**Code :-**

**Models :**

**Parking**

package com.parkingManagement.model;

import java.sql.Timestamp;

public class Parking {

private int id;

private String vehicleNumber;

private int duration;

private String companyName;

private Timestamp entryTime;

private Timestamp exitTime;

private double fine;

public Parking() {}

public Parking(String vehicleNumber, int duration, String companyName, Timestamp entryTime) {

this.vehicleNumber = vehicleNumber;

this.duration = duration;

this.companyName = companyName;

this.entryTime = entryTime;

this.exitTime = null; // Not set at booking

this.fine = 0.0;

}

public int getId() { return id; }

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

public String getVehicleNumber() { return vehicleNumber; }

public void setVehicleNumber(String vehicleNumber) { this.vehicleNumber = vehicleNumber; }

public int getDuration() { return duration; }

public void setDuration(int duration) { this.duration = duration; }

public String getCompanyName() { return companyName; }

public void setCompanyName(String companyName) { this.companyName = companyName; }

public Timestamp getEntryTime() { return entryTime; }

public void setEntryTime(Timestamp entryTime) { this.entryTime = entryTime; }

public Timestamp getExitTime() { return exitTime; }

public void setExitTime(Timestamp exitTime) { this.exitTime = exitTime; }

public double getFine() { return fine; }

public void setFine(double fine) { this.fine = fine; }

}.

**User :**

package com.parkingManagement.model;

public class Users {

private int id;

private int company\_id; // Added company\_id

private String company\_name;

private String company\_email;

private String company\_phone;

private String company\_address;

private int parking\_slots;

private String password;

// Updated constructor to include company\_id

public Users(String company\_name, String company\_email, String company\_phone, String company\_address, Integer parking\_slots, String password) {

this.company\_id = company\_id;

this.company\_name = company\_name;

this.company\_email = company\_email;

this.company\_phone = company\_phone;

this.company\_address = company\_address;

this.parking\_slots = Integer.parseInt(String.valueOf(parking\_slots));

this.password = password;

}

// Default constructor

public Users() {

}

// Getter for companyName

public String getCompanyName() {

return company\_name;

}

// Getter and setter methods for company\_id

public int getCompany\_id() {

return company\_id;

}

public void setCompany\_id(int company\_id) {

this.company\_id = company\_id;

}

// Getter and setter for id

public void setId(int id) {

this.id = id;

}

public int getId() {

return id;

}

// Getter and setter for company\_name

public void setCompany\_name(String company\_name) {

this.company\_name = company\_name;

}

public String getCompany\_name() {

return company\_name;

}

// Getter and setter for company\_email

public void setCompany\_email(String company\_email) {

this.company\_email = company\_email;

}

public String getCompany\_email() {

return company\_email;

}

// Getter and setter for company\_phone

public void setCompany\_phone(String company\_phone) {

this.company\_phone = company\_phone;

}

public String getCompany\_phone() {

return company\_phone;

}

// Getter and setter for company\_address

public String getCompany\_address() {

return company\_address;

}

public void setCompany\_address(String company\_address) {

this.company\_address = company\_address;

}

// Getter and setter for parking\_slots

public int getParking\_slots() {

return parking\_slots;

}

public void setParking\_slots(int parking\_slots) {

this.parking\_slots = parking\_slots;

}

// Getter and setter for password

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

**VehicleType**

package com.parkingManagement.model;

public class VehicleType {

private String id;

private String companyId;

private String vehicleType;

private String vehicleCharge;

private String fine;

// Getters and Setters

public String getId() {

return id;

}

public void setId(int id) {

this.id = String.valueOf(id);

}

public String getCompanyId() {

return companyId;

}

public void setCompanyId(int companyId) {

this.companyId = String.valueOf(companyId);

}

public String getVehicleType() {

return vehicleType;

}

public void setVehicleType(String vehicleType) {

this.vehicleType = vehicleType;

}

public String getVehicleCharge() {

return vehicleCharge;

}

public void setVehicleCharge(double vehicleCharge) {

this.vehicleCharge = String.valueOf(vehicleCharge);

}

public String getFine() {

return fine;

}

public void setFine(double fine) {

this.fine = String.valueOf(fine);

}

}

**Dao :**

**UserDao**

package com.parkingManagement.dao;

import com.parkingManagement.model.Users;

import com.parkingManagement.util.DBConnection;

import java.sql.\*;

public class UserDao {

private static final String INSERT\_USER\_SQL = "INSERT INTO users (company\_name, company\_email, company\_phone, company\_address, parking\_slots, password) VALUES (?, ?, ?, ?, ?, ?)";

private static final String SELECT\_USER\_SQL = "SELECT \* FROM users WHERE company\_email = ? AND password = ?";

private static final String SELECT\_USER\_BY\_ID\_SQL = "SELECT \* FROM users WHERE id = ?"; // New query to select user by id

// Method to register a new user

public boolean registerUser(Users user) {

try (Connection connection = DBConnection.getConnection();

PreparedStatement preparedStatement = connection.prepareStatement(INSERT\_USER\_SQL, Statement.RETURN\_GENERATED\_KEYS)) {

preparedStatement.setString(1, user.getCompany\_name());

preparedStatement.setString(2, user.getCompany\_email());

preparedStatement.setString(3, user.getCompany\_phone());

preparedStatement.setString(4, user.getCompany\_address());

preparedStatement.setInt(5, user.getParking\_slots());

preparedStatement.setString(6, user.getPassword());

int affectedRows = preparedStatement.executeUpdate();

if (affectedRows > 0) {

try (ResultSet generatedKeys = preparedStatement.getGeneratedKeys()) {

if (generatedKeys.next()) {

user.setId(generatedKeys.getInt(1)); // Setting the generated user id

return true;

}

}

}

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

// Method for user login

public Users loginUser(String email, String password) {

try (Connection connection = DBConnection.getConnection();

PreparedStatement preparedStatement = connection.prepareStatement(SELECT\_USER\_SQL)) {

preparedStatement.setString(1, email);

preparedStatement.setString(2, password);

ResultSet rs = preparedStatement.executeQuery();

if (rs.next()) {

Users user = new Users();

user.setId(rs.getInt("id")); // Set user id

user.setCompany\_name(rs.getString("company\_name"));

user.setCompany\_email(rs.getString("company\_email"));

user.setCompany\_phone(rs.getString("company\_phone"));

user.setCompany\_address(rs.getString("company\_address"));

user.setParking\_slots(rs.getInt("parking\_slots"));

user.setPassword(rs.getString("password"));

return user;

}

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

// Optional: Method to get user by ID

public Users getUserById(int userId) {

try (Connection connection = DBConnection.getConnection();

PreparedStatement preparedStatement = connection.prepareStatement(SELECT\_USER\_BY\_ID\_SQL)) {

preparedStatement.setInt(1, userId);

ResultSet rs = preparedStatement.executeQuery();

if (rs.next()) {

Users user = new Users();

user.setId(rs.getInt("id"));

user.setCompany\_name(rs.getString("company\_name"));

user.setCompany\_email(rs.getString("company\_email"));

user.setCompany\_phone(rs.getString("company\_phone"));

user.setCompany\_address(rs.getString("company\_address"));

user.setParking\_slots(rs.getInt("parking\_slots"));

user.setPassword(rs.getString("password"));

return user;

}

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

**ParkingDao**

package com.parkingManagement.dao;

import com.parkingManagement.model.Parking;

import com.parkingManagement.util.DBConnection;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class ParkingDAO {

// Retrieve all parkings for a specific company

public List<Parking> getAllParkings(String companyName) {

List<Parking> parkingList = new ArrayList<>();

String query = "SELECT \* FROM parking WHERE company\_name=?";

try (Connection conn = DBConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, companyName);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

Parking parking = new Parking();

parking.setId(rs.getInt("id"));

parking.setVehicleNumber(rs.getString("vehicle\_number"));

parking.setDuration(rs.getInt("duration"));

parking.setCompanyName(rs.getString("company\_name"));

parking.setEntryTime(rs.getTimestamp("entry\_time"));

parking.setExitTime(rs.getTimestamp("exit\_time"));

parking.setFine(rs.getDouble("fine"));

parkingList.add(parking);

}

} catch (SQLException e) {

e.printStackTrace();

}

return parkingList;

}

// Add new parking record

public void addParking(Parking parking) {

String query = "INSERT INTO parking (vehicle\_number, duration, company\_name, entry\_time, exit\_time, fine) VALUES (?, ?, ?, ?, ?, ?)";

try (Connection conn = DBConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, parking.getVehicleNumber());

stmt.setInt(2, parking.getDuration());

stmt.setString(3, parking.getCompanyName());

stmt.setTimestamp(4, parking.getEntryTime());

stmt.setTimestamp(5, parking.getExitTime());

stmt.setDouble(6, parking.getFine());

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

// Update parking record

public void updateParking(Parking parking) {

String query = "UPDATE parking SET vehicle\_number=?, duration=?, entry\_time=?, exit\_time=?, fine=? WHERE id=?";

try (Connection conn = DBConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, parking.getVehicleNumber());

stmt.setInt(2, parking.getDuration());

stmt.setTimestamp(3, parking.getEntryTime());

stmt.setTimestamp(4, parking.getExitTime());

stmt.setDouble(5, parking.getFine());

stmt.setInt(6, parking.getId());

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

// Delete parking record

public void deleteParking(int id) {

String query = "DELETE FROM parking WHERE id=?";

try (Connection conn = DBConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setInt(1, id);

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

**VehicleTypeDao**

package com.parkingManagement.dao;

import com.parkingManagement.model.VehicleType;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class VehicleTypeDAO {

private Connection connection;

public VehicleTypeDAO(Connection connection) {

this.connection = connection;

}

// Method to save vehicle pricing to the database

public boolean saveVehiclePricing(VehicleType vehicleType) {

String query = "INSERT INTO vehicle\_types (company\_id, vehicle\_type, vehicle\_charge, fine) VALUES (?, ?, ?, ?)";

try (PreparedStatement stmt = connection.prepareStatement(query)) {

stmt.setString(1, vehicleType.getCompanyId());

stmt.setString(2, vehicleType.getVehicleType());

stmt.setString(3, vehicleType.getVehicleCharge());

stmt.setString(4, vehicleType.getFine());

return stmt.executeUpdate() > 0;

} catch (SQLException e) {

e.printStackTrace();

return false;

}

}

// Method to get all vehicle types for a specific company

public List<VehicleType> getVehicleTypesByCompanyId(int companyId) {

List<VehicleType> vehicleTypes = new ArrayList<>();

String query = "SELECT \* FROM vehicle\_types WHERE company\_id = ?";

try (PreparedStatement stmt = connection.prepareStatement(query)) {

stmt.setInt(1, companyId);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

VehicleType vehicleType = new VehicleType();

vehicleType.setId(rs.getInt("id"));

vehicleType.setCompanyId(rs.getInt("company\_id"));

vehicleType.setVehicleType(rs.getString("vehicle\_type"));

vehicleType.setVehicleCharge(rs.getDouble("vehicle\_charge"));

vehicleType.setFine(rs.getDouble("fine"));

vehicleTypes.add(vehicleType);

}

} catch (SQLException e) {

e.printStackTrace();

}

return vehicleTypes;

}

}

**Servlets :**

**Login**

**package com.parkingManagement.servlets;**

**import com.parkingManagement.dao.UserDao;**

**import com.parkingManagement.model.Users;**

**import java.io.IOException;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**@WebServlet("/login")**

**public class Login extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**private UserDao userDao = new UserDao();**

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

**String email = request.getParameter("email");**

**String password = request.getParameter("password");**

**Users user = userDao.loginUser(email, password);**

**if (user != null) {**

**HttpSession session = request.getSession();**

**session.setAttribute("user", user);**

**response.sendRedirect("dashboard.jsp");**

**} else {**

**response.sendRedirect("SignIn-page.jsp.jsp?error=Invalid credentials");**

**}**

**}**

**}**

**Logout**

**package com.parkingManagement.servlets;**

**import java.io.IOException;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**@WebServlet("/logout")**

**public class LogoutServlet extends HttpServlet {**

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

**HttpSession session = request.getSession(false);**

**if (session != null) {**

**session.invalidate(); // Destroy the session**

**}**

**response.sendRedirect("SignIn-page.jsp"); // Redirect to login page after logout**

**}**

**}**

**Parking**

**package com.parkingManagement.servlets;**

**import com.parkingManagement.dao.ParkingDAO;**

**import java.io.IOException;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**@WebServlet("/ParkingServlet")**

**public class ParkingServlet extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**private ParkingDAO parkingDAO = new ParkingDAO();**

**protected void doPost(HttpServletRequest request, HttpServletResponse response)**

**throws ServletException, IOException {**

**String action = request.getParameter("action");**

**if ("delete".equals(action)) {**

**int id = Integer.parseInt(request.getParameter("id"));**

**parkingDAO.deleteParking(id);**

**}**

**}**

**}**

**Signup**

package com.parkingManagement.servlets;

import com.parkingManagement.dao.UserDao;

import com.parkingManagement.model.Users;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/signup")

public class SignUp extends HttpServlet {

private static final long serialVersionUID = 1L;

private UserDao userDao = new UserDao();

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

String companyName = request.getParameter("company\_name");

String companyEmail = request.getParameter("company\_email");

String companyPhone = request.getParameter("company\_phone");

String companyAddress = request.getParameter("company\_address");

int parkingSlots = Integer.parseInt(request.getParameter("parking\_slots"));

String password = request.getParameter("password");

Users newUser = new Users(companyName, companyEmail, companyPhone, companyAddress, parkingSlots, password);

if (userDao.registerUser(newUser)) {

response.sendRedirect("SignIn-page.jsp");

} else {

response.sendRedirect("signup.jsp?error=Registration failed");

}

}

}

**VehicleServlet**

package com.parkingManagement.servlets;

import com.parkingManagement.dao.VehicleTypeDAO;

import com.parkingManagement.model.Users;

import com.parkingManagement.model.VehicleType;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

import java.sql.Connection;

import java.util.List;

@WebServlet("/Vehicle")

public class VehicleServlet extends HttpServlet {

@Override

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

try {

// Get the user session

HttpSession session = request.getSession();

Users user = (Users) session.getAttribute("user");

// Check if user session exists

if (user == null) {

response.getWriter().println("Error: User session not found. Please log in.");

return;

}

int companyId = user.getId();

if (companyId <= 0) {

response.getWriter().println("Error: Invalid company ID.");

return;

}

// Get vehicle pricing details from the form

String[] vehicleTypes = request.getParameterValues("vehicle\_type[]");

String[] prices = request.getParameterValues("price[]");

String[] fines = request.getParameterValues("fine[]");

// Get database connection

Connection connection = (Connection) getServletContext().getAttribute("DBConnection");

if (connection == null) {

response.getWriter().println("Error: Database connection not found.");

return;

}

VehicleTypeDAO vehicleTypeDAO = new VehicleTypeDAO(connection);

// Loop through and save each vehicle type

for (int i = 0; i < vehicleTypes.length; i++) {

VehicleType vehicleType = new VehicleType();

vehicleType.setCompanyId(companyId);

vehicleType.setVehicleType(vehicleTypes[i]);

vehicleType.setVehicleCharge(Double.parseDouble(prices[i]));

vehicleType.setFine(Double.parseDouble(fines[i]));

boolean isInserted = vehicleTypeDAO.saveVehiclePricing(vehicleType);

System.out.println("Inserted: " + isInserted + " for " + vehicleTypes[i]);

if (!isInserted) {

response.getWriter().println("Error: Could not insert " + vehicleTypes[i]);

return;

}

}

// Redirect after saving

response.sendRedirect("vehicles.jsp");

} catch (Exception e) {

e.printStackTrace();

response.sendError(HttpServletResponse.SC\_INTERNAL\_SERVER\_ERROR, "Error while processing the request");

}

}

}

**Util :**

**DBConnection**

package com.parkingManagement.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

private static final String URL = "jdbc:mysql://localhost:3306/parking\_management";

private static final String USER = "root";

private static final String PASSWORD = "";

static{

try{

Class.forName("com.mysql.cj.jdbc.Driver");

}

catch (ClassNotFoundException e) {

throw new RuntimeException(e);

}

}

public static Connection getConnection() throws SQLException {

return DriverManager.getConnection(URL, USER, PASSWORD);

}

}

**Dashboard.jsp**

<%@ page import="com.parkingManagement.model.Users, com.parkingManagement.model.Parking, com.parkingManagement.dao.ParkingDAO, java.util.List" %>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%

// Check if session exists, else redirect to login page

Users user = (Users) session.getAttribute("user");

if (user == null) {

response.sendRedirect("SignIn-page.jsp");

return;

}

// Fetch parking details from database

ParkingDAO parkingDAO = new ParkingDAO();

List<Parking> parkingList = parkingDAO.getAllParkings(user.getCompanyName());

%>

<!DOCTYPE html>

<html>

<head>

<title>Company Portal</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.navbar {

background-color: #007bff;

color: white;

padding: 15px;

display: flex;

justify-content: space-between;

align-items: center;

}

.navbar .company-name {

font-size: 22px;

font-weight: bold;

}

.navbar .nav-buttons {

display: flex;

}

.navbar .btn {

padding: 10px 15px;

margin-left: 10px;

border: none;

cursor: pointer;

border-radius: 5px;

color: white;

font-size: 14px;

text-decoration: none;

}

.add-vehicle { background-color: #28a745; }

.add-parking { background-color: #17a2b8; }

.logout-btn { background-color: red; }

.btn:hover { opacity: 0.8; }

.container {

width: 80%;

margin: 50px auto;

background: white;

padding: 20px;

border-radius: 10px;

box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

table, th, td {

border: 1px solid #ddd;

}

th, td {

padding: 10px;

text-align: center;

}

th {

background-color: #007bff;

color: white;

}

.action-btn {

padding: 5px 10px;

border: none;

cursor: pointer;

border-radius: 5px;

font-size: 14px;

}

.edit-btn { background-color: #ffc107; }

.delete-btn { background-color: #dc3545; }

</style>

</head>

<body>

<div class="navbar">

<div class="company-name"><%= user.getCompanyName() %></div>

<div class="nav-buttons">

<a href="vehicles.jsp" class="btn add-vehicle">Add Vehicle</a>

<a href="Parking.jsp" class="btn add-parking">Add Parking</a>

<form action="logout" method="post" style="display: inline;">

<button type="submit" class="btn logout-btn">Logout</button>

</form>

</div>

</div>

<div class="container">

<h2>Parked Vehicles</h2>

<table>

<tr>

<th>ID</th>

<th>Vehicle Number</th>

<th>Duration (mins)</th>

<th>Entry Time</th>

<th>Exit Time</th>

<th>Fine</th>

<th>Actions</th>

</tr>

<% for (Parking parking : parkingList) { %>

<tr>

<td><%= parking.getId() %></td>

<td><%= parking.getVehicleNumber() %></td>

<td><%= parking.getDuration() %></td>

<td><%= parking.getEntryTime() %></td>

<td><%= parking.getExitTime() != null ? parking.getExitTime() : "N/A" %></td>

<td><%= parking.getFine() %></td>

<td>

<a href="<%= parking.getId() %>" class="action-btn edit-btn">Edit</a>

<a href="ParkingServlet?action=delete&id=<%= parking.getId() %>" class="action-btn delete-btn" onclick="return confirm('Are you sure?')">Delete</a>

</td>

</tr>

<% } %>

</table>

</div>

</body>

</html>

**Editparking.jsp**

<%@ page import="com.parkingManagement.dao.ParkingDAO, com.parkingManagement.model.Parking" %>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%

int parkingId = Integer.parseInt(request.getParameter("id"));

ParkingDAO parkingDAO = new ParkingDAO();

Parking parking = parkingDAO.getParkingById(parkingId);

%>

<!DOCTYPE html>

<html>

<head>

<title>Edit Parking</title>

</head>

<body>

<h2>Edit Parking Details</h2>

<form action="ParkingServlet" method="post">

<input type="hidden" name="id" value="<%= parking.getId() %>">

Vehicle Number: <input type="text" name="vehicleNumber" value="<%= parking.getVehicleNumber() %>" required><br>

Duration (mins): <input type="number" name="duration" value="<%= parking.getDuration() %>" required><br>

<input type="hidden" name="action" value="update">

<button type="submit">Update Parking</button>

</form>

</body>

</html>

**Parking.jsp**

<%@ page import="com.parkingManagement.model.Users" %>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%

// Check if session exists, else redirect to login page

Users user = (Users) session.getAttribute("user");

if (user == null) {

response.sendRedirect("SignIn-page.jsp");

return;

}

%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Park Vehicle</title>

</head>

<body>

<h2>Vehicle Parking Form</h2>

<form action="ParkingServlet" method="post">

<input type="hidden" name="action" value="add"> <!-- Action type for servlet -->

<label>Vehicle Number:</label>

<input type="text" name="vehicleNumber" required><br><br>

<label>Duration (minutes):</label>

<input type="number" name="duration" required><br><br>

<label>Entry Time:</label>

<input type="datetime-local" name="entryTime" required><br><br>

<!-- Hidden input for company name (stored as per database field) -->

<input name="companyName" value="<%= user.getCompanyName() %>">

<!-- Exit time is initially NULL, so it won't be included in the form -->

<!-- Fine is default 0.0 and will be managed via backend logic -->

<input type="submit" value="Park Vehicle">

</form>

</body>

</html>

**Signin-page.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

</head>

<body>

<form action="login" method="post">

Email: <input type="email" name="email" required><br>

Password: <input type="password" name="password" required><br>

<button type="submit">Login</button>

</form>

</body>

</html>

**SignUp-page.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Sign Up</title>

</head>

<body>

<form action="signup" method="post">

Company Name: <input type="text" name="company\_name" required><br>

Email: <input type="email" name="company\_email" required><br>

Phone: <input type="text" name="company\_phone" required><br>

Address: <input type="text" name="company\_address" required><br>

Parking Slots: <input type="number" name="parking\_slots" required><br>

Password: <input type="password" name="password" required><br>

<button type="submit">Sign Up</button>

</form>

</body>

</html>

**Vehicles.jsp**

**<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>**

**<%@ page import="com.parkingManagement.model.Users" %>**

**<%**

**// Check if session exists, else redirect to login page**

**Users user = (Users) session.getAttribute("user");**

**if (user == null) {**

**response.sendRedirect("SignIn-page.jsp");**

**return;**

**}**

**%>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Admin Panel - Vehicle Pricing</title>**

**<style>**

**body {**

**font-family: Arial, sans-serif;**

**background-color: #f4f4f4;**

**margin: 0;**

**padding: 0;**

**}**

**.navbar {**

**background-color: #007bff;**

**color: white;**

**padding: 15px;**

**display: flex;**

**justify-content: space-between;**

**align-items: center;**

**}**

**.navbar .company-name {**

**font-size: 22px;**

**font-weight: bold;**

**}**

**.navbar .nav-buttons {**

**display: flex;**

**}**

**.navbar .btn {**

**padding: 10px 15px;**

**margin-left: 10px;**

**border: none;**

**cursor: pointer;**

**border-radius: 5px;**

**color: white;**

**font-size: 14px;**

**text-decoration: none;**

**}**

**.add-vehicle { background-color: #28a745; }**

**.add-parking { background-color: #17a2b8; }**

**.logout-btn { background-color: red; }**

**.btn:hover { opacity: 0.8; }**

**</style>**

**<script>**

**// Function to dynamically add vehicle pricing fields**

**function addPricingFields() {**

**var container = document.getElementById("pricing-items");**

**var newItem = document.createElement("div");**

**newItem.className = "pricing-item";**

**newItem.innerHTML = `**

**<input type="text" name="vehicle\_type[]" placeholder="Vehicle Type" required>**

**<input type="number" name="price[]" placeholder="Price" min="0" step="0.01" required>**

**<input type="number" name="fine[]" placeholder="Fine Amount" min="0" step="0.01" required>**

**<button type="button" onclick="this.parentElement.remove()">Remove</button>**

**`;**

**container.appendChild(newItem);**

**}**

**// Add first item when page loads**

**window.onload = function() {**

**addPricingFields();**

**};**

**</script>**

**</head>**

**<body>**

**<div class="navbar">**

**<div class="company-name"><%= user.getCompanyName() %></div>**

**<div class="nav-buttons">**

**<a href="vehicles.jsp" class="btn add-vehicle">Add Vehicle</a>**

**<a href="Parking.jsp" class="btn add-parking">Add Parking</a>**

**<form action="logout" method="post" style="display: inline;">**

**<button type="submit" class="btn logout-btn">Logout</button>**

**</form>**

**</div>**

**</div>**

**<div class="pricing-container">**

**<h2 class="pricing-header">Vehicle Pricing Setup</h2>**

**<form action="Vehicle" method="POST">**

**<div class="pricing-table-header">**

**<div class="pricing-item" style="background-color: #f2f2f2; font-weight: bold;">**

**<span style="flex: 1; padding: 0 8px;">Vehicle Type</span>**

**<span style="flex: 1; padding: 0 8px;">Parking Price (Per Day)</span>**

**<span style="flex: 1; padding: 0 8px;">Fine Amount (Per Day)</span>**

**<span style="width: 80px;"></span>**

**</div>**

**</div>**

**<div id="pricing-items">**

**<!-- Dynamic pricing items will be added here -->**

**</div>**

**<div class="button-row">**

**<button type="button" class="add-button" onclick="addPricingFields()">Add Vehicle Type</button>**

**<button type="submit" class="save-button">Save Pricing</button>**

**</div>**

**</form>**

**</div>**

**</body>**

**</html>**

**OUTPUT**













