

Railway Crossing Status- Source Code

Servlet Files:

1.AddCrossingServlet

```
package com.Project;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class AddCrossingServlet extends HttpServlet {

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

        // Retrieve the form data from the request
        String name = request.getParameter("name");
        String address = request.getParameter("address");
        String landmark = request.getParameter("landmark");
        String trainSchedules = request.getParameter("trainSchedule");
        String personInCharge = request.getParameter("personInCharge");
        String status = request.getParameter("crossing_status");

        // Create a new RailwayCrossing object with the form data
        RailwayCrossing crossing = new RailwayCrossing(name, address, landmark,
        trainSchedules, personInCharge, status);

        // Save the new crossing to the database using RailwayCrossingDAO
        RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
        crossingDAO.addCrossing(crossing);

        // Redirect to the admin homepage after adding the crossing
        response.sendRedirect("adminHome.jsp");
    }
}
```

2.AddToFavoritesServlet

```
package com.Project;
```

```
import java.io.IOException;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
public class AddToFavoritesServlet extends HttpServlet {
```

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

```
        throws ServletException, IOException {
```

```
        String crossingName = request.getParameter("crossingName"); // Assuming the parameter  
        name is "crossingName"
```

```
        if (crossingName != null && !crossingName.isEmpty()) {
```

```
            // Perform the database operation to add the crossing to favorites
```

```
            RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
```

```
            crossingDAO.addToFavorites(crossingName); // Implement this method in  
            RailwayCrossingDAO
```

```
        }
```

```
        // Redirect back to the userHome.jsp page
```

```
        response.sendRedirect("userHome.jsp");
```

```
    }
```

```
}
```

3.AdminHomePageServlet

```
package com.Project;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import java.util.List;
```

```

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class AdminHomePageServlet extends HttpServlet {
    private RailwayCrossingDAO crossingDAO;
    public void init() {
        crossingDAO = new RailwayCrossingDAO();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response)
    throws ServletException, IOException {
        // Fetch all railway crossing details from the database
        List<RailwayCrossing> crossings = crossingDAO.getAllCrossings();
        // Pass the crossing details to the JSP page
        request.setAttribute("crossings", crossings);
        request.getRequestDispatcher("adminHome.jsp").forward(request,
        response);

    }
}

```

4.AdminLoginServlet

```

package com.Project;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```

public class AdminLoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String ADMIN_ID = "Deepu";
    private static final String ADMIN_PASSWORD = "Deepu12*";
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response)
    throws ServletException, IOException {
        String email = request.getParameter("email");
        String password = request.getParameter("password");
        if (email != null && email.equals(ADMIN_ID) && password != null &&
        password.equals(ADMIN_PASSWORD)) {
            response.sendRedirect("adminHome.jsp");
        } else {
            request.getRequestDispatcher("adminLogin.jsp").forward(request,
            response);
        }
    }
}

```

5.DeleteCrossingServlet

```

package com.Project;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class DeleteCrossingServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            // Retrieve the crossing name from the request

```

```

String crossingName = request.getParameter("name");

if (crossingName != null && !crossingName.isEmpty()) {
    // Perform the delete operation
    RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
    crossingDAO.deleteCrossing(crossingName);
    // Redirect to the admin homepage after the deletion
    response.sendRedirect("adminHome.jsp");
} else {
    // Handle case where crossingName is empty or null
    throw new ServletException("Crossing name is missing.");
}
} catch (Exception e) {
    // Handle exceptions
    throw new ServletException("An error occurred during the crossing deletion.", e);
}
}
}

```

6.LoginServlet

```

package com.Project;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
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 LoginServlet extends HttpServlet {
    // JDBC URL, username, and password
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/railway";
    private static final String JDBC_USER = "root";
    private static final String JDBC_PASSWORD = "Deepu12*";

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Retrieve the form data
        String email = request.getParameter("email");
        String password = request.getParameter("password");

        // Establish database connection
        try (Connection connection = DriverManager.getConnection(JDBC_URL, JDBC_USER,
            JDBC_PASSWORD)) {
            // Prepare SQL statement to check user credentials
            String sql = "SELECT * FROM users WHERE email = ? AND password = ?";
            try (PreparedStatement statement = connection.prepareStatement(sql)) {
                statement.setString(1, email);
                statement.setString(2, password);

                // Execute query
                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        // Authentication successful
                        // Create User object
                        User user = new User(email, password);
                    }
                }
            }
        }
    }
}

```

```

        HttpSession session = request.getSession();
        session.setAttribute("user", user);

        // Redirect to user home page
        response.sendRedirect("userHome.jsp");
        return; // End method execution after redirect
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}

// Authentication failed, redirect to error page
response.sendRedirect("userRegister.jsp?error=1");
}
}

```

7.RemoveFromFavoriteServlet

```
package com.Project;
```

```

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(name = "removeFromFavoritesServlet", urlPatterns = { "/removeFromFavorites"})
public class RemoveFromFavoritesServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)

```

```

        throws ServletException, IOException {
    String crossingName = request.getParameter("crossingName"); // Assuming the parameter
name is "crossingName"
    if (crossingName != null && !crossingName.isEmpty()) {
        // Perform the database operation to remove the crossing from favorites
        RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
        crossingDAO.removeFromFavorites(crossingName); // Implement this method in
RailwayCrossingDAO
    }
    // Redirect back to the userHome.jsp page
    response.sendRedirect("userHome.jsp");
}
}

```

8.SearchFromCrossingServlet

```
package com.Project;
```

```

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```

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(name = "searchCrossingServlet", urlPatterns = { "/searchCrossing"})
public class SearchCrossingServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)

```



```

        throws ServletException, IOException {
// Retrieve the search ID from the request
String searchName = request.getParameter("searchName");

// Retrieve crossing details by ID from the database
RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
RailwayCrossing crossing = crossingDAO.getCrossingByName(searchName);

// Set the search result in the request attribute
request.setAttribute("crossing", crossing);

// Redirect to the appropriate page based on user role
String role = (String) request.getSession().getAttribute("role");
if (role != null && role.equals("admin")) {
    request.getRequestDispatcher("adminHome.jsp").forward(request, response);
} else {
    request.getRequestDispatcher("userHome.jsp").forward(request, response);
}
}
}

```

9.UpdateCrossingServlet

```

package com.Project;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class UpdateCrossingServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)

```

```
throws ServletException, IOException {
try {
    // Retrieve the crossing name from the request parameter
    String name = request.getParameter("name");

    // Retrieve the updated details for the crossing
    String crossingStatus = request.getParameter("crossingStatus");
    String personInCharge = request.getParameter("personInCharge");
    String trainSchedules = request.getParameter("trainSchedules");
    String landmark = request.getParameter("landmark");
    String address = request.getParameter("address");

    // Create a new RailwayCrossing object with the updated details
    RailwayCrossing updatedCrossing = new RailwayCrossing();
    updatedCrossing.setName(name);
    updatedCrossing.setCrossing_Status(crossingStatus);
    updatedCrossing.setPersonInCharge(personInCharge);
    updatedCrossing.setTrainSchedule(trainSchedules);
    updatedCrossing.setLandmark(landmark);
    updatedCrossing.setAddress(address);

    // Perform the update operation
    RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
    crossingDAO.updateCrossing(updatedCrossing);

    // Redirect to the admin homepage after the update
    response.sendRedirect("adminHome.jsp");
} catch (Exception e) {
    // Handle exceptions
    e.printStackTrace();
    throw new ServletException("An error occurred during the crossing update.", e);
}
```

```
    }  
  }  
}
```

10. RegisterServlet

```
package com.Project;
```

```
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 java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;
```

```
@WebServlet("/register")
```

```
public class RegisterServlet extends HttpServlet {
```

```
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/railway";
```

```
    private static final String JDBC_USER = "root";
```

```
    private static final String JDBC_PASSWORD = "Deepu12*";
```

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

```
        throws ServletException, IOException {
```

```
        // Retrieve the form data
```

```
        String name = request.getParameter("name");
```

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

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

```

// Establish database connection
try (Connection connection = DriverManager.getConnection(JDBC_URL, JDBC_USER,
JDBC_PASSWORD)) {
    // Prepare SQL statement to insert user data into the 'users' table
    String sql = "INSERT INTO users ( email, password) VALUES ( ?, ?)";
    try (PreparedStatement statement = connection.prepareStatement(sql)) {
        statement.setString(1, email);
        statement.setString(2, password);

        // Execute the SQL statement to insert user data
        statement.executeUpdate();
    }
} catch (SQLException e) {
    e.printStackTrace();
}

// Redirect to login page after successful registration
response.sendRedirect("userLogin.jsp");
}
}

```

Class Files:

1. User

```

package com.Project;

public class User {
    private String email;
    private String password;
    public User() {
        super();
    }
}

```

```
public User( String email, String password) {
```

```
    this.email = email;
```

```
    this.password = password;
```

```
}
```

```
public String getEmail() {
```

```
    return email;
```

```
}
```

```
public void setEmail(String email) {
```

```
    this.email = email;
```

```
}
```

```
public String getPassword() {
```

```
    return password;
```

```
}
```

```
public void setPassword(String password) {
```

```
    this.password = password;
```

```
}
```

```
}
```

2. UserDao

```
package com.Project;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
public class UserDao {
```

```
    private Connection connection;
```

```
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/railway";
```

```
    private static final String JDBC_USER = "root";
```

```

private static final String JDBC_PASSWORD = "Deepu12*";

private static final String AUTHENTICATE_USER_QUERY = "SELECT * FROM users
WHERE email = ? AND password = ?";
private static final String REGISTER_USER_QUERY = "INSERT INTO users (name,
email, password) VALUES (?, ?, ?)";

public User authenticateUser(String email, String password) {
    try (Connection connection = DriverManager.getConnection(JDBC_URL,
JDBC_USER, JDBC_PASSWORD);
        PreparedStatement statement =
connection.prepareStatement(AUTHENTICATE_USER_QUERY)) {
        statement.setString(1, email);
        statement.setString(2, password);
        try (ResultSet resultSet = statement.executeQuery()) {
            if (resultSet.next())
            {
                return new User( email, password);
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}

public void registerUser(User user) {
    try (Connection connection = DriverManager.getConnection(JDBC_URL,
JDBC_USER, JDBC_PASSWORD);
        PreparedStatement statement =
connection.prepareStatement(REGISTER_USER_QUERY)) {
        statement.setString(1, user.getEmail());
        statement.setString(2, user.getPassword());
        statement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public User getUserByEmail(String email) {
    try {
        String query = "SELECT * FROM user_signup WHERE email=?";
        PreparedStatement statement = connection.prepareStatement(query);
        statement.setString(1, email);
    }
}

```

```

ResultSet resultSet = statement.executeQuery();
if (resultSet.next()) {
    User user = new User();
    user.setEmail(resultSet.getString("email"));
    user.setPassword(resultSet.getString("password"));
    return user;
}
} catch (SQLException e) {
    e.printStackTrace();
}
}
return null;
}}

```

3. RailwayCrossing

```

package com.Project;
public class RailwayCrossing {
    private String name;
    private String address;
    private String landmark;
    private String trainSchedule;
    private String personInCharge;
    private String crossing_status;
    public RailwayCrossing() {
        super();
        // TODO Auto-generated constructor stub
    }
    public RailwayCrossing(
        String name, String address, String
        landmark, String trainSchedule,
        String personInCharge, String status) {
        super();
        this.name = name;
        this.address = address;
        this.landmark = landmark;
        this.trainSchedule = trainSchedule;
        this.personInCharge = personInCharge;
        this.crossing_status = status;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}

```

```

    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getLandmark() {
        return landmark;
    }
    public void setLandmark(String landmark) {
        this.landmark = landmark;
    }
    public String getTrainSchedule() {
        return trainSchedule;
    }
    public void setTrainSchedule(String trainSchedule) {
        this.trainSchedule = trainSchedule;
    }
    public String getPersonInCharge() {
        return personInCharge;
    }
    public void setPersonInCharge(String personInCharge) {
        this.personInCharge = personInCharge;
    }
    public String getCrossing_Status() {
        return crossing_status;
    }
    public void setCrossing_Status(String crossing_status) {
        this.crossing_status = crossing_status;
    }
}

```

4. RailwayCrossingDAO

```
package com.Project;
```

```

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

```



```

public class RailwayCrossingDAO {

    private Connection connection;
    public RailwayCrossingDAO()
    {
        // Initialize the database connection
        connection = DatabaseConnection.getConnection();
    }
    // Fetch all values from the database
    public List<RailwayCrossing> getAllCrossings() {
        List<RailwayCrossing> crossings = new ArrayList<>();
        try {
            String query = "SELECT * FROM railway_crossings";
            PreparedStatement statement = connection.prepareStatement(query);
            ResultSet resultSet = statement.executeQuery();
            while (resultSet.next()) {
                RailwayCrossing crossing = new RailwayCrossing();
                crossing.setName(resultSet.getString("name"));
                crossing.setCrossing_Status(resultSet.getString("crossing_status"));
                crossing.setPersonInCharge(resultSet.getString("person_in_charge"));
                crossing.setTrainSchedule(resultSet.getString("train_schedule"));
                crossing.setLandmark(resultSet.getString("landmark"));
                crossing.setAddress(resultSet.getString("address"));
                crossings.add(crossing);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return crossings;
    }

    // Add a railway crossing to the database
    public void addCrossing(RailwayCrossing crossing) {
        try {
            String query = "INSERT INTO railway_crossings (name, crossing_status,
            person_in_charge, train_schedules, landmark, address) VALUES (?, ?, ?, ?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(query);
            statement.setString(1, crossing.getName());
            statement.setString(2, crossing.getCrossing_Status());
            statement.setString(3, crossing.getPersonInCharge());
            statement.setString(4, crossing.getTrainSchedule());
            statement.setString(5, crossing.getLandmark());
            statement.setString(6, crossing.getAddress());
        }
    }
}

```

```

        statement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void updateCrossing(RailwayCrossing crossing) {
    try {
        String query = "UPDATE railway_crossings SET crossing_status=?,
person_in_charge=?, train_schedules=?, landmark=?, address=? WHERE name=?";
        PreparedStatement statement = connection.prepareStatement(query);
        statement.setString(1, crossing.getName());
        statement.setString(2, crossing.getCrossing_Status());
        statement.setString(3, crossing.getPersonInCharge());
        statement.setString(4, crossing.getTrainSchedule());
        statement.setString(5, crossing.getLandmark());
        statement.setString(6, crossing.getAddress());
        statement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// Delete a railway crossing from the database
public void deleteCrossing(String crossingName) {
    try {
        String query = "DELETE FROM railway_crossings WHERE name = ?";
        PreparedStatement statement = connection.prepareStatement(query);
        statement.setString(1, crossingName);
        statement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// Get favorite railway crossings from the database
public List<RailwayCrossing> getFavoriteCrossings() {
    List<RailwayCrossing> favoriteCrossings = new ArrayList<>();
    try {
        String query = "SELECT * FROM railway_crossings rc JOIN favorite_crossings fc
ON rc.name = fc.railway_crossing_name";
        PreparedStatement statement = connection.prepareStatement(query);
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            RailwayCrossing crossing = new RailwayCrossing();

```

```

        crossing.setName(resultSet.getString("name"));
        crossing.setCrossing_Status(resultSet.getString("crossing_status"));
        crossing.setPersonInCharge(resultSet.getString("person_in_charge"));
        crossing.setTrainSchedule(resultSet.getString("train_schedules"));
        crossing.setLandmark(resultSet.getString("landmark"));
        crossing.setAddress(resultSet.getString("address"));
        favoriteCrossings.add(crossing);
    }
} catch (SQLException e) {
    e.printStackTrace();
}
return favoriteCrossings;
}

// Add a railway crossing to favorites
public void addToFavorites(String crossingName) {
    try {
        String query = "INSERT INTO favorite_crossings (railway_crossing_name)
VALUES (?)";
        PreparedStatement statement = connection.prepareStatement(query);
        statement.setString(1, crossingName);
        statement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// Remove a railway crossing from favorites
public void removeFromFavorites(String crossingName) {
    try {
        String query = "DELETE FROM favorite_crossings WHERE
railway_crossing_name = ?";
        PreparedStatement statement = connection.prepareStatement(query);
        statement.setString(1, crossingName);
        statement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public RailwayCrossing getCrossingByName(String crossingName) {
    RailwayCrossing crossing = null;
    try {
        String query = "SELECT * FROM railway_crossings WHERE name = ?";
        PreparedStatement statement = connection.prepareStatement(query);
        statement.setString(1, crossingName);
    }
}

```

```

        ResultSet resultSet = statement.executeQuery();
        if (resultSet.next()) {
            crossing = new RailwayCrossing();
            crossing.setName(resultSet.getString("name"));
            crossing.setCrossing_Status(resultSet.getString("crossing_status"));
            crossing.setPersonInCharge(resultSet.getString("person_in_charge"));
            crossing.setTrainSchedule(resultSet.getString("train_schedule"));
            crossing.setLandmark(resultSet.getString("landmark"));
            crossing.setAddress(resultSet.getString("address"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return crossing;
}
}

```

Database Connection File

```

package com.Project;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class DatabaseConnection {
    private static final String URL ="jdbc:mysql://localhost:3306/railway";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "Deepu12*";
    public static Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
        return connection;
    }
}

```

JSP Files

```

1. addRailwayCrossing
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>

```

```
<head>
<meta charset="ISO-8859-1">
<title>Add Railway Crossing</title>
<style>
body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
text-align: center;
background-color: #f2f2f2;
}
.panel {
width: 500px;
padding: 40px;
background-color: #ffffff;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
.panel label {
display: block;
margin-bottom: 10px;
text-align: left;
}
.panel input[type="text"], .panel select {
width: 100%;
padding: 8px;
border: 1px solid #ccc;
border-radius: 4px;
}
.panel button[type="submit"] {
width: 100%;
padding: 10px;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
}
.panel button[type="submit"]:hover {
background-color: #45a049;
}
</style>
</head>
```

```

<body>
<div class="panel">
<h2>Railway Crossings</h2>
<h4>Add Railway Crossing Information</h4>
<form action="AddCrossingServlet" method="post">
<label for="name">Enter Name</label> <input type="text"
id="name"
name="name" required><br> <label
for="address">Address</label>
<input type="text" id="address" name="address"
required><br>
<label for="landmark">Landmark</label> <input type="text"
id="landmark" name="landmark" required><br> <label
for="trainSchedule">Train Schedules</label> <input
type="text"
id="trainSchedule" name="trainSchedule"
required><br>
<label for="personInCharge">Person in Charge</label>
<input
type="text" id="personInCharge"
name="personInCharge" required><br>
<label for="crossing_status">Crossing Status</label> <select
id="crossing_status"
name="crossing_status">
<option value="Open">Open</option>
<option value="Closed">Closed</option>
</select><br>
<button type="submit">Submit</button>
</form>
</div>
</body>
</html>

```

2. adminHome

```

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

<%@ page import="com.Project.RailwayCrossing"%>
<%@ page import="com.Project.RailwayCrossingDAO"%>
<%@ page import="java.util.List"%>
<%

// Create an instance of RailwayCrossingDAO
RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();

```

```

// Retrieve all crossings from the DAO
List<RailwayCrossing> crossings = crossingDAO.getAllCrossings();
// Get the name from the search request parameter
String searchName = request.getParameter("searchName");
// Check if the searchName parameter is provided
if (searchName != null && !searchName.isEmpty()) {
    // Retrieve the crossing by name from the DAO
    RailwayCrossing searchedCrossing = crossingDAO.getCrossingByName(searchName);
    // Set the crossing as an attribute to be displayed in the table
    request.setAttribute("crossing", searchedCrossing);
}
%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Admin Home Page</title>
    <style>
        /* Button Styles */
        .button-container {
            display: flex;
            justify-content: flex-start;
            margin-bottom: 10px;
            margin-left: 5px;
        }
        .button-container form {
            margin-right: 100px;
        }
        .button-container input[type="text"] {
            margin-right: 5px;
            padding: 5px;

```

```
border-radius: 4px;
border: 1px solid #ccc;
}
.button-container button[type="submit"] {
padding: 10px 20px;
border-radius: 4px;
border: none;
background-color: #B6B6B6;
font-weight: bold;
color: black;
cursor: pointer;
}
.button-container button[type="submit"]:hover {
background-color: #45a049;
}
.update-button, .delete-button {
display: inline-block;
vertical-align: middle;
margin-right: 10px;
padding: 7px;
width: 80px;
height: 30px;
}
/* Table Styles */
table {
border-collapse: collapse;
width: 100%;
}
th, td {
text-align: left;
padding: 8px;
```



```
border-bottom: 1px solid #ddd;
}
th {
background-color: #f2f2f2;
}
/* Responsive Styles */
@media only screen and (max-width: 768px) {
.button-container form {
margin-right: 5px;
}
.update-button, .delete-button {
float: none;
margin-right: 5px;
margin-bottom: 5px;
}
}
</style>
</head>
<body>
<h1>Admin Home Page</h1>
<h3>Railway Crossing[<%=crossings.size()%>]</h3>
<!-- Buttons -->
<!-- Table -->
<table>
<thead>
<tr>
<th>Sr.No</th>
<th>Name</th>
<th>Address</th>
<th>Landmark</th>
<th>Train Schedule</th>
```

```

        <th>Person In-Charge</th>

        <th>Status</th>

        <th>Action</th>
    </tr>
</thead>
<tbody>
    <%
    if (request.getAttribute("crossing") != null) { // Check if a specific crossing is found
        RailwayCrossing crossing = (RailwayCrossing) request.getAttribute("crossing");
    %>
        <tr>
            <td>1</td>
            <td><%=crossing.getName()%></td>
            <td><%=crossing.getAddress()%></td>
            <td><%=crossing.getLandmark()%></td>
            <td><%=crossing.getTrainSchedule()%></td>
            <td><%=crossing.getPersonInCharge()%></td>
            <td><%=crossing.getCrossing_Status()%></td>
            <td></td>
        </tr>
    <%
    } else { // Display all crossings
        int count = 1;
        for (RailwayCrossing anotherCrossing : crossings) {
    %>
        <tr>
            <td><%=count%></td>
            <td><%=anotherCrossing.getName()%></td>
            <td><%=anotherCrossing.getAddress()%></td>
            <td><%=anotherCrossing.getLandmark()%></td>
            <td><%=anotherCrossing.getTrainSchedule()%></td>

```

```

        <td><%=anotherCrossing.getPersonInCharge()%></td>
        <td><%=anotherCrossing.getCrossing_Status()%></td>
        <td>
            <!-- Action buttons for update and delete -->
        </td>
    </tr>
<%
    count++;
}
}
%>
</tbody>
</table>
</body>
</html>

```

3. adminLogin

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Admin Login</title>
<style>
body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
text-align: center;
background-color: #f2f2f2;

```

```
}  
.panel {  
width: 400px;  
padding: 40px;  
background-color: #ffffff;  
border-radius: 5px;  
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
}  
.panel label {  
display: block;  
margin-bottom: 10px;  
text-align: left;  
}  
.panel input[type="text"], .panel input[type="password"] {  
width: 100%;  
padding: 10px;  
border: 1px solid #ccc;  
border-radius: 4px;  
}  
.panel button[type="submit"] {  
width: 100%;  
padding: 10px;  
background-color: #4CAF50;  
color: white;  
border: none;  
border-radius: 4px;  
cursor: pointer;  
}  
.panel button[type="submit"]:hover {  
background-color: #45a049;  
}
```

```

.panel .signup-link {
display: block;
margin-top: 20px;
}
</style>
</head>
<body>
<div class="panel">
<h2>Railway Crossing</h2>
<h4>Admin Login</h4>
<form action="AdminLoginServlet" method="post">
<label for="email">Email:</label> <input type="text"
id="email"
name="email" required><br> <label
for="password">Password:</label>
<input type="password" id="password" name="password"
required><br>
<button type="submit">Login</button>
</form>
</div>
</body>
</html>

```

4.HomePage

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Railway Crossing Status</title>

```

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap
.min.css"
rel="stylesheet"
integrity="sha384-
9ndCyUalbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"crossorigin
="anonymous">
<style>
body {
background-image:
url("https://source.unsplash.com/1000x1000/? indian railway,railway crossing");
background-repeat: no-repeat;
background-size: cover;
}
.navbar-brand span.quiz-text {
color: red;
}
.navbar-nav.custom-ml-auto {
margin-left: auto;
}
.navbar-nav.custom-ml-auto .nav-item {
margin-left: 10px;
}
</style>
</head>
<body>
<nav
class="navbar navbar-expand-lg bg-body-tertiary navbar-dark
border-bottom border-bottom-dark"
data-bs-theme="dark">
<div class="container-fluid">
<a class="navbar-brand" href="#">Railway Crossing
```

Status

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

```
<div class="collapse navbar-collapse"
id="navbarSupportedContent">
<ul class="navbar-nav custom-ml-auto">
<!-- Move the "Home", "Quiz", "Login/Signup"
links to the right using custom CSS -->
<li class="nav-item"><a class="nav-link
active"
aria-current="page"
href="#">Home</a></li>
<li class="nav-item"><a class="nav-link"
href="adminLogin.jsp">Admin-Login/Signup</a></li>
<li class="nav-item"><a class="nav-link"
href="userLogin.jsp">User-Login/Signup</a>
</li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
<script
```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.b
undle.min.js"
```

```
integrity="sha384-
```

```
geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68Sly3Te4Bkz"
```

```
crossorigin="anonymous"></script>
```

```
</body>
</html>
5. searchCrossing
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Search Crossing</title>
</head>
<body>
<h1>Search Crossing</h1>
<form action="searchCrossing" method="post">
<label for="searchName">Search Name:</label> <input type="text"
id="searchName" name="searchName" required> <input
type="submit"
value="Search">
</form>
<c:if test="${not empty crossing}">
<h2>Search Result</h2>
<table>
<tr>
<th>Name</th>
<th>Crossing Status</th>
<th>Person in charge</th>
<th>Train Schedules</th>
<th>Landmark</th>
<th>Address</th>
</tr>
```



```

<tr>
<td>${crossing.name}</td>
<td>${crossing.crossing_status}</td>
<td>${crossing.personInCharge}</td>
<td>${crossing.trainSchedules}</td>
<td>${crossing.landmark}</td>
<td>${crossing.address}</td>
</tr>
</table>
</c:if>
<a href="adminHome.jsp">Back to Admin Homepage</a>
</body>
</html>

```

6. updateRailwayCrossingForm

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Update Railway Crossing</title>
<style>
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    text-align: center;
    background-color: #f2f2f2;
}
.panel {
    width: 500px;

```

```
padding: 40px;
background-color: #ffffff;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
.panel label {
display: block;
margin-bottom: 10px;
text-align: left;
}
.panel input[type="text"], .panel select {
width: 100%;
padding: 8px;
border: 1px solid #ccc;
border-radius: 4px;
}
.panel button[type="submit"] {
width: 100%;
padding: 10px;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
}
.panel button[type="submit"]:hover {
background-color: #45a049;
}
</style>
</head>
<body>
```

```

<div class="panel">
    <h2>Railway Crossings</h2>
    <h4>Update Railway Crossing Information</h4>
    <form action="updateCrossing" method="post">
        <input type="hidden" name="name" value="${crossing.name}">
        <label for="address">Address</label>
        <input type="text" id="address" name="address" value="${crossing.address}"
required><br>
        <label for="landmark">Landmark</label>
        <input type="text" id="landmark" name="landmark" value="${crossing.landmark}"
required><br>
        <label for="trainSchedules">Train Schedules</label>
        <input type="text" id="trainSchedules" name="trainSchedules"
value="${crossing.trainSchedule}" required><br>
        <label for="personInCharge">Person in Charge</label>
        <input type="text" id="personInCharge" name="personInCharge"
value="${crossing.personInCharge}" required><br>
        <label for="status">Crossing Status</label>
        <select id="status" name="status">
            <option value="Open" ${crossing.status == 'Open' ? 'selected' : ''}>Open</option>
            <option value="Closed" ${crossing.status == 'Closed' ? 'selected' : ''}>Closed</option>
        </select><br>
        <button type="submit">Submit</button>
    </form>
</div>
</body>
</html>

```

7.userHome

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-
8"%>
<%@ page import="com.Project.RailwayCrossing"%>
<%@ page import="com.Project.RailwayCrossingDAO"%>
<%@ page import="java.util.List"%>

```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Railway Crossings - User Home Page</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 20px;
}
.container {
border: 1px solid #ccc;
border-radius: 10px;
padding: 20px;
margin-bottom: 20px;
margin-top: 20px;
}
.container h3 {
margin-top: 0;
}
.container p {
margin: 0;
}
.crossing {
margin-bottom: 10px;
}
.crossing button {
margin-left: 10px;
}
.status {
```

```
display: inline-block;
padding: 5px;
font-weight: bold;
}
.status-open {
color: white;
border-radius: 30%;
background-color: green;
}
.status-closed {
color: white;
background-color: red;
}
</style>
</head>
<body>
<h1 style="font-weight: bold;">User Home Page</h1>
<!-- Display Search Crossings button -->
<button onclick="location.href='HomePage.jsp'"
style="border-radius: 10px; font-weight: bold;">Home</button>
<!-- Display All button -->
<button onclick="location.href='userHome.jsp'"
style="border-radius: 10px; margin-left: 50px; font-weight:
bold;">All
Crossings</button>
<!-- Display Favorite Crossings button -->
<button onclick="location.href='userHomeFavoriteCrossing.jsp'"
style="border-radius: 10px; margin-left: 50px; font-weight:
bold;">Favorite
Crossings</button>
<!-- Display Search Crossings button -->
```

```

<button onclick="location.href='userSearchCrossing.jsp'"
style="border-radius: 10px; margin-left: 50px; font-weight:
bold;">Search
Crossings</button>

<!-- Logout button -->

<button onclick="location.href='userLogin.jsp'"
style="border-radius: 10px; margin-left: 800px; font-weight:
bold;">Logout</button>

<%-- Create an instance of RailwayCrossingDAO --%>
<%
RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
List<RailwayCrossing> allCrossings = crossingDAO.getAllCrossings();
for (RailwayCrossing crossing : allCrossings) {
String status = crossing.getStatus();
String statusClass = status.equalsIgnoreCase("Open") ? "statusopen" : "status-closed";
%>
<div class="container">
<h3><%=crossing.getName()%></h3>
<p>
Status: <strong><span class="status
<%=statusClass%>"><%=crossing.getStatus()%></span></strong>
</p>
<p>
Person in Charge:
<strong><%=crossing.getPersonInCharge()%></strong>
</p>
<p>
Train Schedules:
<strong><%=crossing.getTrainSchedule()%></strong>
</p>
<p>

```

Landmark: <%=crossing.getLandmark()%>

</p>

<p>

Address: <%=crossing.getAddress()%>

</p>

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

<input type="hidden" name="crossingId"

value="<%=crossing.getId()%>">

<button type="submit"

style="border-radius: 10px; font-weight: bold;

margin-top: 10px; margin-left: 0px;">ADD

TO FAVORITES</button>

</form>

</div>

<%

}

%>

</body>

</html>

8.userHomeFavoritePage

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

<%@ page import="com.Project.RailwayCrossing"%>

<%@ page import="com.Project.RailwayCrossingDAO"%>

<%@ page import="java.util.List"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Railway Crossings - Favorite Crossings</title>

<style>

```
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 20px;
}
.container {
border: 1px solid #ccc;
border-radius: 10px;
padding: 20px;
margin-bottom: 20px;
margin-top: 20px;
}
.container h3 {
margin-top: 0;
}
.container p {
margin: 0;
}
.status {
display: inline-block;
padding: 5px;
color: white;
font-weight: bold;
}
.status-open {
background-color: green;
}
.status-closed {
background-color: red;
}
</style>
```



```

</head>
<body>
<h2>User Home Page</h2>
<!-- Display All button -->
<button onclick="location.href='userHome.jsp'">All Crossings</button>
<!-- Favorite Crossings Container -->
<div class="favorites">
<%
RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
List<RailwayCrossing> favoriteCrossings =
crossingDAO.getFavoriteCrossings();
for (RailwayCrossing crossing : favoriteCrossings) {
String status = crossing.getStatus();
String statusClass = status.equalsIgnoreCase("Open") ?
"status-open" : "status-closed";
%>
<div class="container">
<h3><%=crossing.getName()%></h3>
<p>
Status: <strong><span class="status
<%=statusClass%>"><%=status%></span></strong>
</p>
<p>
Person in Charge:
<strong><%=crossing.getPersonInCharge()%></strong>
</p>
<p>
Train Schedules:
<strong><%=crossing.getTrainSchedule()%></strong>
</p>
<p>

```

Landmark:

```
<strong><%=crossing.getLandmark()%></strong>
```

```
</p>
```

```
<p>
```

Address:

```
<strong><%=crossing.getAddress()%></strong>
```

```
</p>
```

```
<form action="removeFromFavorites" method="post"
```

```
style="display: inline;">
```

```
<input type="hidden" name="crossingId"
```

```
value="<%=crossing.getId()%>">
```

```
<button type="submit"
```

```
style="border-radius: 10px; font-weight:
```

```
bold; margin-top: 10px;">REMOVE
```

```
FROM FAVORITES</button>
```

```
</form>
```

```
</div>
```

```
<%
```

```
}
```

```
%>
```

```
</div>
```

```
</body>
```

```
</html>
```

9.userlogin

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

```
pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>User Login</title>
```

```
<style>
body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
text-align: center;
background-color: #f2f2f2;
}
.panel {
width: 400px;
padding: 40px;
background-color: #ffffff;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
.panel label {
display: block;
margin-bottom: 10px;
text-align: left;
}
.panel input[type="text"], .panel input[type="password"] {
width: 100%;
padding: 10px;
border: 1px solid #ccc;
border-radius: 4px;
}
.panel button[type="submit"] {
width: 100%;
padding: 10px;
background-color: #4CAF50;
```

```
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
}
.panel button[type="submit"]:hover {
background-color: #45a049;
}
.panel .signup-link {
display: block;
margin-top: 20px;
}
</style>
</head>
<body>
<div class="panel">
<h2>Railway Crossing</h2>
<h4>User Login</h4>
<form action="login" method="post">
<label for="email">Email:</label> <input type="text" id="email"
name="email" required><br> <label
for="password">Password:</label>
<input type="password" id="password" name="password"
required><br>
<button type="submit">Login</button>
</form>
<div class="signup-link">
<p>
Don't have an account? <a
href="userRegister.jsp">Create New
Account</a>
```

```
</p>
</div>
</div>
</body>
</html>
10.userRegister
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>User Registration</title>
<style>
body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
text-align: center;
background-color: #f2f2f2;
}
.panel {
width: 500px;
padding: 40px;
background-color: #ffffff;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
.panel label {
display: block;
```

```
margin-bottom: 10px;
text-align: left;
}
.panel input[type="text"], .panel input[type="password"] {
width: 100%;
padding: 10px;
border: 1px solid #ccc;
border-radius: 4px;
}
.panel button[type="submit"] {
width: 100%;
padding: 10px;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
}
.panel button[type="submit"]:hover {
background-color: #45a049;
}
.panel .signup-link {
display: block;
margin-top: 20px;
}
</style>
</head>
<body>
<div class="panel">
<h2>Railway Crossing</h2>
<h4>User Register</h4>
```

```

<form action="register" method="post">
<label for="name">Name:</label> <input type="text"
id="name"
name="name" required><br> <label
for="email">Email:</label>
<input type="text" id="email" name="email" required><br>
<label for="password">Password:</label> <input
type="password"
id="password" name="password" required><br>
<button type="submit">Register</button>
</form>
<div class="signup-link">
<p>
Already have an account? <a
href="userLogin.jsp">Sign in</a>
</p>
</div>
</div>
</body>
</html>

```

11.userSearchCrossing

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="com.Project.RailwayCrossing"%>
<%@ page import="com.Project.RailwayCrossingDAO"%>
<%@ page import="java.util.List"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Railway Crossings - Search Crossings</title>

```

```
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 20px;
}
.container {
border: 1px solid #ccc;
border-radius: 10px;
padding: 20px;
margin-bottom: 20px;
margin-top: 20px;
}
.container h3 {
margin-top: 0;
}
.container p {
margin: 0;
}
.search-container {
margin-top: 60px;
margin-bottom: 20px;
}
.search-container input[type="text"] {
padding: 10px;
width: 200px;
}
.logout-button {
position: absolute;
padding: 10px;
top: 50px;
```



```
right: 30px;
border-radius: 10px;
font-weight: bold;
top: 50px;
}
.allCrossing-button {
position: absolute;
padding: 5px;
top: 75px;
left: 10px;
}
.favouriteCrossing-button {
position: absolute;
padding: 5px;
top: 75px;
left: 130px;
}
.status {
display: inline-block;
padding: 5px;
font-weight: bold;
}
.status-open {
color: white;
border-radius: 30%;
background-color: green;
}
.status-closed {
color: white;
background-color: red;
}
```

```

.crossing {
margin-bottom: 10px;
}
.crossing button {
margin-left: 10px;
}
</style>
<script>
function searchCrossings() {
var input = document.getElementById("searchInput");
var filter = input.value.toUpperCase();
var crossings = document.getElementsByClassName("crossing");
for (var i = 0; i < crossings.length; i++) {
var crossing = crossings[i];
var name =
crossing.getElementsByTagName("h3")[0].innerText;
if (name.toUpperCase().indexOf(filter) > -1) {
crossing.style.display = "";
} else {
crossing.style.display = "none";
}
}
}
</script>
</head>
<body>
<h2>Search Crossings</h2>
<!-- Display All button -->
<button onclick="location.href='userHome.jsp'"
class="allCrossing-button"
style="border-radius: 10px; font-weight: bold;">All

```

```

Crossings</button>
<!-- Display Favorite Crossings button -->
<button onclick="location.href='userHomeFavoriteCrossing.jsp'"
class="favouriteCrossing-button"
style="border-radius: 10px; font-weight: bold;">Favorite
Crossings</button>
<!-- Search input -->
<div class="search-container">
<input type="text" id="searchInput" placeholder="Search by
name"
onkeyup="searchCrossings()">
<!-- Logout button -->
<button onclick="location.href='userLogin.jsp'" class="logoutbutton"
style="border-radius: 10px; font-weight:
bold;">Logout</button>
</div>
<!-- Search Results Container -->
<div class="search-results">
<%
RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
List<RailwayCrossing> allCrossings =
crossingDAO.getAllCrossings();
for (RailwayCrossing crossing : allCrossings) {
String status = crossing.getStatus();
String statusClass = status.equalsIgnoreCase("Open") ?
"status-open" : "status-closed";
%>
<div class="container crossing">
<h3><%=crossing.getName()%></h3>
<p>
Status: <strong><span class="status

```

<%=statusClass%>"><%=crossing.getStatus()%>

</p>

<p>

Person in Charge:

<%=crossing.getPersonInCharge()%>

</p>

<p>

Train Schedules:

<%=crossing.getTrainSchedule()%>

</p>

<p>

Landmark:

<%=crossing.getLandmark()%>

</p>

<p>

Address:

<%=crossing.getAddress()%>

</p>

<form action="addToFavorites" method="post"

style="display: inline;">

<input type="hidden" name="crossingId"

value="<%=crossing.getId()%>">

<button type="submit"

style="border-radius: 10px; font-weight:

bold; margin-top: 10px; margin-left: 0px;">ADD

TO FAVORITES</button>

</form>

</div>

<%

