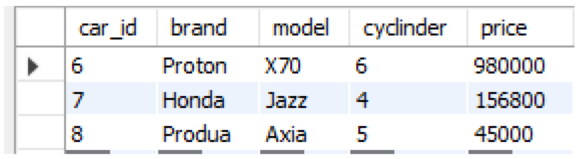
car_id	brand	model	cyclinder	price
6	Proton	X70	6	980000
7	Honda	Jazz	4	156800
8	Produa	Axia	5	45000
9	Produa	Myvi	3	55000

- Delete car



Car List

ID	Brand	Model	Cyclinder	Price	Actions
6	Proton	X70	6	980000.0	Edit Delete
7	Honda	Jazz	4	156800.0	Edit Delete
8	Produa	Axia	5	45000.0	Edit Delete



car_id	brand	model	cyclinder	price
6	Proton	X70	6	980000
7	Honda	Jazz	4	156800
8	Produa	Axia	5	45000