UNIVERSITI MALAYSIA TERENGGANU

CSM3023 – WEB BASED APPLICATION DEVELOPMENT

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 8

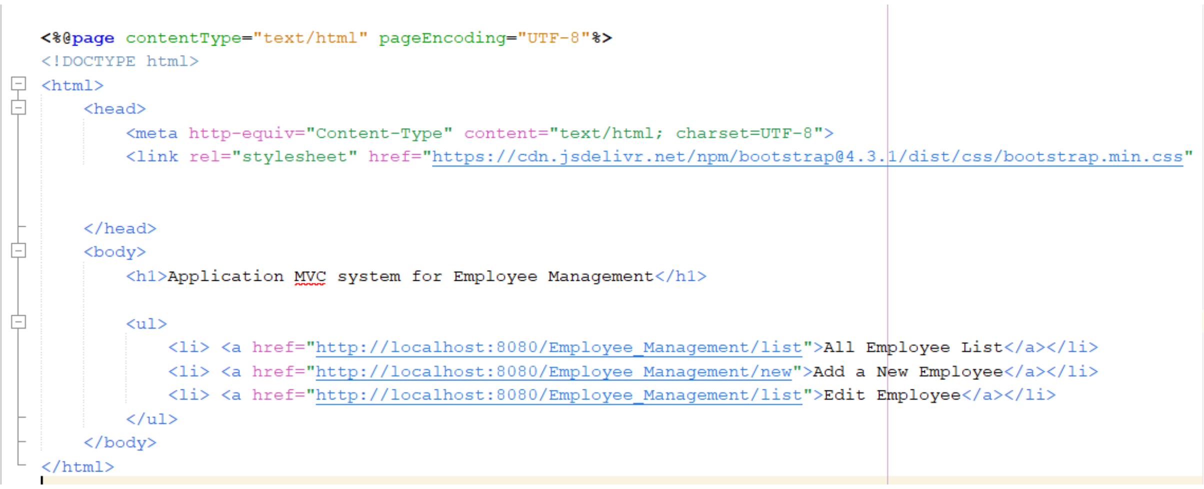
SEMESTER II 2023/2024

Prepared for:

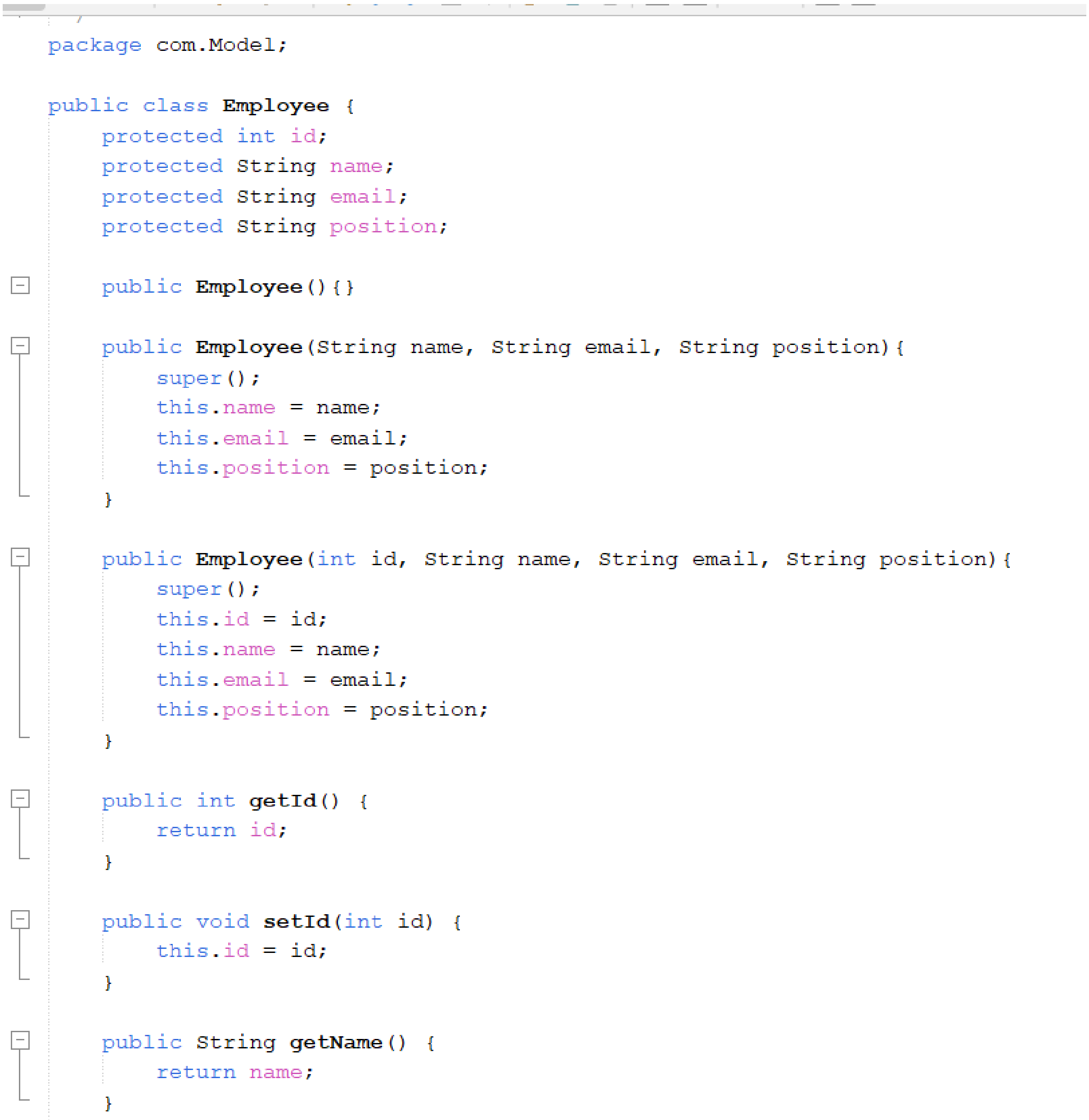
DR. MOHAMAD NOR BIN HASSAN

Prepared by:

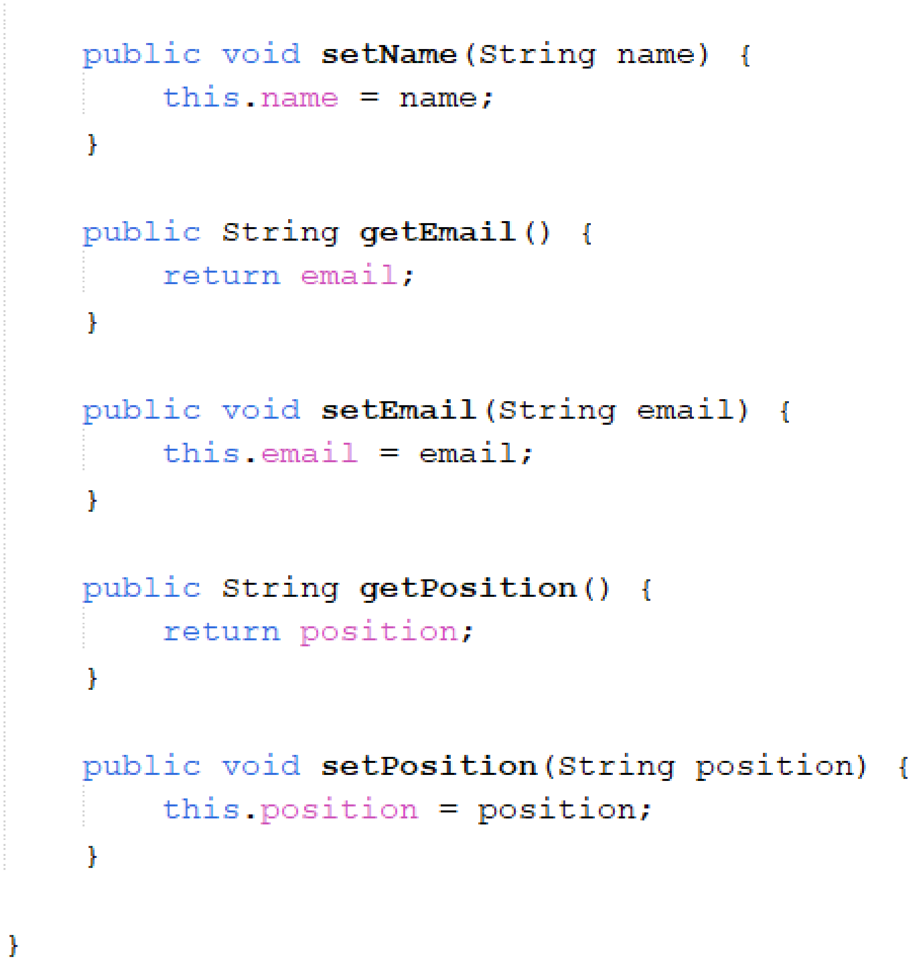
MARLIANTI BT MUF PIARLIS (S66353)



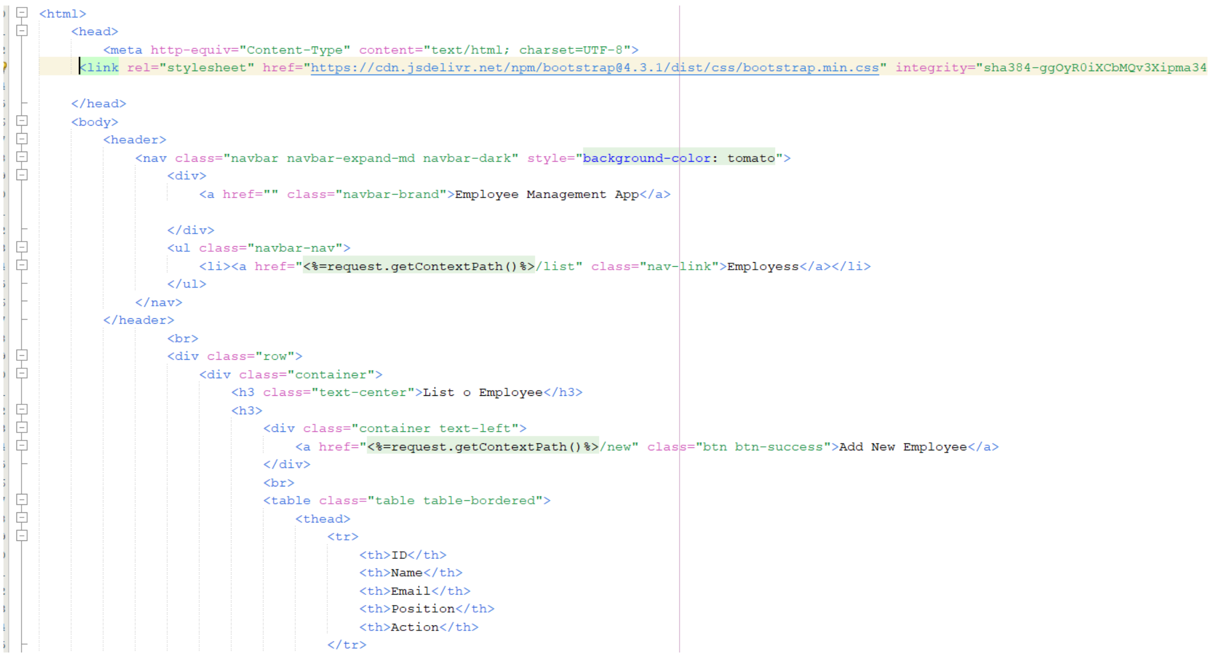
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"


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

        <ul>
            <li> <a href="http://localhost:8080/Employee_Management/list">All Employee List</a></li>
            <li> <a href="http://localhost:8080/Employee_Management/new">Add a New Employee</a></li>
            <li> <a href="http://localhost:8080/Employee_Management/list">Edit Employee</a></li>
        </ul>
    </body>
</html>
```



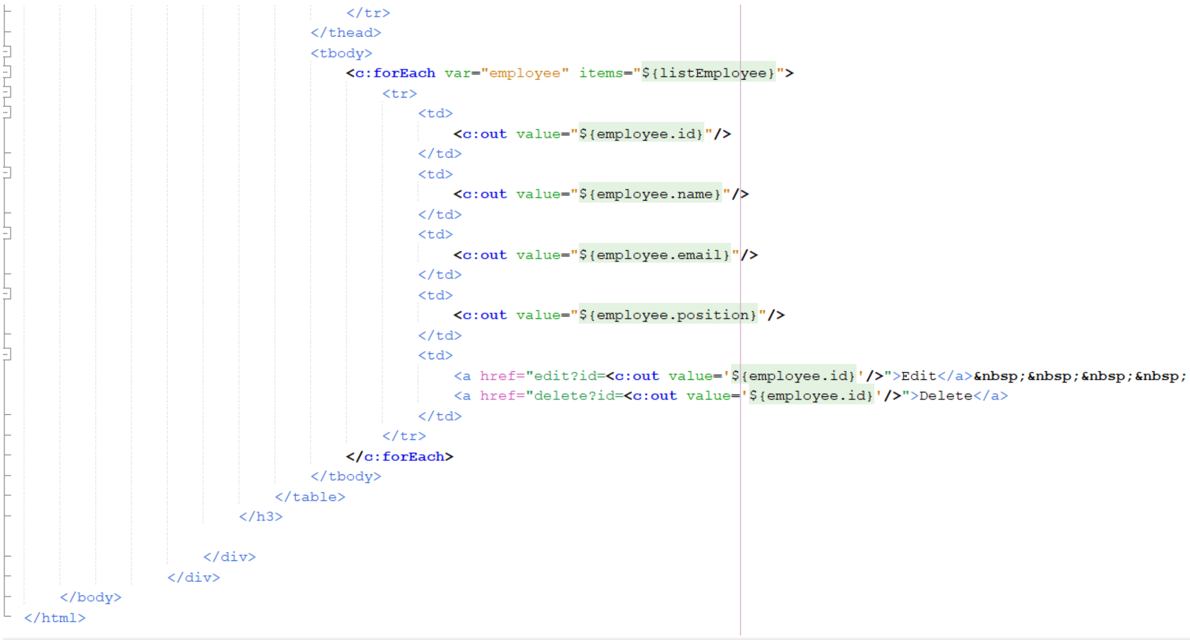
```
package com.Model;

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 int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }
```



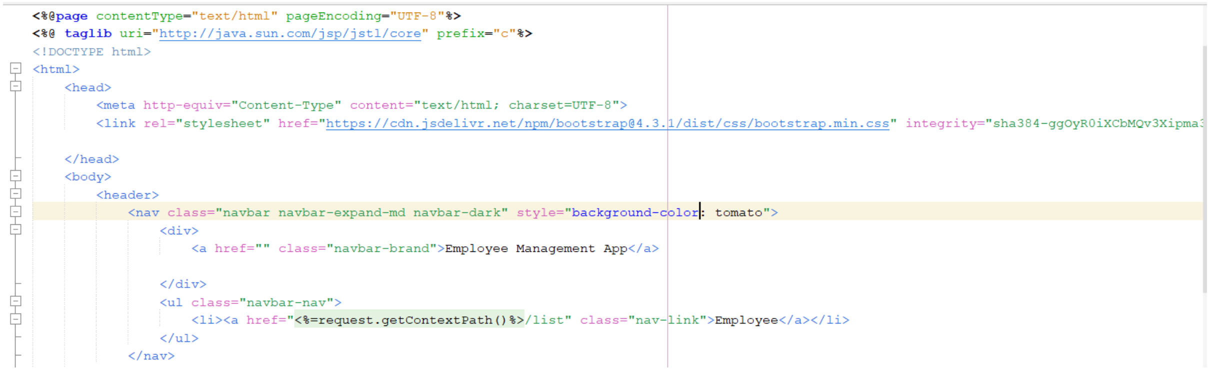
```java
    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

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

    public String getPosition() {
        return position;
    }

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

}
```



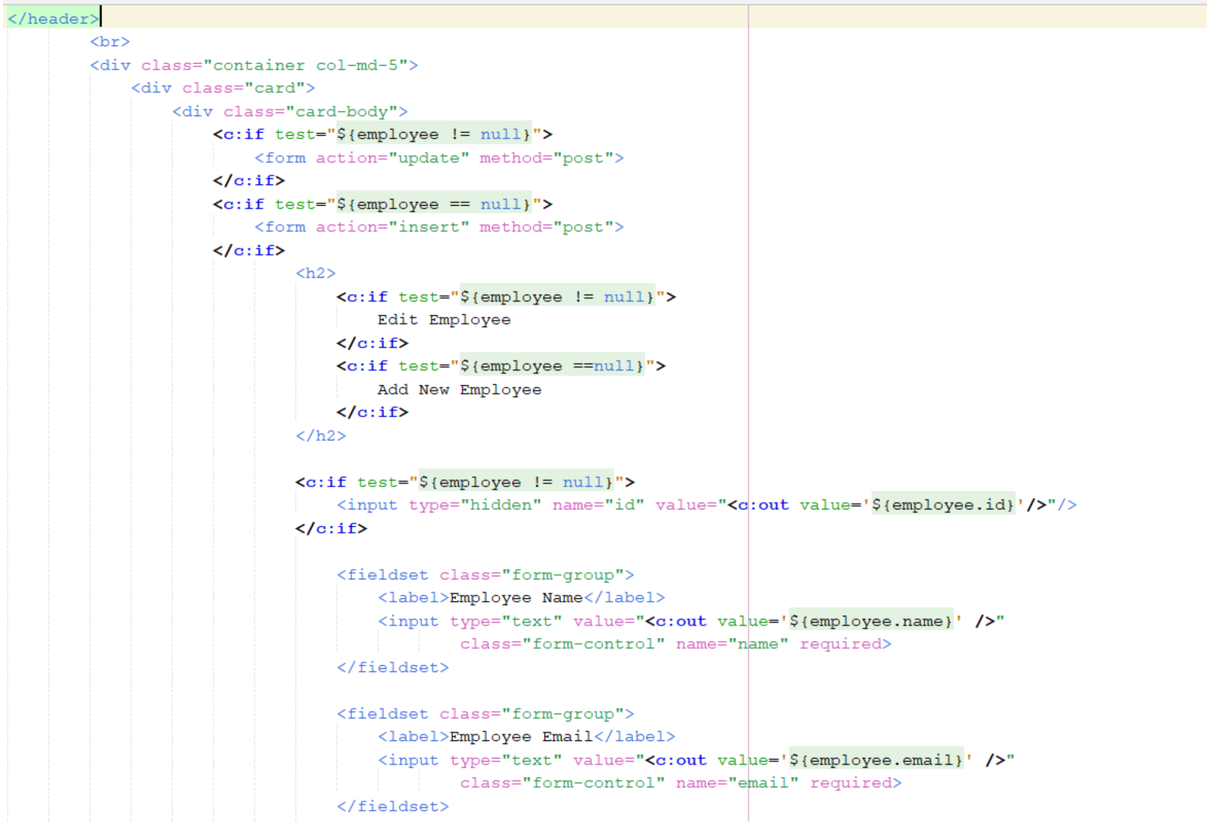
```html
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34

    </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">Employess</a></li>
                </ul>
            </nav>
        </header>
                <br>
                <div class="row">
                    <div class="container">
                        <h3 class="text-center">List o Employee</h3>
                        <h3>
                            <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>Action</th>
                                    </tr>
```



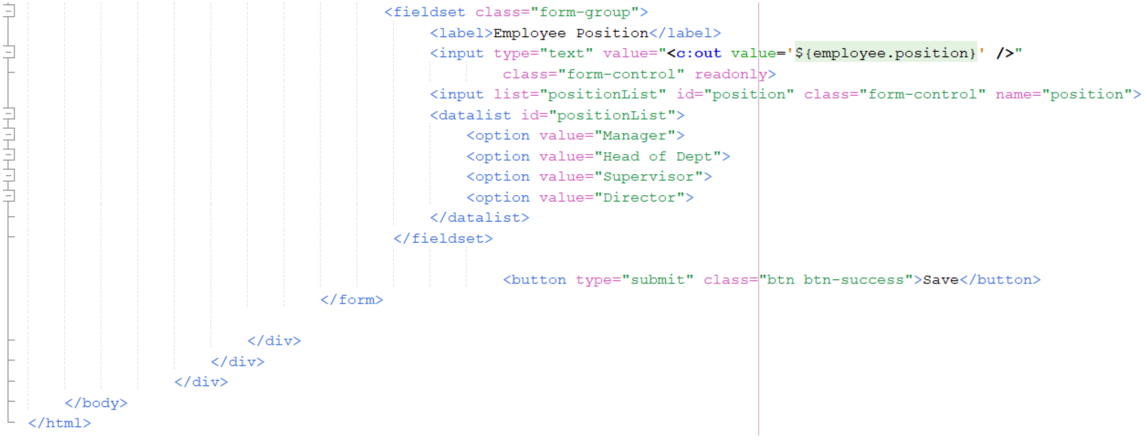
```
                                    </tr>
                                </thead>
                                <tbody>
                                    <c:forEach var="employee" items="${listEmployee}">
                                        <tr>
                                            <td>
                                                <c:out value="${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>
                        </h3>

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



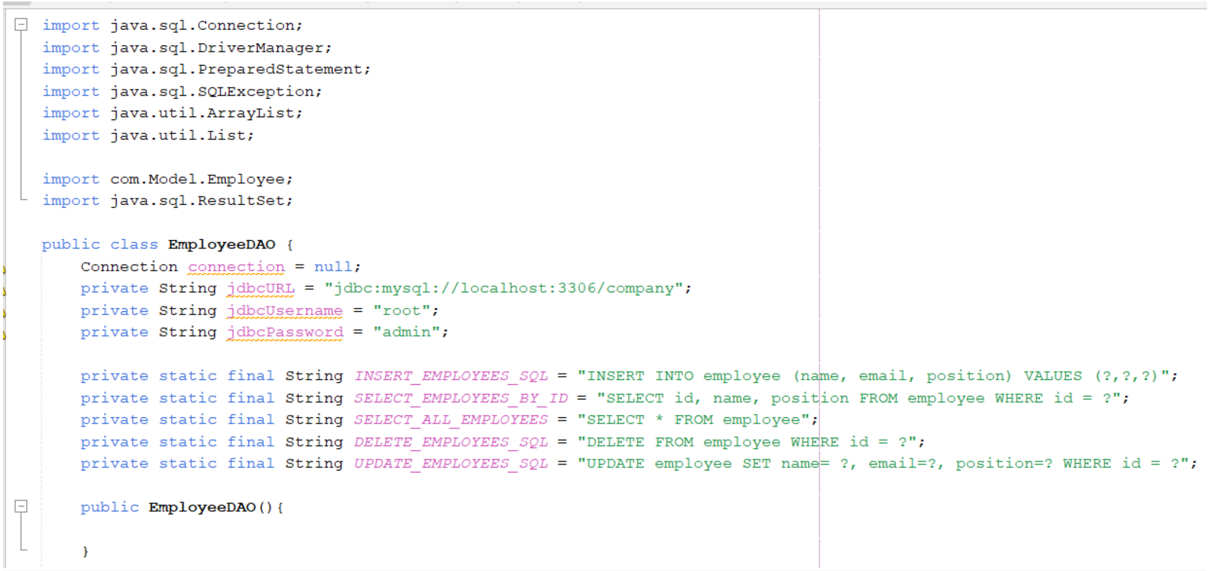
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma3

    </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">Employee</a></li>
                </ul>
            </nav>
```



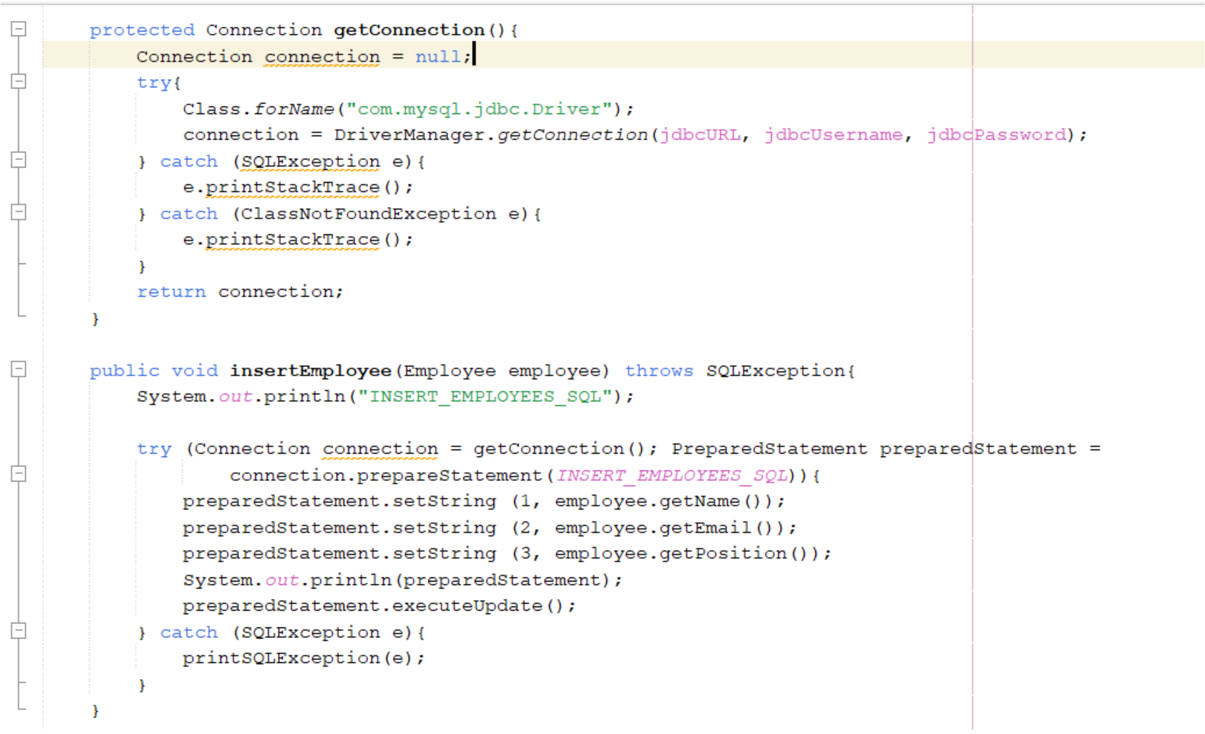
```
</header>
        <br>
        <div class="container col-md-5">
            <div class="card">
                <div class="card-body">
                    <c:if test="${employee != null}">
                        <form action="update" method="post">
                    </c:if>
                    <c:if test="${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>Employee Name</label>
                                    <input type="text" value="<c:out value='${employee.name}' />"
                                            class="form-control" name="name" required>
                                </fieldset>

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



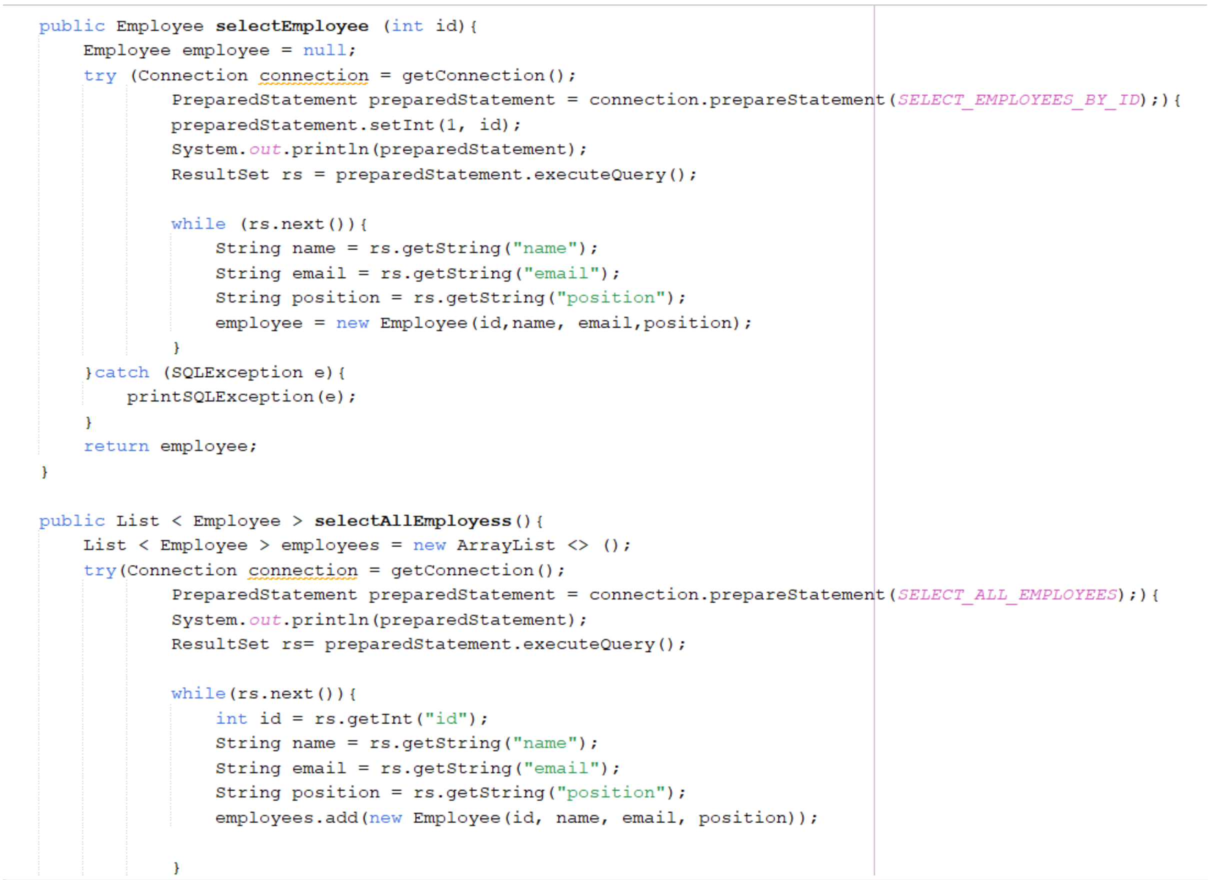
```
                                        <fieldset class="form-group">
                                            <label>Employee Position</label>
                                            <input type="text" value="<c:out value='${employee.position}' />"
                                                    class="form-control" readonly>
                                            <input list="positionList" id="position" class="form-control" name="position">
                                            <datalist id="positionList">
                                                <option value="Manager">
                                                <option 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>
```



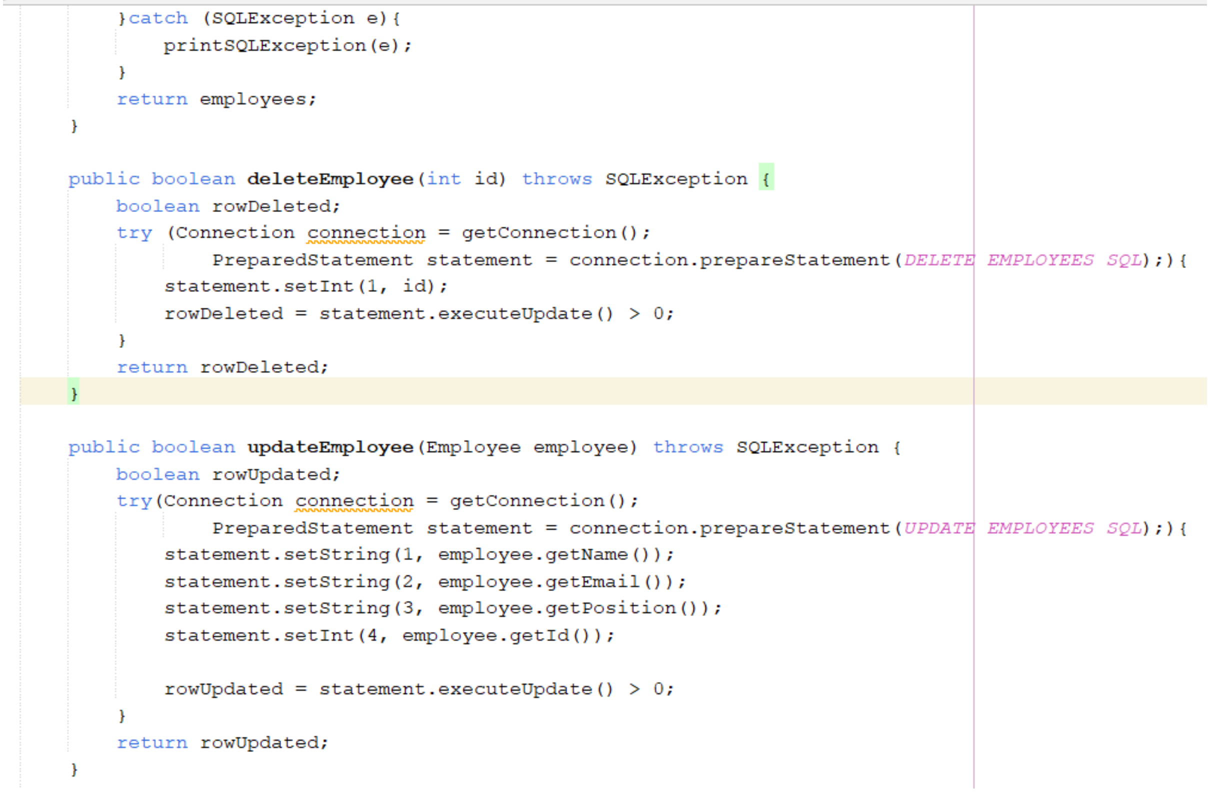
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.Model.Employee;
import java.sql.ResultSet;

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

    private static final String INSERT_EMPLOYEES_SQL = "INSERT INTO employee (name, email, position) VALUES (?,?,?)";
    private static final String SELECT_EMPLOYEES_BY_ID = "SELECT id, name, position FROM employee WHERE id = ?";
    private static final String SELECT_ALL_EMPLOYEES = "SELECT * FROM employee";
    private static final String DELETE_EMPLOYEES_SQL = "DELETE FROM employee WHERE id = ?";
    private static final String UPDATE_EMPLOYEES_SQL = "UPDATE employee SET name= ?, email=?, position=? WHERE id = ?";

    public EmployeeDAO(){

    }
```



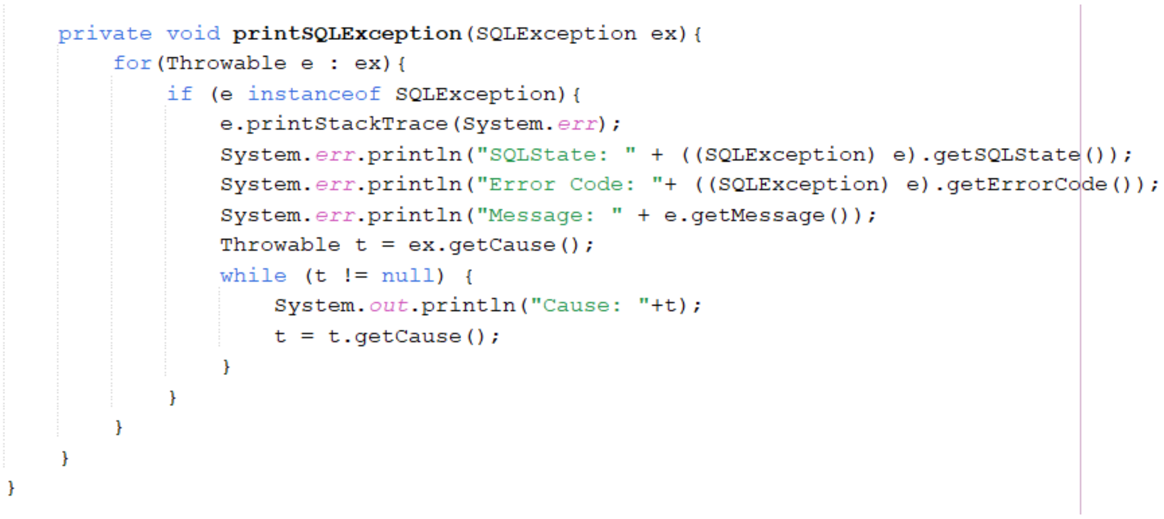
```
    protected Connection getConnection(){
        Connection connection = null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);
        } catch (SQLException e){
            e.printStackTrace();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

    public void insertEmployee(Employee employee) throws SQLException{
        System.out.println("INSERT_EMPLOYEES_SQL");

        try (Connection connection = getConnection(); PreparedStatement preparedStatement =
                connection.prepareStatement(INSERT_EMPLOYEES_SQL)){
            preparedStatement.setString (1, employee.getName());
            preparedStatement.setString (2, employee.getEmail());
            preparedStatement.setString (3, employee.getPosition());
            System.out.println(preparedStatement);
            preparedStatement.executeUpdate();
        } catch (SQLException e){
            printSQLException(e);
        }
    }
```



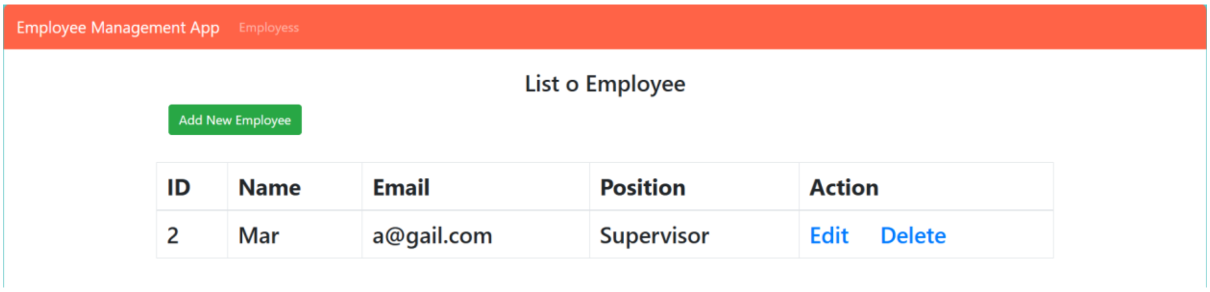
```
public Employee selectEmployee (int id){
    Employee employee = null;
    try (Connection connection = getConnection();
            PreparedStatement preparedStatement = connection.prepareStatement(SELECT_EMPLOYEES_BY_ID);){
            preparedStatement.setInt(1, id);
            System.out.println(preparedStatement);
            ResultSet rs = preparedStatement.executeQuery();

            while (rs.next()){
                String name = rs.getString("name");
                String email = rs.getString("email");
                String position = rs.getString("position");
                employee = new Employee(id,name, email,position);
            }
    }catch (SQLException e){
        printSQLException(e);
    }
    return employee;
}

public List < Employee > selectAllEmployess(){
    List < Employee > employees = new ArrayList <> ();
    try(Connection connection = getConnection();
            PreparedStatement preparedStatement = connection.prepareStatement(SELECT_ALL_EMPLOYEES);){
            System.out.println(preparedStatement);
            ResultSet rs= preparedStatement.executeQuery();

            while(rs.next()){
                int id = rs.getInt("id");
                String name = rs.getString("name");
                String email = rs.getString("email");
                String position = rs.getString("position");
                employees.add(new Employee(id, name, email, position));

            }
```



```
    }catch (SQLException e){
        printSQLException(e);
    }
    return employees;
}

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

public boolean updateEmployee(Employee employee) throws SQLException {
    boolean rowUpdated;
    try(Connection connection = getConnection();
            PreparedStatement statement = connection.prepareStatement(UPDATE EMPLOYEES SQL);){
        statement.setString(1, employee.getName());
        statement.setString(2, employee.getEmail());
        statement.setString(3, employee.getPosition());
        statement.setInt(4, employee.getId());

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

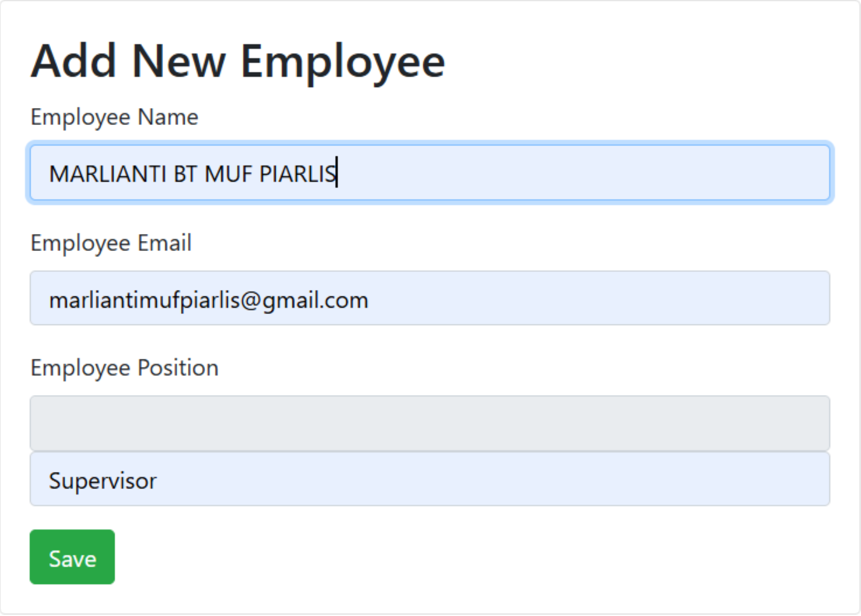


```
    private void printSQLException(SQLException ex){
        for(Throwable e : ex){
            if (e instanceof SQLException){
                e.printStackTrace(System.err);
                System.err.println("SQLState: " + ((SQLException) e).getSQLState());
                System.err.println("Error Code: "+ ((SQLException) e).getErrorCode());
                System.err.println("Message: " + e.getMessage());
                Throwable t = ex.getCause();
                while (t != null) {
                    System.out.println("Cause: "+t);
                    t = t.getCause();
                }
            }
        }
    }
}
```



List o Employee

ID	Name	Email	Position	Action
2	Mar	a@gail.com	Supervisor	Edit Delete



Add New Employee

Employee Name

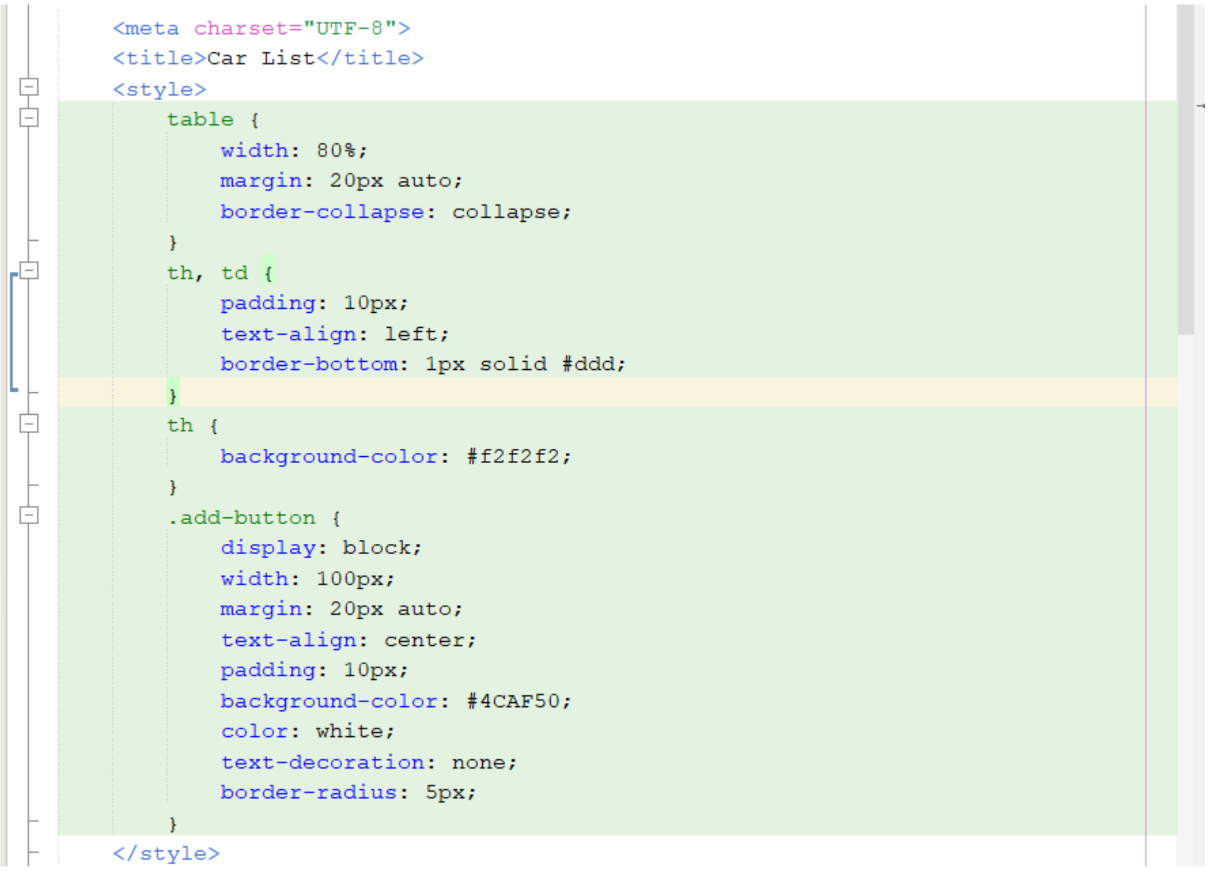
MARLIANTI BT MUF PIARLIS

Employee Email

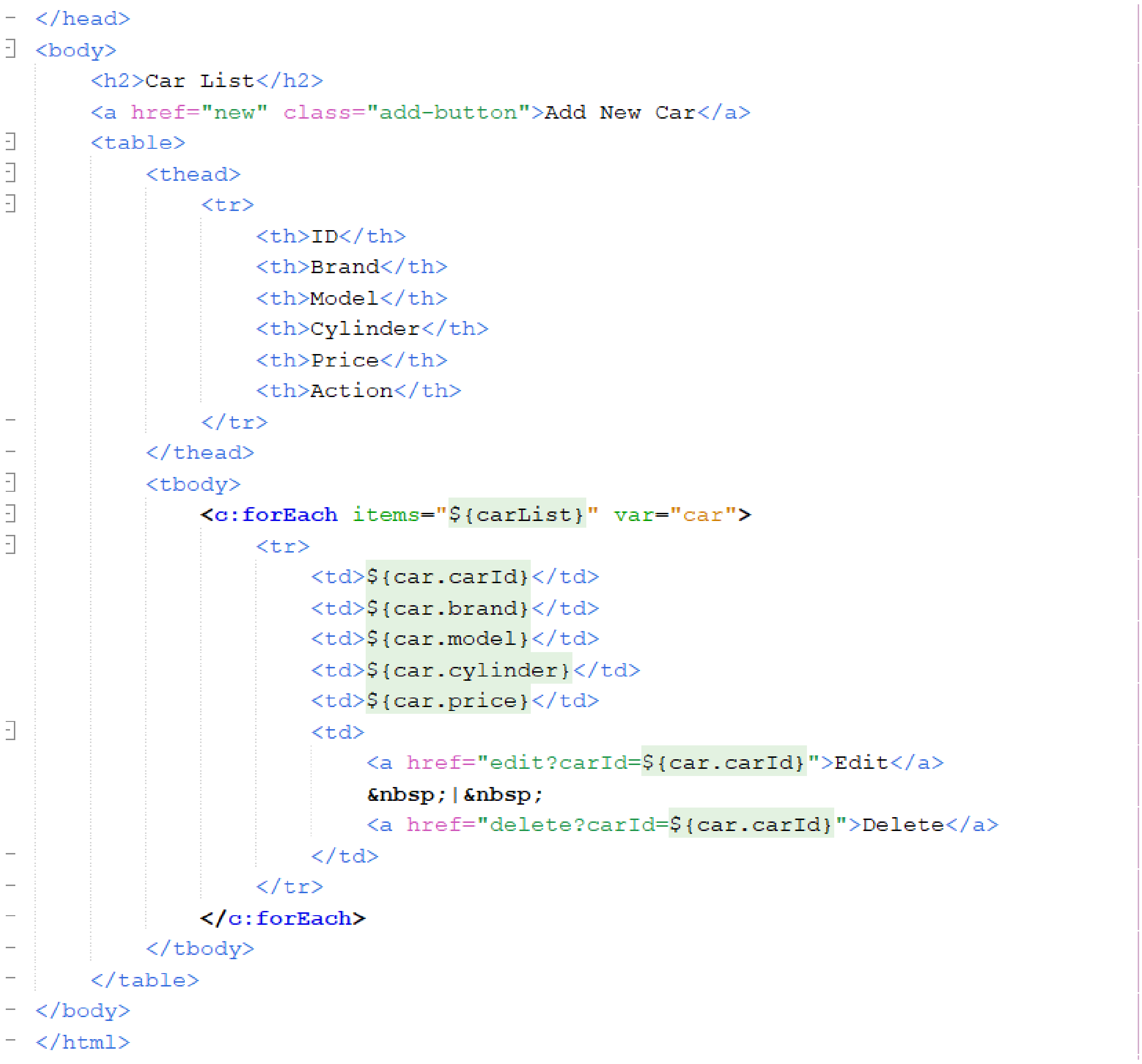
marliantimufpiarlis@gmail.com

Employee Position

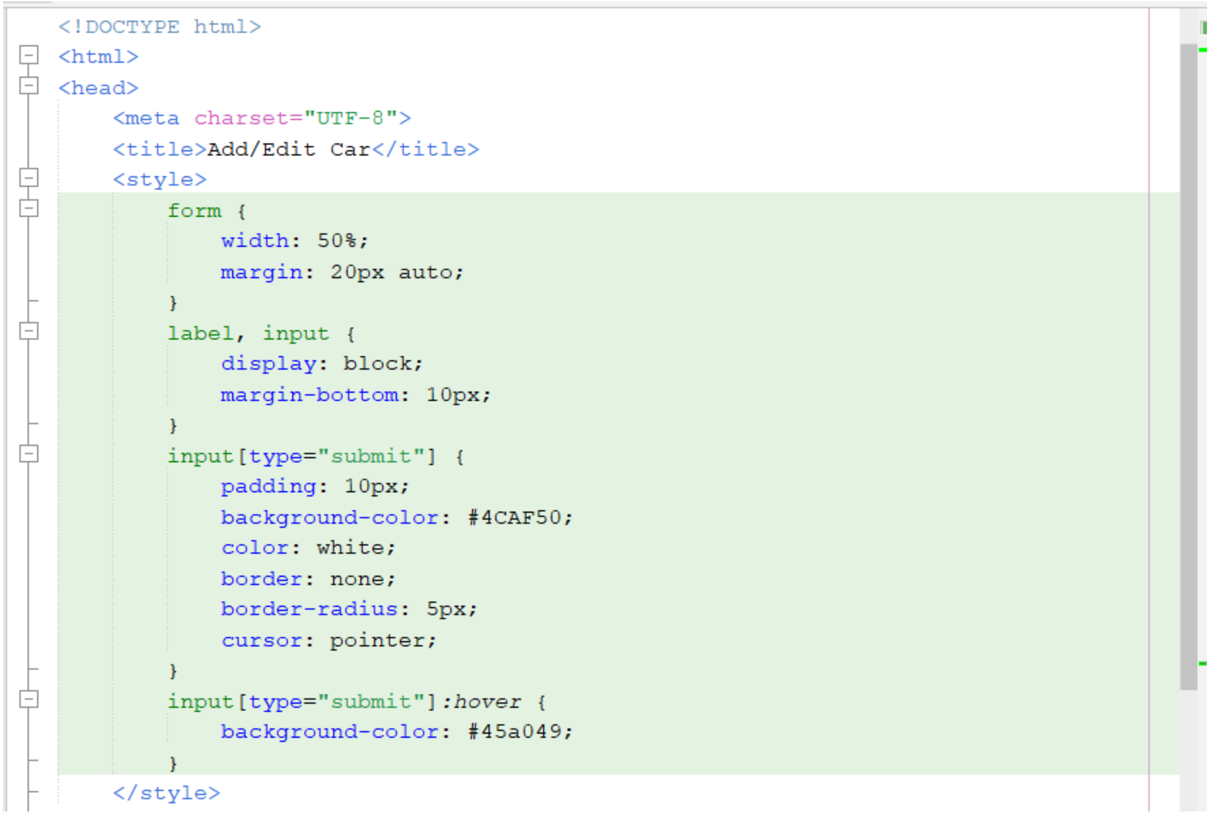
Supervisor



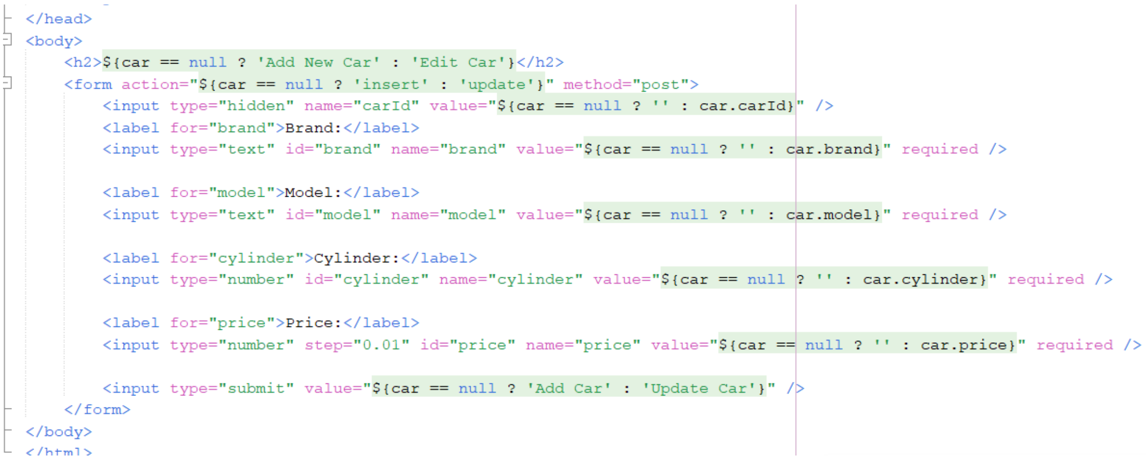
```
    <meta charset="UTF-8">
    <title>Car List</title>
    <style>
        table {
            width: 80%;
            margin: 20px auto;
            border-collapse: collapse;
        }
        th, td {
            padding: 10px;
            text-align: left;
            border-bottom: 1px solid #ddd;
        }
        th {
            background-color: #f2f2f2;
        }
        .add-button {
            display: block;
            width: 100px;
            margin: 20px auto;
            text-align: center;
            padding: 10px;
            background-color: #4CAF50;
            color: white;
            text-decoration: none;
            border-radius: 5px;
        }
    </style>
```



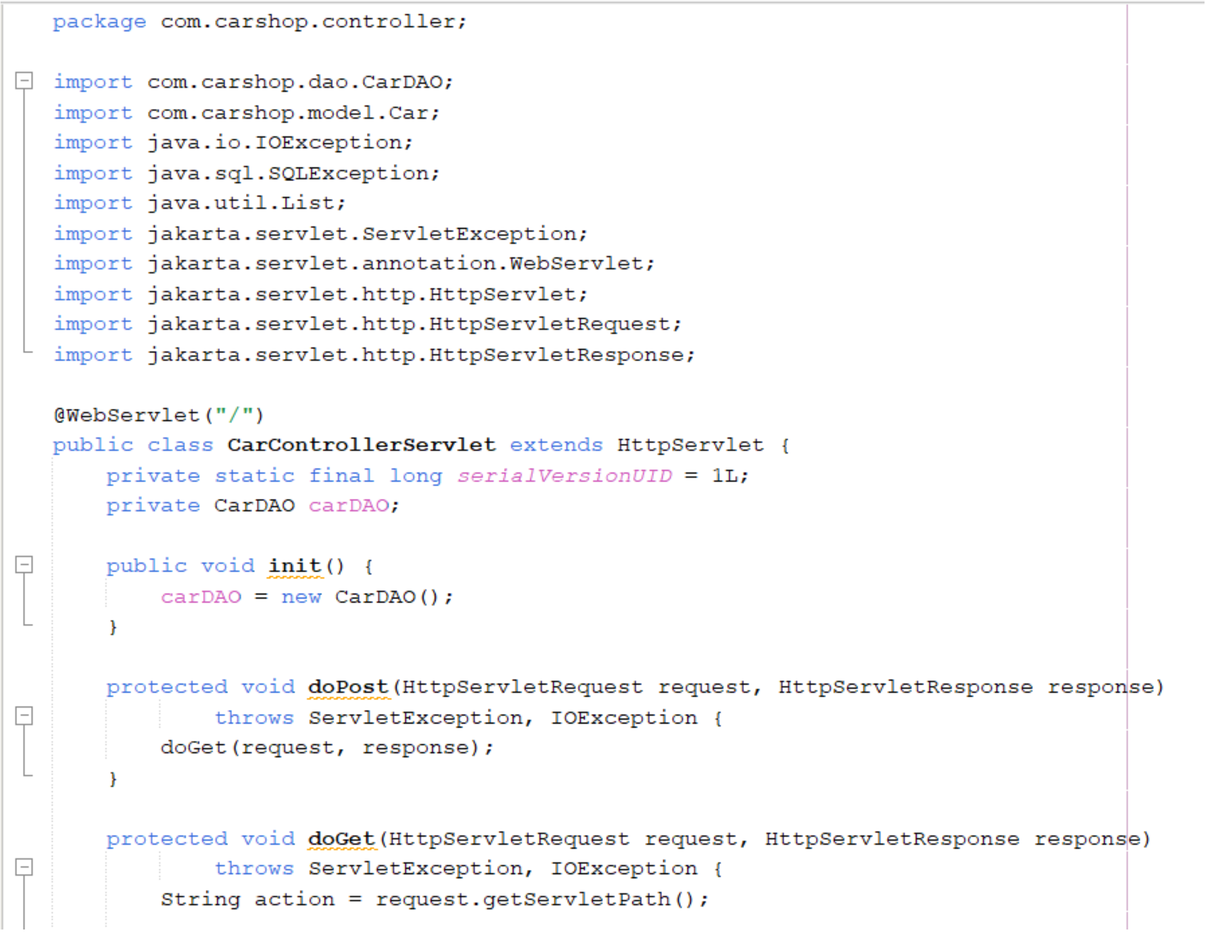
```
</head>
<body>
    <h2>Car List</h2>
    <a href="new" class="add-button">Add New Car</a>
    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Brand</th>
                <th>Model</th>
                <th>Cylinder</th>
                <th>Price</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <c:forEach items="${carList}" var="car">
                <tr>
                    <td>${car.carId}</td>
                    <td>${car.brand}</td>
                    <td>${car.model}</td>
                    <td>${car.cylinder}</td>
                    <td>${car.price}</td>
                    <td>
                        <a href="edit?carId=${car.carId}">Edit</a>
                        &nbsp;|&nbsp;
                        <a href="delete?carId=${car.carId}">Delete</a>
                    </td>
                </tr>
            </c:forEach>
        </tbody>
    </table>
</body>
</html>
```



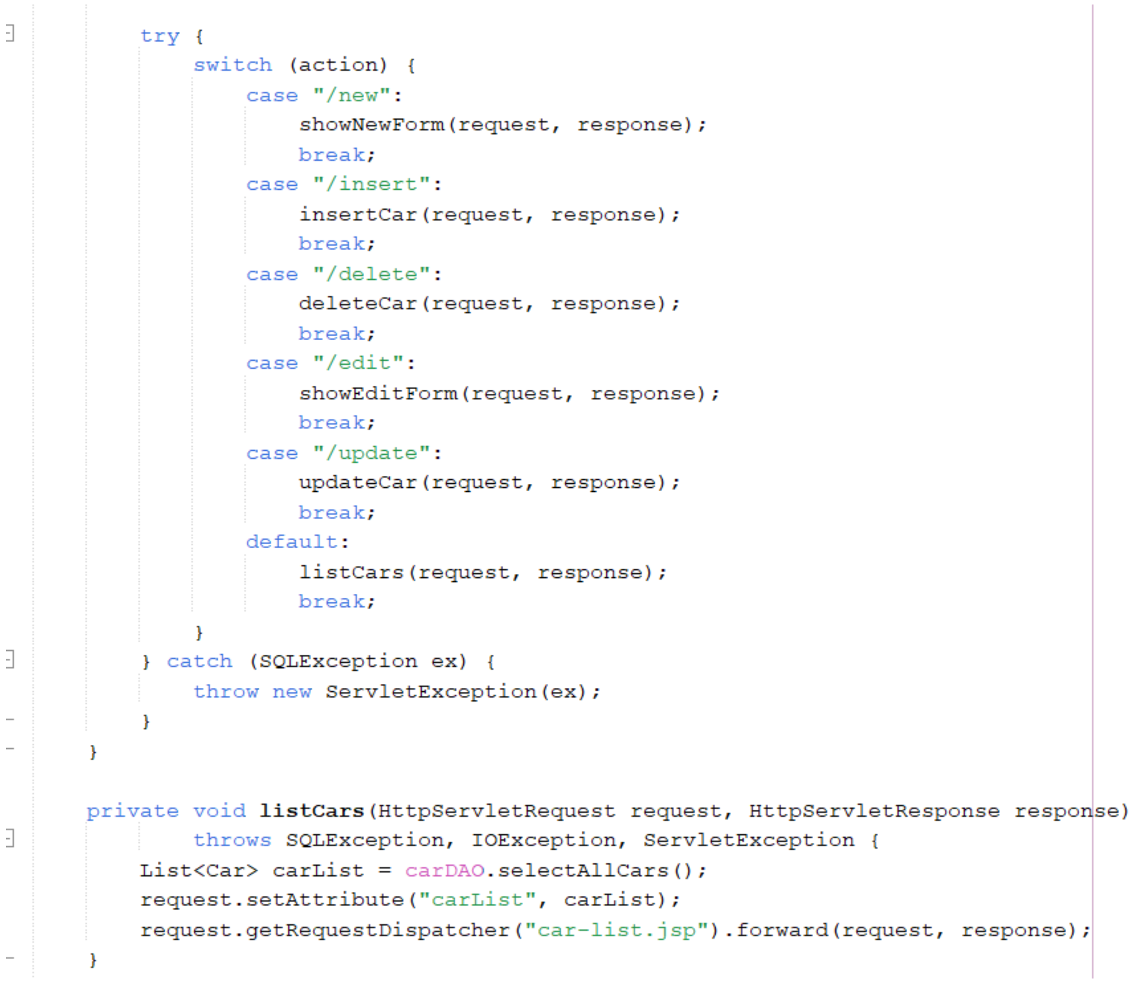
```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Add/Edit Car</title>
    <style>
        form {
            width: 50%;
            margin: 20px auto;
        }
        label, input {
            display: block;
            margin-bottom: 10px;
        }
        input[type="submit"] {
            padding: 10px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
```



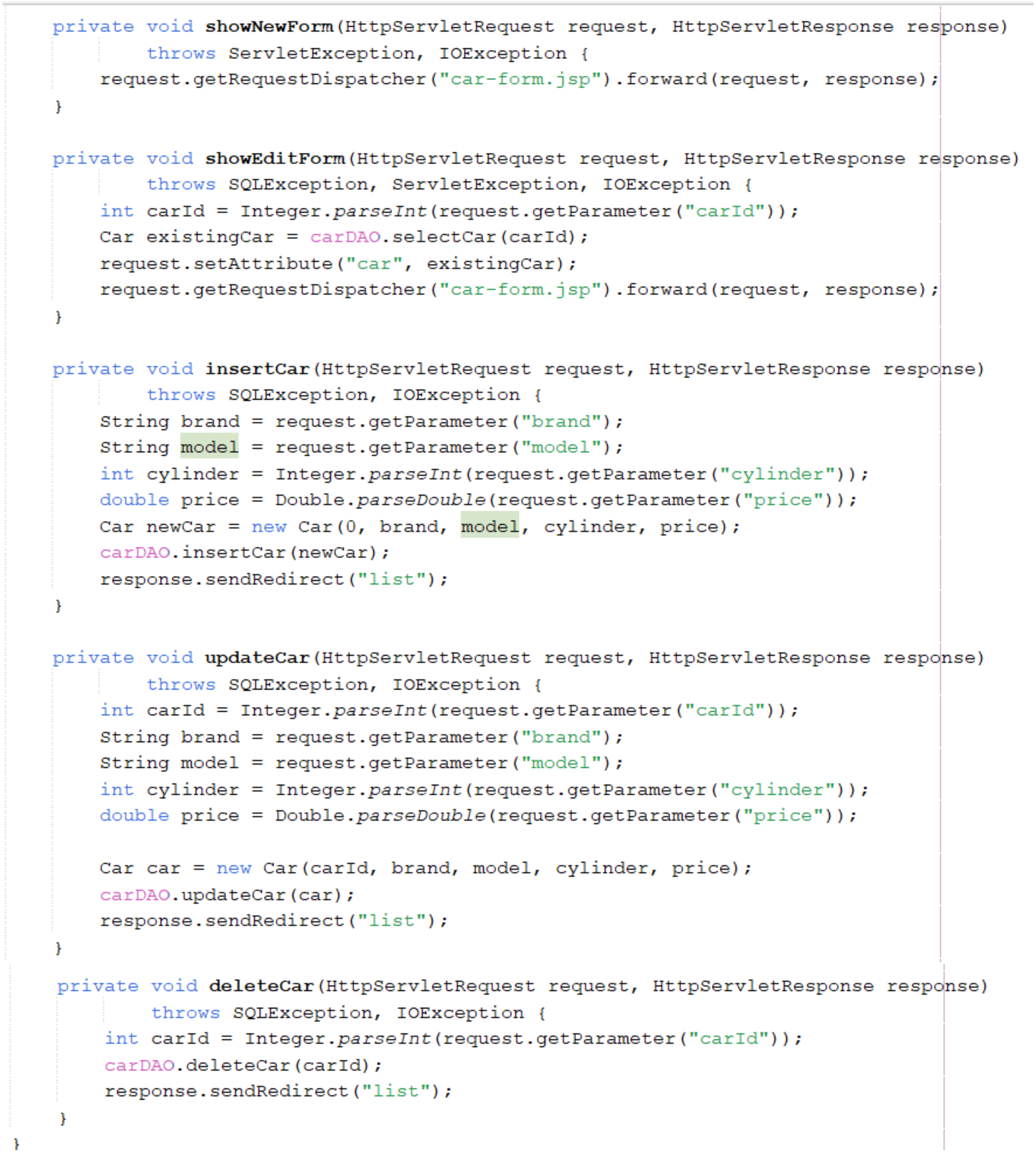
```html
</head>
<body>
    <h2>${car == null ? 'Add New Car' : 'Edit Car'}</h2>
    <form action="${car == null ? 'insert' : 'update'}" method="post">
        <input type="hidden" name="carId" value="${car == null ? '' : car.carId}" />
        <label for="brand">Brand:</label>
        <input type="text" id="brand" name="brand" value="${car == null ? '' : car.brand}" required />

        <label for="model">Model:</label>
        <input type="text" id="model" name="model" value="${car == null ? '' : car.model}" required />

        <label for="cylinder">Cylinder:</label>
        <input type="number" id="cylinder" name="cylinder" value="${car == null ? '' : car.cylinder}" required />

        <label for="price">Price:</label>
        <input type="number" step="0.01" id="price" name="price" value="${car == null ? '' : car.price}" required />

        <input type="submit" value="${car == null ? 'Add Car' : 'Update Car'}" />
    </form>
</body>
</html>
```



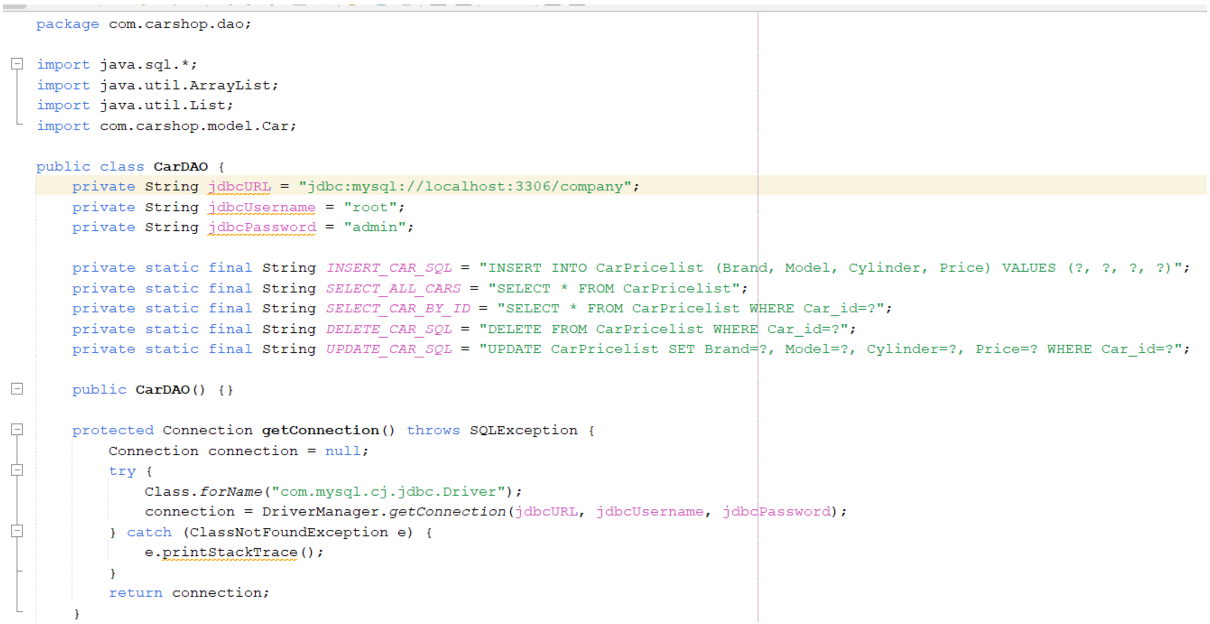
```java
package com.carshop.controller;

import com.carshop.dao.CarDAO;
import com.carshop.model.Car;
import java.io.IOException;
import java.sql.SQLException;
import java.util.List;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet("/")
public class CarControllerServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    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();
```



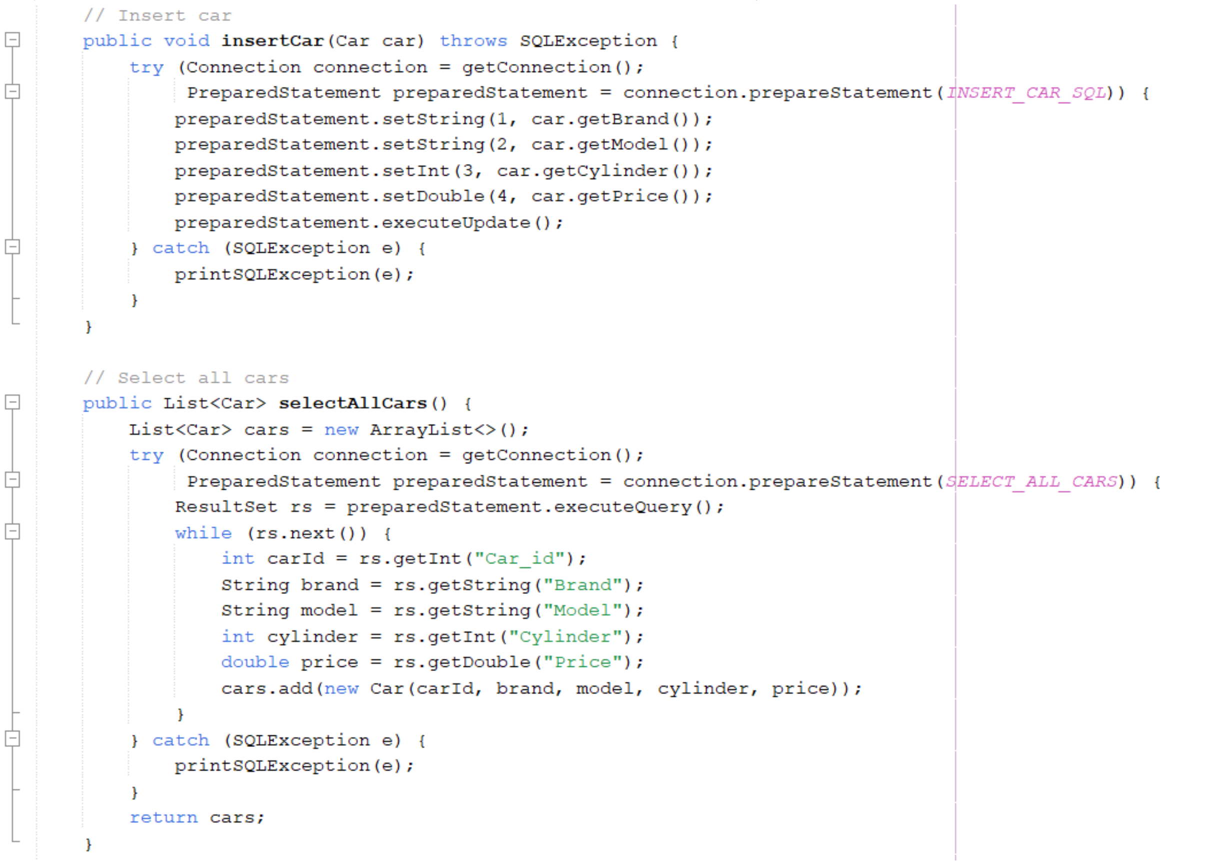
```
        try {
            switch (action) {
                case "/new":
                    showNewForm(request, response);
                    break;
                case "/insert":
                    insertCar(request, response);
                    break;
                case "/delete":
                    deleteCar(request, response);
                    break;
                case "/edit":
                    showEditForm(request, response);
                    break;
                case "/update":
                    updateCar(request, response);
                    break;
                default:
                    listCars(request, response);
                    break;
            }
        } catch (SQLException ex) {
            throw new ServletException(ex);
        }
    }

    private void listCars(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException, ServletException {
        List<Car> carList = carDAO.selectAllCars();
        request.setAttribute("carList", carList);
        request.getRequestDispatcher("car-list.jsp").forward(request, response);
    }
```



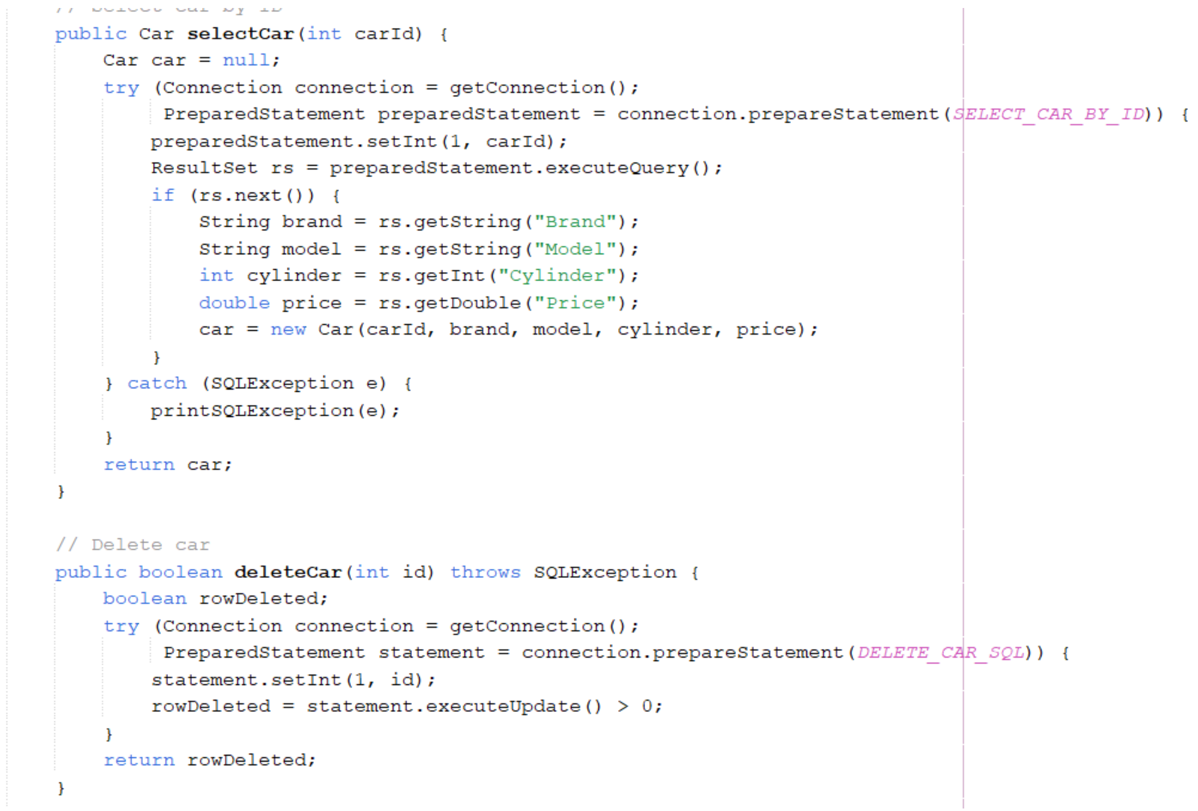
```java
    private void showNewForm(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        request.getRequestDispatcher("car-form.jsp").forward(request, response);
    }

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

    private void insertCar(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        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 newCar = new Car(0, brand, model, cylinder, price);
        carDAO.insertCar(newCar);
        response.sendRedirect("list");
    }

    private void updateCar(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        int carId = Integer.parseInt(request.getParameter("carId"));
        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(carId, brand, model, cylinder, price);
        carDAO.updateCar(car);
        response.sendRedirect("list");
    }
    private void deleteCar(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {
        int carId = Integer.parseInt(request.getParameter("carId"));
        carDAO.deleteCar(carId);
        response.sendRedirect("list");
    }
}
```



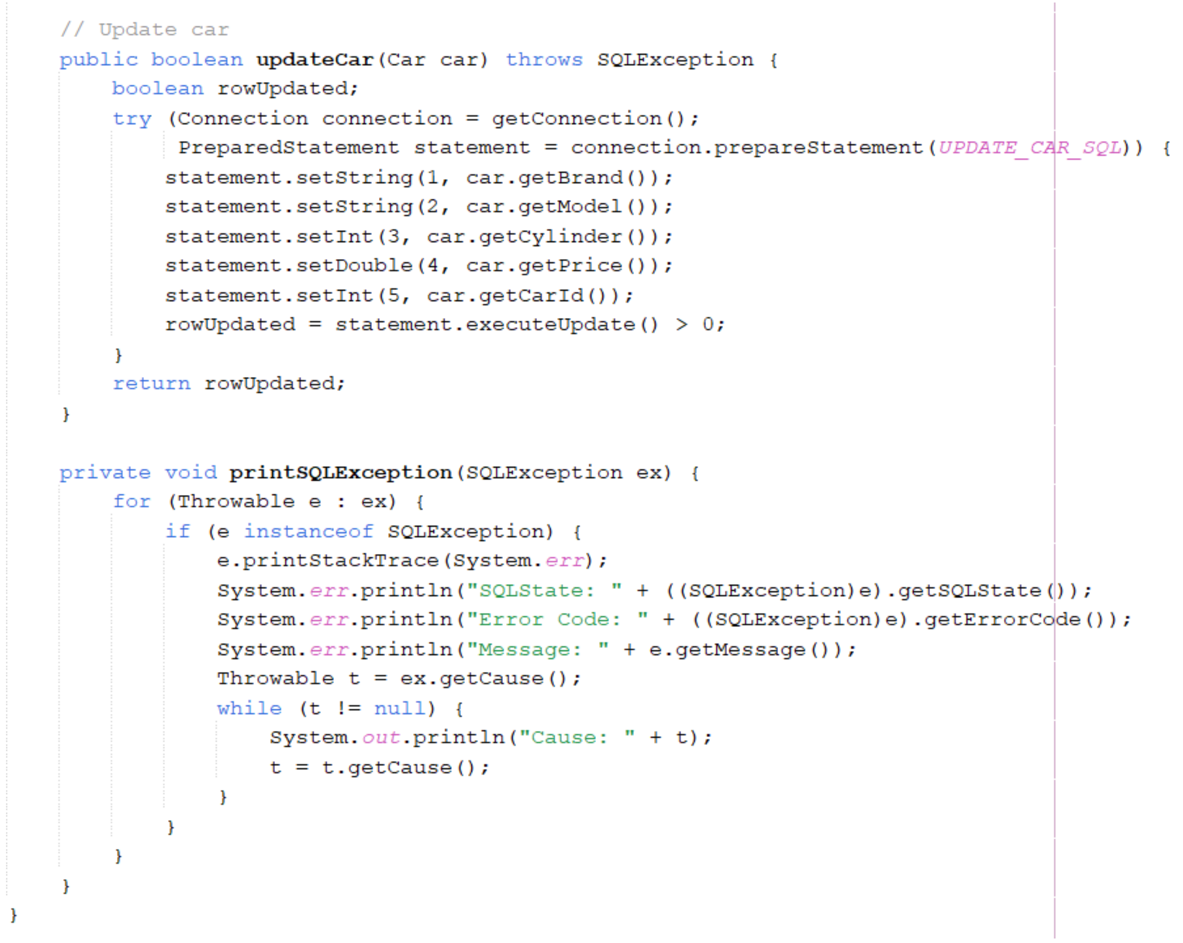
```java
package com.carshop.dao;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import com.carshop.model.Car;

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

    private static final String INSERT_CAR_SQL = "INSERT INTO CarPricelist (Brand, Model, Cylinder, Price) VALUES (?, ?, ?, ?)";
    private static final String SELECT_ALL_CARS = "SELECT * FROM CarPricelist";
    private static final String SELECT_CAR_BY_ID = "SELECT * FROM CarPricelist WHERE Car_id=?";
    private static final String DELETE_CAR_SQL = "DELETE FROM CarPricelist WHERE Car_id=?";
    private static final String UPDATE_CAR_SQL = "UPDATE CarPricelist SET Brand=?, Model=?, Cylinder=?, Price=? WHERE Car_id=?";

    public CarDAO() {}

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



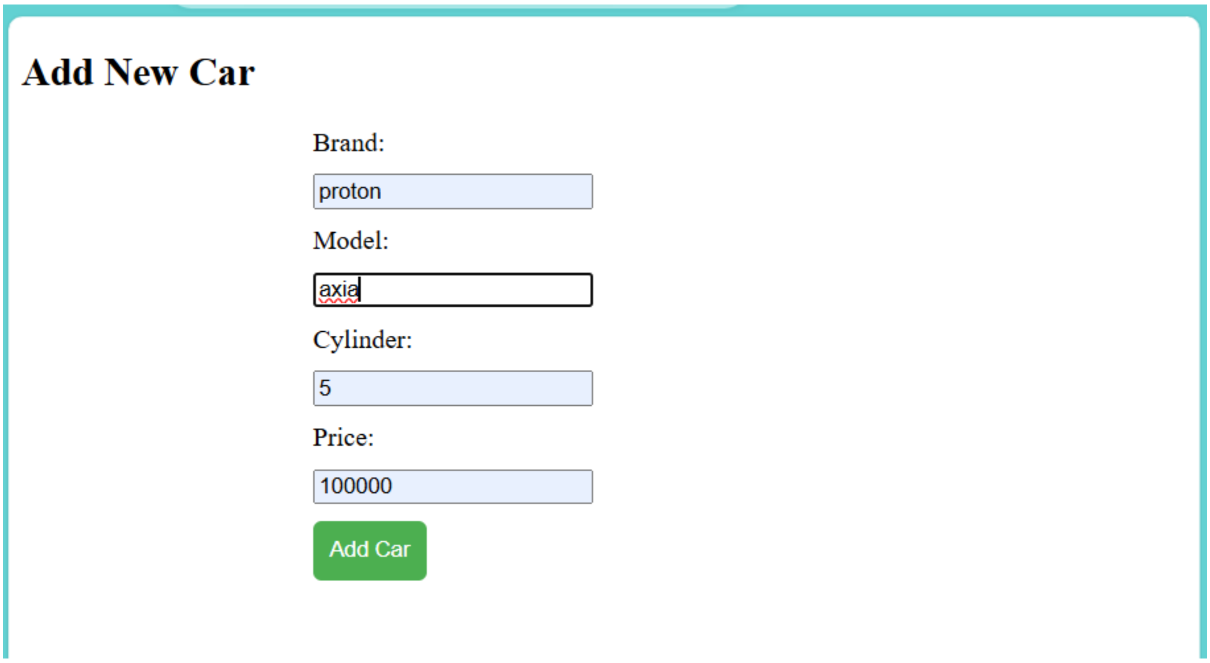
```java
    // Insert car
    public void insertCar(Car car) throws SQLException {
        try (Connection connection = getConnection();
             PreparedStatement preparedStatement = connection.prepareStatement(INSERT_CAR_SQL)) {
            preparedStatement.setString(1, car.getBrand());
     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

    // Select all cars
    public List<Car> selectAllCars() {
        List<Car> cars = new ArrayList<>();
        try (Connection connection = getConnection();
             PreparedStatement preparedStatement = connection.prepareStatement(SELECT_ALL_CARS)) {
            ResultSet rs = preparedStatement.executeQuery();
            while (rs.next()) {
                int carId = rs.getInt("Car_id");
                String brand = rs.getString("Brand");
                String model = rs.getString("Model");
                int cylinder = rs.getInt("Cylinder");
                double price = rs.getDouble("Price");
                cars.add(new Car(carId, brand, model, cylinder, price));
            }
        } catch (SQLException e) {
            printSQLException(e);
        }
        return cars;
    }
```



```
    // Select car by ID
    public Car selectCar(int carId) {
        Car car = null;
        try (Connection connection = getConnection();
             PreparedStatement preparedStatement = connection.prepareStatement(SELECT_CAR_BY_ID)) {
            preparedStatement.setInt(1, carId);
            ResultSet rs = preparedStatement.executeQuery();
            if (rs.next()) {
                String brand = rs.getString("Brand");
                String model = rs.getString("Model");
                int cylinder = rs.getInt("Cylinder");
                double price = rs.getDouble("Price");
                car = new Car(carId, brand, model, cylinder, price);
            }
        } catch (SQLException e) {
            printSQLException(e);
        }
        return car;
    }

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



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



Add New Car

Brand:

proton

Model:

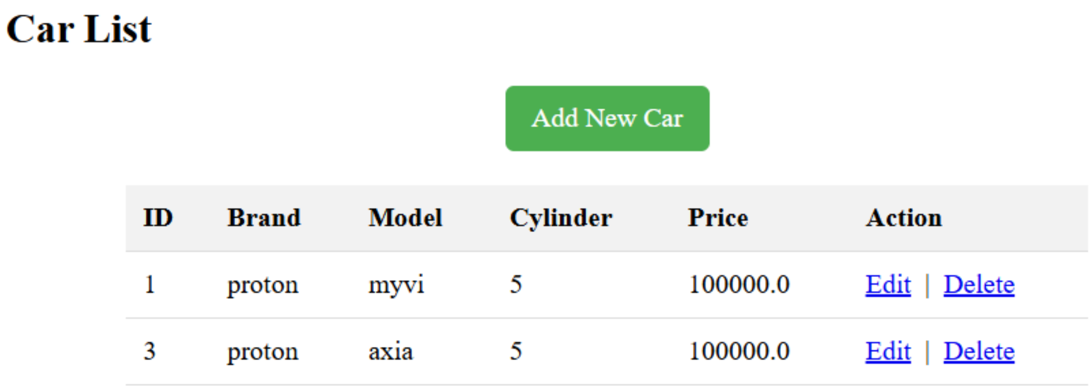
axia

Cylinder:

5

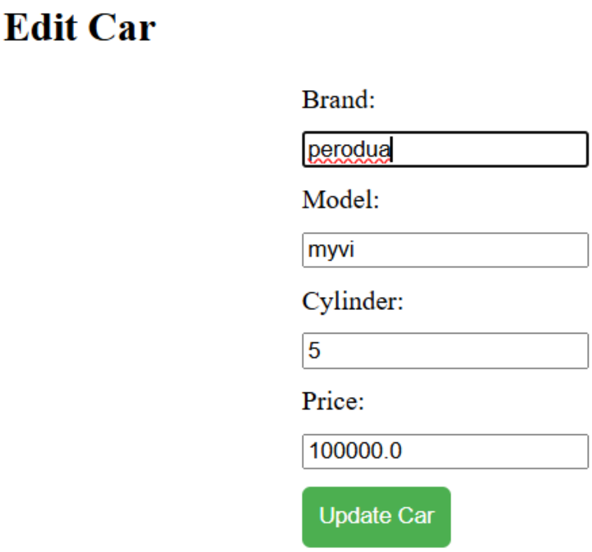
Price:

100000



Car List

ID	Brand	Model	Cylinder	Price	Action
1	proton	myvi	5	100000.0	Edit | Delete
3	proton	axia	5	100000.0	Edit | Delete



Edit Car

Brand:

perodua

Model:

myvi

Cylinder:

5

Price:

100000.0

Car List

Add New Car

ID	Brand	Model	Cylinder	Price	Action
1	perodua	myvi	5	100000.0	Edit | Delete
3	proton	axia	5	100000.0	Edit | Delete