

I had also pushed the Code to git below is the URL link

<https://github.com/R-NandaKumar/Java-Fsd>

## PHASE-02

# Railway Crossing Status (SOURCE CODE)

```
1 package com.simpli;
2
3 import java.io.IOException;
4
5 @WebServlet(name = "AddCrossingServlet", urlPatterns = { "/addCrossing" })
6 public class AddCrossingServlet extends HttpServlet {
7     protected void doPost(HttpServletRequest request, HttpServletResponse response)
8         throws ServletException, IOException {
9         // Retrieve the form data from the request
10         String name = request.getParameter("name");
11         String address = request.getParameter("address");
12         String landmark = request.getParameter("landmark");
13         String trainSchedules = request.getParameter("trainSchedules");
14         String personInCharge = request.getParameter("personInCharge");
15         String status = request.getParameter("status");
16
17         // Create a new RailwayCrossing object with the form data
18         RailwayCrossing crossing = new RailwayCrossing();
19         crossing.setName(name);
20         crossing.setAddress(address);
21         crossing.setLandmark(landmark);
22         crossing.setTrainSchedule(trainSchedules);
23         crossing.setPersonInCharge(personInCharge);
24         crossing.setStatus(status);
25
26         // Save the new crossing to the database
27         RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
28         crossingDAO.addCrossing(crossing);
29
30         // Redirect to the admin homepage after adding the crossing
31         response.sendRedirect("adminHome.jsp");
32     }
33 }
```

```
1 package com.simpli;
2
3 import javax.servlet.ServletException;
4
5 public class AddToFavoritesServlet extends HttpServlet {
6     protected void doPost(HttpServletRequest request, HttpServletResponse response)
7         throws ServletException, IOException {
8         String crossingId = request.getParameter("crossingId");
9         if (crossingId != null && !crossingId.isEmpty()) {
10             int railwayCrossingId = Integer.parseInt(crossingId);
11             // Perform the database operation to add the crossing to favorites
12             RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
13             crossingDAO.addToFavorites(railwayCrossingId); // Implement this method in RailwayCrossingDAO
14         }
15         // Redirect back to the userHome.jsp page
16         response.sendRedirect("userHome.jsp");
17     }
18 }
```

```
1 package com.simpli;
2
3 import java.io.IOException;
4
5 @WebServlet("/adminHomePage")
6 public class AdminHomePageServlet extends HttpServlet {
7     private RailwayCrossingDAO crossingDAO;
8
9     public void init() {
10         crossingDAO = new RailwayCrossingDAO();
11     }
12
13     protected void doGet(HttpServletRequest request, HttpServletResponse response)
14         throws ServletException, IOException {
15         // Fetch all railway crossing details from the database
16         List<RailwayCrossing> crossings = crossingDAO.getAllCrossings();
17
18         // Pass the crossing details to the JSP page
19         request.setAttribute("crossings", crossings);
20         request.getRequestDispatcher("adminHome.jsp").forward(request, response);
21     }
22 }
```

```

AddCrossingServlet.java AddToFavoritesServlet.java AdminHomePageServlet.java *AdminLoginServlet.java × DatabaseConnection.java LoginServlet.java
1 package com.simpli;
2 import java.io.IOException;
3
4 @WebServlet("/adminLogin")
5 public class AdminLoginServlet extends HttpServlet {
6     private static final long serialVersionUID = 1L;
7
8     private static final String ADMIN_ID = "nanda@gmail.com";
9     private static final String ADMIN_PASSWORD = "nanda";
10
11     protected void doPost(HttpServletRequest request, HttpServletResponse response)
12         throws ServletException, IOException {
13         String email = request.getParameter("email");
14         String password = request.getParameter("password");
15
16         if (email != null && email.equals(ADMIN_ID) && password != null && password.equals(ADMIN_PASSWORD)) {
17             response.sendRedirect("adminHome.jsp");
18         } else {
19             request.getRequestDispatcher("adminLogin.jsp").forward(request, response);
20         }
21     }
22 }
23
24

```

```

AddCrossingServlet.java AddToFavoritesServlet.java AdminHomePageServlet.java *AdminLoginServlet.java DatabaseConnection.java × LoginServlet.java
1 package com.simpli;
2
3 import java.sql.Connection;
4
5 public class DatabaseConnection {
6     private static final String URL = "jdbc:mysql://localhost:3306/railway_crossing_status";
7     private static final String USERNAME = "root";
8     private static final String PASSWORD = "root123";
9
10     public static Connection getConnection() {
11         Connection connection = null;
12         try {
13             Class.forName("com.mysql.cj.jdbc.Driver");
14             connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
15         } catch (ClassNotFoundException | SQLException e) {
16             e.printStackTrace();
17         }
18         return connection;
19     }
20 }
21
22

```

```

AddCrossingServlet.java AddToFavoritesServlet.java AdminHomePageServlet.java *AdminLoginServlet.java DatabaseConnection.java *LoginServlet.java ×
1 package com.simpli;
2 import java.io.IOException;
3
4 @WebServlet("/login")
5 public class LoginServlet extends HttpServlet {
6     protected void doPost(HttpServletRequest request, HttpServletResponse response)
7         throws ServletException, IOException {
8         // Retrieve the form data
9         String email = request.getParameter("email");
10        String password = request.getParameter("password");
11
12        // Check if the provided credentials are valid
13        UserDAO userDAO = new UserDAO();
14        User user = userDAO.getUserByEmail(email);
15
16        if (user != null && user.getPassword().equals(password)) {
17            // Authentication successful
18            // You can store user details in session for further use
19            HttpSession session = request.getSession();
20            session.setAttribute("user", user);
21
22            // Redirect to a success page or perform any other necessary actions
23            response.sendRedirect("userHome.jsp");
24        } else {
25            // Authentication failed
26            // Redirect to an error page or display an error message
27            response.sendRedirect("userRegister.jsp?error=1");
28        }
29    }
30 }
31
32

```

## RailwayCrossing.java

```
package com.simpli;

public class RailwayCrossing {

    private int id;

    private String name;

    private String address;

    private String landmark;

    private String trainSchedule;

    private String personInCharge;

    private String status;

    public RailwayCrossing() {

        super();
    }

    public RailwayCrossing(int id, String name, String address, String landmark, String
trainSchedule,

String personInCharge, String status) {

        super();

        this.id = id;

        this.name = name;

        this.address = address;

        this.landmark = landmark;

        this.trainSchedule = trainSchedule;

        this.personInCharge = personInCharge;

        this.status = status;}

    public int getId() {

        return id;}
}
```

```
public void setId(int id) {  
  
    this.id = id;}  
  
public String getName() {return name;}  
  
public void setName(String name) {  
  
    this.name = name;}  
  
public String 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 getStatus() {  
  
    return status;}  
  
public void setStatus(String status) {  
  
    this.status = status;}}
```

## RailwayCrossingDAO.java

```
package com.simpli;

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_crossing";
            PreparedStatement statement = connection.prepareStatement(query);
            ResultSet resultSet = statement.executeQuery();

            while (resultSet.next()) {

                RailwayCrossing crossing = new RailwayCrossing();
                crossing.setId(resultSet.getInt("id"));
                crossing.setName(resultSet.getString("name"));
                crossing.setAddress(resultSet.getString("address"));
                crossing.setLandmark(resultSet.getString("landmark"));
            }

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

        return crossings;
    }
}
```

```

        crossing.setTrainSchedule(resultSet.getString("train_schedule"));

        crossing.setPersonInCharge(resultSet.getString("person_in_charge"));
        crossing.setStatus(resultSet.getString("status"));
        crossings.add(crossing);
    }
} catch (SQLException e) {
    e.printStackTrace();
}
return crossings;
}

// Search values by id and display it from the database
public RailwayCrossing getCrossingById(int crossingId) {
    RailwayCrossing crossing = null;
    try {
        String query = "SELECT * FROM railway_crossing WHERE id = ?";
        PreparedStatement statement = connection.prepareStatement(query);
        statement.setInt(1, crossingId);
        ResultSet resultSet = statement.executeQuery();
        if (resultSet.next()) {
            crossing = new RailwayCrossing();
            crossing.setId(resultSet.getInt("id"));
            crossing.setName(resultSet.getString("name"));
            crossing.setAddress(resultSet.getString("address"));
            crossing.setLandmark(resultSet.getString("landmark"));

            crossing.setTrainSchedule(resultSet.getString("train_schedule"));

            crossing.setPersonInCharge(resultSet.getString("person_in_charge"));
            crossing.setStatus(resultSet.getString("status"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return crossing;
}

```

```

    }

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

    return crossing;
}

// update the values in the database
public void updateCrossing(RailwayCrossing crossing) {
    try {
        String query = "UPDATE railway_crossing SET name=?, address=?,
landmark=?, train_schedule=?, person_in_charge=?, status=? WHERE id=?";

        PreparedStatement statement = connection.prepareStatement(query);
        statement.setString(1, crossing.getName());
        statement.setString(2, crossing.getAddress());
        statement.setString(3, crossing.getLandmark());
        statement.setString(4, crossing.getTrainSchedule());
        statement.setString(5, crossing.getPersonInCharge());
        statement.setString(6, crossing.getStatus());
        statement.setInt(7, crossing.getId());

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

// Delete Railway Crossing
public void deleteCrossing(int crossingId) {
    try {
        connection.setAutoCommit(false); // Start transaction

```

```

// Delete associated favorite crossings

String deleteFavoriteCrossingsQuery = "DELETE FROM
favorite_crossing WHERE railway_crossing_id = ?";

try (PreparedStatement deleteFavoriteCrossingsStatement = connection
    .prepareStatement(deleteFavoriteCrossingsQuery)) {
    deleteFavoriteCrossingsStatement.setInt(1, crossingId);
    deleteFavoriteCrossingsStatement.executeUpdate();
}

// Delete the railway crossing

String deleteRailwayCrossingQuery = "DELETE FROM railway_crossing
WHERE id = ?";

try (PreparedStatement deleteRailwayCrossingStatement = connection
    .prepareStatement(deleteRailwayCrossingQuery)) {
    deleteRailwayCrossingStatement.setInt(1, crossingId);
    deleteRailwayCrossingStatement.executeUpdate();
}

connection.commit(); // Commit the transaction
connection.setAutoCommit(true); // Reset auto-commit to true
} catch (SQLException e) {
    try {
        connection.rollback(); // Rollback the transaction if an error
occurs

    } catch (SQLException rollbackException) {
        rollbackException.printStackTrace();
    }
    e.printStackTrace();
}
}

```



```

// Add values to the database

public void addCrossing(RailwayCrossing crossing) {
    try {
        String query = "INSERT INTO railway_crossing (name, address,
landmark, train_schedule, person_in_charge, status) VALUES (?, ?, ?, ?, ?, ?)";

        PreparedStatement statement = connection.prepareStatement(query);

        statement.setString(1, crossing.getName());
        statement.setString(2, crossing.getAddress());
        statement.setString(3, crossing.getLandmark());
        statement.setString(4, crossing.getTrainSchedule());
        statement.setString(5, crossing.getPersonInCharge());
        statement.setString(6, crossing.getStatus());

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

public List<RailwayCrossing> getFavoriteCrossings() {
    List<RailwayCrossing> favoriteCrossings = new ArrayList<>();

    try (Connection connection = DatabaseConnection.getConnection()) {
        String query = "SELECT rc.* FROM railway_crossing rc "
            + "JOIN favorite_crossing fc ON rc.id =
fc.railway_crossing_id";

        PreparedStatement statement = connection.prepareStatement(query);
        ResultSet resultSet = statement.executeQuery();

        while (resultSet.next()) {
            RailwayCrossing crossing = new RailwayCrossing();
            crossing.setId(resultSet.getInt("id"));

```

```

        crossing.setName(resultSet.getString("name"));
        crossing.setAddress(resultSet.getString("address"));
        crossing.setLandmark(resultSet.getString("landmark"));

        crossing.setTrainSchedule(resultSet.getString("train_schedule"));

        crossing.setPersonInCharge(resultSet.getString("person_in_charge"));
        crossing.setStatus(resultSet.getString("status"));

        favoriteCrossings.add(crossing);
    }
} catch (SQLException e) {
    e.printStackTrace();
}

return favoriteCrossings;
}

public void addToFavorites(int crossingId) {
    try (Connection connection = DatabaseConnection.getConnection()) {
        String sql = "INSERT INTO favorite_crossing (railway_crossing_id)
VALUES (?)";

        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setInt(1, crossingId);
        statement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
        // Handle the exception as needed
    }
}
}

```

```

        public void removeFromFavorites(int crossingId) {

            try (Connection connection = DatabaseConnection.getConnection()) {

                String sql = "DELETE FROM favorite_crossing WHERE
railway_crossing_id = ?";

                PreparedStatement statement = connection.prepareStatement(sql);

                statement.setInt(1, crossingId);

                statement.executeUpdate();

            } catch (SQLException e) {

                e.printStackTrace();

                // Handle the exception as needed

            }

        }

    }
}

```

```

1 package com.simpli;
2 import java.io.IOException;
3
4 /**
5  * Servlet implementation class RegisterServlet
6  */
7 @WebServlet("/register")
8 public class RegisterServlet extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11     protected void doPost(HttpServletRequest request, HttpServletResponse response)
12         throws ServletException, IOException {
13         // Retrieve the form data
14         String name = request.getParameter("name");
15         String email = request.getParameter("email");
16         String password = request.getParameter("password");
17
18         // Create a new User object
19         User user = new User(name, email, password);
20
21         // Store the user in the database
22         UserDao userDao = new UserDao();
23         userDao.registerUser(user);
24
25         // Redirect to a success page or perform any other necessary actions
26         response.sendRedirect("userLogin.jsp");
27     }
28 }

```

```

1 package com.simpli;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class RemoveFromFavoritesServlet
7  */
8
9 public class RemoveFromFavoritesServlet extends HttpServlet {
10     protected void doPost(HttpServletRequest request, HttpServletResponse response)
11         throws ServletException, IOException {
12         // TODO Auto-generated method stub
13         String crossingId = request.getParameter("crossingId");
14         if (crossingId != null && !crossingId.isEmpty()) {
15             int railwayCrossingId = Integer.parseInt(crossingId);
16             // Perform the database operation to remove the crossing from favorites
17             RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
18             crossingDAO.removeFromFavorites(railwayCrossingId); // Implement this method in RailwayCrossingDAO
19         }
20         // Redirect back to the userHome.jsp page
21         response.sendRedirect("userHome.jsp");
22     }
23 }
24
25
26
27
28

```

```

1 package com.simpli;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class SearchCrossingServlet
7  */
8
9 @WebServlet(name = "SearchCrossingServlet", urlPatterns = { "/searchCrossing" })
10 public class SearchCrossingServlet extends HttpServlet {
11     protected void doPost(HttpServletRequest request, HttpServletResponse response)
12         throws ServletException, IOException {
13         // Retrieve the search ID from the request
14         int searchId = Integer.parseInt(request.getParameter("searchId"));
15
16         // Implement your logic to search for the crossing by ID and retrieve its
17         // details from the database
18         RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
19         RailwayCrossing crossing = crossingDAO.getCrossingById(searchId);
20
21         // Set the search result in the request attribute
22         request.setAttribute("crossing", crossing);
23
24         // Redirect to the admin homepage with the search results
25         request.getRequestDispatcher("adminHome.jsp").forward(request, response);
26     }
27 }
28
29

```

```

1 package com.simpli;
2 import java.io.IOException;
3 @WebServlet("/updateCrossing")
4 public class UpdateCrossingServlet extends HttpServlet {
5     protected void doPost(HttpServletRequest request, HttpServletResponse response)
6         throws ServletException, IOException {
7         try {
8             // Retrieve the crossing ID from the hidden field
9             String idParam = request.getParameter("crossingId");
10             if (idParam != null && !idParam.isEmpty()) {
11                 int crossingId = Integer.parseInt(idParam);
12                 // Retrieve the updated details for the crossing
13                 String name = request.getParameter("name");
14                 String address = request.getParameter("address");
15                 String landmark = request.getParameter("landmark");
16                 String trainSchedules = request.getParameter("trainSchedules");
17                 String personInCharge = request.getParameter("personInCharge");
18                 String status = request.getParameter("status");
19                 // Create a new RailwayCrossing object with the updated details
20                 RailwayCrossing updatedCrossing = new RailwayCrossing();
21                 updatedCrossing.setId(crossingId);
22                 updatedCrossing.setName(name);
23                 updatedCrossing.setAddress(address);
24                 updatedCrossing.setLandmark(landmark);
25                 updatedCrossing.setTrainSchedule(trainSchedules);
26                 updatedCrossing.setPersonInCharge(personInCharge);
27                 updatedCrossing.setStatus(status);
28                 // Perform the update operation
29                 RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
30                 crossingDAO.updateCrossing(updatedCrossing);
31                 // Redirect to the admin homepage after the update
32                 response.sendRedirect("adminHome.jsp");
33             } else {
34                 // Handle case where idParam is empty or null
35                 throw new ServletException("Crossing ID is missing.");
36             }
37         } catch (NumberFormatException e) {
38             // Handle case where idParam is not a valid integer
39             throw new ServletException("Invalid Crossing ID format.", e);
40         } catch (Exception e) {
41             // Handle other exceptions
42             throw new ServletException("An error occurred during the crossing update.", e);
43         }
44     }
45 }
46

```

```

1 package com.simpli;
2 public class User {
3     private String name;
4     private String email;
5     private String password;
6     public User() {
7         super();
8     }
9     public User(String name, String email, String password) {
10         this.name = name;
11         this.email = email;
12         this.password = password;
13     }
14     public String getName() {
15         return name;
16     }
17     public void setName(String name) {
18         this.name = name;
19     }
20     public String getEmail() {
21         return email;
22     }
23     public void setEmail(String email) {
24         this.email = email;
25     }
26     public String getPassword() {
27         return password;
28     }
29     public void setPassword(String password) {
30         this.password = password;
31     }
32 }

```

```

1 import java.sql.Connection;
2 public class UserDao {
3     private Connection connection;
4
5     public UserDao() {
6         // Initialize the database connection
7         connection = DatabaseConnection.getConnection();
8     }
9     public void registerUser(User user) {
10         try {
11             String query = "INSERT INTO user_signup (name, email, password) VALUES (?, ?, ?)";
12             PreparedStatement statement = connection.prepareStatement(query);
13             statement.setString(1, user.getName());
14             statement.setString(2, user.getEmail());
15             statement.setString(3, user.getPassword());
16             statement.executeUpdate();
17         } catch (SQLException e) {
18             e.printStackTrace();
19         }
20     }
21     public boolean loginUser(String email, String password) {
22         try {
23             String query = "SELECT * FROM user_signup WHERE email=? AND password=?";
24             PreparedStatement statement = connection.prepareStatement(query);
25             statement.setString(1, email);
26             statement.setString(2, password);
27             ResultSet resultSet = statement.executeQuery();
28             return resultSet.next();
29         } catch (SQLException e) {
30             e.printStackTrace();
31             return false;
32         }
33     }
34     public User getUserByEmail(String email) {
35         try {
36             String query = "SELECT * FROM user_signup WHERE email=?";
37             PreparedStatement statement = connection.prepareStatement(query);
38             statement.setString(1, email);
39             ResultSet resultSet = statement.executeQuery();
40             if (resultSet.next()) {
41                 User user = new User();
42                 user.setName(resultSet.getString("name"));
43                 user.setEmail(resultSet.getString("email"));
44                 user.setPassword(resultSet.getString("password"));
45                 return user;
46             }
47         } catch (SQLException e) {
48             e.printStackTrace();
49             return null;
50         }
51     }
52 }

```

# JSP FILES

```
addRailCross... × adminHome.jsp adminLogin.jsp HomePage.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Add Railway Crossing</title>
8 <style>
9   body {
10     display: flex;
11     justify-content: center;
12     align-items: center;
13     height: 100vh;
14     text-align: center;
15     background-color: #f2f2f2;
16   }
17
18   .panel {
19     width: 500px;
20     padding: 40px;
21     background-color: #ffffff;
22     border-radius: 5px;
23     box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
24   }
25
26   .panel label {
27     display: block;
28     margin-bottom: 10px;
29     text-align: left;
30   }
31
32   .panel input[type="text"], .panel select {
33     width: 100%;
34     padding: 8px;
35     border: 1px solid #ccc;
36     border-radius: 4px;
37   }
38
```

```
addRailCross... × adminHome.jsp adminLogin.jsp HomePage.jsp
33   width: 100%;
34   padding: 8px;
35   border: 1px solid #ccc;
36   border-radius: 4px;
37 }
38
39 .panel button[type="submit"] {
40   width: 100%;
41   padding: 10px;
42   background-color: #4CAF50;
43   color: white;
44   border: none;
45   border-radius: 4px;
46   cursor: pointer;
47 }
48
49 .panel button[type="submit"]:hover {
50   background-color: #45a049;
51 }
52 </style>
53 </head>
54 <body>
55   <div class="panel">
56     <h2>Railway Crossings</h2>
57     <h4>Add Railway Crossing Information</h4>
58     <form action="addCrossing" method="post">
59       <label for="name">Enter Name</label> <input type="text" id="name"
60         name="name" required><br> <label for="address">Address</label>
61       <input type="text" id="address" name="address" required><br>
62       <label for="landmark">Landmark</label> <input type="text"
63         id="landmark" name="landmark" required><br> <label
64         for="trainSchedules">Train Schedules</label> <input type="text"
65         id="trainSchedules" name="trainSchedules" required><br>
66       <label for="personInCharge">Person in Charge</label> <input
67         type="text" id="personInCharge" name="personInCharge" required><br>
68       <label for="status">Crossing Status</label> <select id="status"
69         name="status">
70         <option value="Open">Open</option>
71         <option value="Closed">Closed</option>
72       </select><br>
73       <button type="submit">Submit</button>
74     </form>
75   </div>
76 </body>
77 </html>
78
```

```

addRailCross... adminHome.jsp x adminLogin.jsp HomePage.jsp searchCrossi...
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="com.simpli.RailwayCrossing"%>
4 <%@ page import="com.simpli.RailwayCrossingDAO"%>
5 <%@ page import="java.util.List"%>
6
7 <%
8   // Create an instance of RailwayCrossingDAO
9   RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
10
11   // Retrieve all crossings from the DAO
12   List<RailwayCrossing> crossings = crossingDAO.getAllCrossings();
13
14   // Get the ID from the search request parameter
15   String searchId = request.getParameter("searchId");
16
17   // Check if the searchId parameter is provided
18   if (searchId != null && !searchId.isEmpty()) {
19     // Parse the searchId as an int
20     int crossingId = Integer.parseInt(searchId);
21
22     // Retrieve the crossing by ID from the DAO
23     RailwayCrossing searchedCrossing = crossingDAO.getCrossingById(crossingId);
24
25     // Set the crossing as an attribute to be displayed in the table
26     request.setAttribute("crossing", searchedCrossing);
27   }
28 %>
29 <!DOCTYPE html>
30 <html>
31 <head>
32 <meta charset="UTF-8">
33 <title>Admin Home Page</title>
34 <style>
35 /* Button Styles */
36 .button-container {
37   display: flex;
38   justify-content: flex-start;
39   margin-bottom: 10px;
40   margin-left: 5px;
41 }
42
43 .button-container form {
44   margin-right: 100px;
45 }
46

```

```

addRailCross... adminHome.jsp x adminLogin.jsp HomePage.jsp searchCrossi... updateRailCr...
46
47 .button-container input[type="text"] {
48   margin-right: 5px;
49   padding: 5px;
50   border-radius: 4px;
51   border: 1px solid #ccc;
52 }
53
54 .button-container button[type="submit"] {
55   padding: 10px 20px;
56   border-radius: 4px;
57   border: none;
58   background-color: #B68686;
59   font-weight: bold;
60   color: black;
61   cursor: pointer;
62 }
63
64 .button-container button[type="submit"]:hover {
65   background-color: #450449;
66 }
67
68 .update-button, .delete-button {
69   display: inline-block;
70   vertical-align: middle;
71   margin-right: 10px;
72   padding: 7px;
73   width: 80px;
74   height: 30px;
75 }
76
77 /* Table Styles */
78 table {
79   border-collapse: collapse;
80   width: 100%;
81 }
82
83 th, td {
84   text-align: left;
85   padding: 8px;
86   border-bottom: 1px solid #ddd;
87 }
88
89 th {
90   background-color: #f2f2f2;
91 }

```

```

addRailCross... adminHome.jsp x adminLogin.jsp HomePage.jsp searchCrossi... updateRailCr...
90 background-color: #f2f2f2;
91 }
92
93 /* Responsive Styles */
94 @media only screen and (max-width: 768px) {
95     .button-container form {
96         margin-right: 5px;
97     }
98     .update-button, .delete-button {
99         float: none;
100         margin-right: 5px;
101         margin-bottom: 5px;
102     }
103 }
104 </style>
105 </head>
106 <body>
107     <h1>Admin Home Page</h1>
108     <h3>
109         Railway Crossing[<%= crossings.size() %>]
110     </h3>
111
112     <!-- Buttons -->
113     <div class="button-container">
114         <form action="HomePage.jsp" method="post">
115             <button type="submit">Home</button>
116         </form>
117         <form action="addRailCrossingForm.jsp" method="post">
118             <button type="submit">Add Railway Crossing</button>
119         </form>
120         <form action="searchCrossing" method="post">
121             <input type="text" name="searchId" placeholder="Enter ID">
122             <button type="submit">Search Crossing</button>
123         </form>
124         <form action="adminHome.jsp" method="post">
125             <button type="submit">Display All Railway Crossings</button>
126         </form>
127     <!-- Logout Button -->
128     <form action="adminRegister.jsp" method="post">
129         <button type="submit">Logout</button>
130     </form>
131 </div>
132
133 <!-- Table -->
134 <table>
135     <thead>

```

```

addRailCross... adminHome.jsp x adminLogin.jsp HomePage.jsp searchCrossi... updateRailCr...
135 <thead>
136     <tr>
137         <th>Sr.No</th>
138         <th>Name</th>
139         <th>Address</th>
140         <th>Landmark</th>
141         <th>Train Schedule</th>
142         <th>Person In-Charge</th>
143         <th>Status</th>
144         <th>Action</th>
145     </tr>
146 </thead>
147 <tbody>
148     <%
149         if (request.getAttribute("crossing") != null) { // Check if a specific crossing is found
150             RailwayCrossing crossing = (RailwayCrossing) request.getAttribute("crossing");
151             %>
152             <tr>
153                 <td><%= crossing.getId() %></td>
154                 <td><%= crossing.getName() %></td>
155                 <td><%= crossing.getAddress() %></td>
156                 <td><%= crossing.getLandmark() %></td>
157                 <td><%= crossing.getTrainSchedule() %></td>
158                 <td><%= crossing.getPersonInCharge() %></td>
159                 <td><%= crossing.getStatus() %></td>
160                 <td></td>
161             </tr>
162             <%
163         } else { // Display all crossings
164             for (RailwayCrossing anotherCrossing : crossings) {
165                 %>
166                 <tr>
167                     <td><%= anotherCrossing.getId() %></td>
168                     <td><%= anotherCrossing.getName() %></td>
169                     <td><%= anotherCrossing.getAddress() %></td>
170                     <td><%= anotherCrossing.getLandmark() %></td>
171                     <td><%= anotherCrossing.getTrainSchedule() %></td>
172                     <td><%= anotherCrossing.getPersonInCharge() %></td>
173                     <td><%= anotherCrossing.getStatus() %></td>
174                     <td>
175                         <form action="updateRailCrossingForm.jsp" method="post">
176                             <input type="hidden" name="id"
177                                 value="<%= anotherCrossing.getId() %>"
178                             <button type="submit" class="update-button">Update</button>
179                         </form>
180                         <form action="deleteCrossing" method="post">

```

```

179         </form>
180         <form action="deleteCrossing" method="post">
181             <input type="hidden" name="id"
182                 value="<%= anotherCrossing.getId() %>"
183             <button type="submit" class="delete-button">Delete</button>
184         </form>
185     </td>
186 </tr>
187     <%
188     }
189     %>
190 </tbody>
191 </table>
192 </body>
193 </html>
194
195

```



```

adminLogin.jsp × HomePage.jsp searchCrossing.jsp updateRailCrossingForm.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Admin Login</title>
8 <style>
9 body {
10  display: flex;
11  justify-content: center;
12  align-items: center;
13  height: 100vh;
14  text-align: center;
15  background-color: #f2f2f2;
16 }
17
18 .panel {
19  width: 400px;
20  padding: 40px;
21  background-color: #ffffff;
22  border-radius: 5px;
23  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
24 }
25
26 .panel label {
27  display: block;
28  margin-bottom: 10px;
29  text-align: left;
30 }
31
32 .panel input[type="text"], .panel input[type="password"] {
33  width: 100%;
34  padding: 10px;
35  border: 1px solid #ccc;
36  border-radius: 4px;
37 }
38
39 .panel button[type="submit"] {
40  width: 100%;
41  padding: 10px;
42  background-color: #4CAF50;
43  color: white;
44  border: none;
45  border-radius: 4px;
46  cursor: pointer;

```

```

44  border: none;
45  border-radius: 4px;
46  cursor: pointer;
47 }
48
49 .panel button[type="submit"]:hover {
50  background-color: #45a049;
51 }
52
53 .panel .signup-link {
54  display: block;
55  margin-top: 20px;
56 }
57 </style>
58 </head>
59 <body>
60 <div class="panel">
61   <h2>Railway Crossing</h2>
62   <h4>Admin Login</h4>
63   <form action="adminLogin" method="post">
64     <label for="email">Email:</label> <input type="text" id="email"
65       name="email" required><br> <label for="password">Password:</label>
66     <input type="password" id="password" name="password" required><br>
67     <button type="submit">Login</button>
68   </form>
69 </div>
70 </body>
71 </html>
72

```

```
HomePage.jsp × searchCrossing.jsp updateRailCrossingForm.jsp userHome.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2 pageEncoding="UTF-8"%>
3 <!doctype html>
4 <html lang="en">
5 <head>
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8 <title>Railway Crossing Status </title>
9 <link
10 href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
11 rel="stylesheet"
12 integrity="sha384-9ndCyul1bzA12FUVX7J0CjmCapSmO7SnpJef0486ghLnuZ2cdeR002iuk66FUVVM"
13 crossorigin="anonymous">
14
15 <style>
16 body {
17 background-image:
18 url("https://source.unsplash.com/1000x1000/?nature,mountain");
19 background-repeat: no-repeat;
20 background-size: cover;
21 }
22
23 .navbar-brand span, a, a:active {
24 color: red;
25 }
26
27 .navbar-nav custom, a:active {
28 margin-left: auto;
29 }
30
31 .navbar-nav custom, a:active {
32 margin-left: 10px;
33 }
34 </style>
35 </head>
36 <body>
37 <nav
38 class="navbar navbar-expand-lg bg-body-tertiary navbar-dark border-bottom border-bottom-dark"
39 data-bs-theme="dark">
40 <div class="container-fluid">
41 <a class="navbar-brand" href="#">Railway Crossing Status </a>
42 <button class="navbar-toggler" type="button"
43 data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
44 aria-controls="navbarSupportedContent" aria-expanded="false"
45 aria-label="Toggle navigation">
46 <span class="navbar-toggler-icon"> </span>
47 </button>
48 <div class="collapse navbar-collapse" id="navbarSupportedContent">
49 <ul class="navbar-nav custom ml-auto">
50 <li> Move the "Home", "Quit", "Login/Signup" links to the right using custom CSS -->
51 <li class="nav-item"> <a class="nav-link active"
52 aria-current="page" href="#">Home </a> </li>
53 <li class="nav-item"> <a class="nav-link" href="adminLogin.jsp">Admin-Login/Signup </a> </li>
54 <li class="nav-item"> <a class="nav-link" href="userLogin.jsp">User-Login/Signup </a>
55 </li>
56 </ul>
57 </div>
58 </div>
59 </nav>
60 <script
61 src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
62 integrity="sha384-WF76RCmltn28qWWowPQNgul.3RmW+VBC9Fh6dIKrxdIJTgb./j/685Ly3Te48kz"
63 crossorigin="anonymous"> </script>
64 </body>
65 </html>
66
```

```
searchCrossing.jsp × updateRailCrossingForm.jsp userHome.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2 pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="UTF-8">
8 <title>Search Crossing </title>
9 </head>
10
11 <body>
12 <h1>Search Crossing </h1>
13
14 <form action="searchCrossing" method="post">
15 <label for="searchId">Search ID: </label> <input type="number"
16 id="searchId" name="searchId" required> <input type="submit"
17 value="Search">
18 </form>
19
20 <div test="${not empty crossing}">
21 <h2>Search Result </h2>
22 <table>
23 <tr>
24 <th>ID </th>
25 <th>Name </th>
26 <th>Address </th>
27 <th>Landmark </th>
28 <th>Train Schedule </th>
29 <th>Person in Charge </th>
30 <th>Status </th>
31 </tr>
32 <tr>
33 <td>${crossing.id} </td>
34 <td>${crossing.name} </td>
35 <td>${crossing.address} </td>
36 <td>${crossing.landmark} </td>
37 <td>${crossing.trainSchedule} </td>
38 <td>${crossing.personInCharge} </td>
39 <td>${crossing.status} </td>
40 </tr>
41 </table>
42 </div>
43 <a href="adminHome.jsp">Back to Admin Homepage </a>
44 </body>
45 </html>
46
47
```

```
updateRailCrossingForm.jsp × userHome.jsp userHomeFavoriteCrossing.jsp userLogin.jsp userRegister.jsp userSearchCrossing.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7   <title>Update Railway Crossing</title>
8   <style>
9     body {
10       display: flex;
11       justify-content: center;
12       align-items: center;
13       height: 100vh;
14       text-align: center;
15       background-color: #f2f2f2;
16     }
17
18     .panel {
19       width: 500px;
20       padding: 40px;
21       background-color: #ffffff;
22       border-radius: 5px;
23       box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
24     }
25
26     .panel label {
27       display: block;
28       margin-bottom: 10px;
29       text-align: left;
30     }
31
32     .panel input[type="text"], .panel select {
33       width: 100%;
34       padding: 8px;
35       border: 1px solid #ccc;
36       border-radius: 4px;
37     }
38
39     .panel button[type="submit"] {
40       width: 100%;
41       padding: 10px;
42       background-color: #4CAF50;
43       color: white;
44       border: none;
45       border-radius: 4px;
46       cursor: pointer;
```

```
46       cursor: pointer;
47     }
48
49     .panel button[type="submit"]:hover {
50       background-color: #45a049;
51     }
52   </style>
53 </head>
54 <body>
55   <div class="panel">
56     <h2>Railway Crossings</h2>
57     <h4>Update Railway Crossing Information</h4>
58     <form action="updateCrossing" method="post">
59       <label for="crossingId">Crossing ID</label> <input type="text"
60         id="crossingId" name="crossingId" value="{crossing.crossingId}"
61         required><br> <label for="name">Enter Name</label> <input
62         type="text" id="name" name="name" value="{crossing.name}" required><br>
63       <label for="address">Address</label> <input type="text" id="address"
64         name="address" value="{crossing.address}" required><br>
65       <label for="landmark">Landmark</label> <input type="text"
66         id="landmark" name="landmark" value="{crossing.landmark}" required><br>
67       <label for="trainSchedules">Train Schedules</label> <input
68         type="text" id="trainSchedules" name="trainSchedules"
69         value="{crossing.trainSchedule}" required><br> <label
70         for="personInCharge">Person in Charge</label> <input type="text"
71         id="personInCharge" name="personInCharge"
72         value="{crossing.personInCharge}" required><br> <label
73         for="status">Crossing Status</label> <select id="status"
74         name="status">
75         <option value="Open" {crossing.status == 'Open' ? 'selected' : ''}>Open</option>
76         <option value="Closed"
77           {crossing.status == 'Closed' ? 'selected' : ''}>Closed</option>
78       </select><br>
79       <button type="submit">Submit</button>
80     </form>
81   </div>
82 </body>
83 </html>
84
```

```

1  *userHome.jsp × userHomeFavoriteCrossing.jsp userLogin.jsp userRegister.jsp userSearchCrossing.jsp
2  <%@ page language="java" contentType="text/html; charset=UTF-8"%>
3  <%@ include page="com.singul.RailwayCrossing"%>
4  <%@ include page="com.singul.RailwayCrossingDAO"%>
5  <%@ include page="com.singul.List"%>
6  <DOCTYPE html>
7  <html>
8  <head>
9  <meta charset="UTF-8"%>
10 <title>Railway Crossings - User Home Page</title>
11 <style>
12 body {
13 font-family: Arial, sans-serif;
14 margin: 0;
15 padding: 20px;
16 }
17
18 .container {
19 border: 1px solid #ccc;
20 border-radius: 10px;
21 padding: 10px;
22 margin-bottom: 20px;
23 margin-top: 20px;
24 }
25
26 .container h3 {
27 margin-top: 0;
28 }
29
30 .container p {
31 margin: 0;
32 }
33
34 .crossing {
35 margin-bottom: 10px;
36 }
37
38 .crossing button {
39 margin-left: 10px;
40 }
41
42 .status {
43 display: inline-block;
44 padding: 5px;
45 font-weight: bold;
46 }
47
48 .status-open {
49 color: white;
50 border-radius: 30%;
51 background-color: green;
52 }
53
54 .status-closed {
55 color: white;
56 background-color: red;
57 }
58 </style>
59 </head>
60 <body>
61 <h1 style="font-weight: bold;">User Home Page</h1>
62
63 <!-- Display Search Crossings button -->
64 <button onclick="location.href='HomePage.jsp'"
65 style="border-radius: 10px; font-weight: bold;">Home</button>
66
67 <!-- Display All button -->
68 <button onclick="location.href='userHome.jsp'"
69 style="border-radius: 10px; margin-left: 50px; font-weight: bold;">All
70 Crossings</button>
71
72 <!-- Display Favorite Crossings button -->
73 <button onclick="location.href='userHomeFavoriteCrossing.jsp'"
74 style="border-radius: 10px; margin-left: 50px; font-weight: bold;">Favorite
75 Crossings</button>

```

```

74 style="border-radius: 10px; margin-left: 50px; font-weight: bold;">Favorite
75 Crossings</button>
76
77 <!-- Display Search Crossings button -->
78 <button onclick="location.href='userSearchCrossing.jsp'"
79 style="border-radius: 10px; margin-left: 50px; font-weight: bold;">Search
80 Crossings</button>
81
82 <!-- Logout button -->
83 <button onclick="location.href='userLogin.jsp'"
84 style="border-radius: 10px; margin-left: 800px; font-weight: bold;">Logout</button>
85
86 <!-- Create an instance of RailwayCrossingDAO -->
87 <%
88 RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
89 List<RailwayCrossing> allCrossings = crossingDAO.getAllCrossings();
90 for (RailwayCrossing crossing : allCrossings) {
91 String status = crossing.getStatus();
92 String statusClass = status.equalsIgnoreCase("Open") ? "status-open" : "status-closed";
93
94 <div class="container">
95 <h3><%=crossing.getName()%></h3>
96 <p>
97 Status: <strong><span class="status <%=statusClass%>"><%=crossing.getStatus()%></span></strong>
98 </p>
99 <p>
100 Person in Charge: <strong><%=crossing.getPersonInCharge()%></strong>
101 </p>
102 <p>
103 Train Schedules: <strong><%=crossing.getTrainSchedule()%></strong>
104 </p>
105 <p>
106 Landmark: <strong><%=crossing.getLandmark()%></strong>
107 </p>
108 <p>
109 Address: <strong><%=crossing.getAddress()%></strong>
110 </p>
111 <form action="addToFavorites" method="post" style="display: inline;">
112 <input type="hidden" name="crossingId" value="<%=crossing.getId()%>">
113 <button type="submit"
114 style="border-radius: 10px; font-weight: bold; margin-top: 10px; margin-left: 0px;">ADD
115 TO FAVORITES</button>
116 </form>
117 </div>
118 <%
119 }
120 %>
121 </body>
122 </html>
123

```

```

1  *userHome.jsp
2  <%@ page language="java" contentType="text/html; charset=UTF-8"
3  pageEncoding="UTF-8"%>
4  <%@ page import="com.simpli.RailwayCrossing"%>
5  <%@ page import="com.simpli.RailwayCrossingDAO"%>
6  <%@ page import="java.util.List"%>
7  <!DOCTYPE html>
8  <html>
9  <head>
10 <meta charset="UTF-8">
11 <title>Railway Crossings - Favorite Crossings</title>
12 <style>
13 body {
14 font-family: Arial, sans-serif;
15 margin: 0;
16 padding: 20px;
17 }
18 .container {
19 border: 1px solid #ccc;
20 border-radius: 10px;
21 padding: 20px;
22 margin-bottom: 20px;
23 margin-top: 20px;
24 }
25 .container h3 {
26 margin-top: 0;
27 }
28 .container p {
29 margin: 0;
30 }
31 }
32
33 .status {
34 display: inline-block;
35 padding: 5px;
36 color: white;
37 font-weight: bold;
38 }
39
40 .status-open {
41 background-color: green;
42 }
43
44 .status-closed {
45 background-color: red;
46 }
47 </style>
48 </head>
49 <body>
50 <h2>User Home Page</h2>
51
52 <!-- Display All button -->
53 <button onclick="location.href='userHome.jsp'">All Crossings</button>
54
55 <!-- Favorite Crossings Container -->
56 <div class="favorites">
57 <%
58 RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
59 List<RailwayCrossing> favoriteCrossings = crossingDAO.getFavoriteCrossings();
60 for (RailwayCrossing crossing : favoriteCrossings) {
61 String status = crossing.getStatus();
62 String statusClass = status.equalsIgnoreCase("Open") ? "status-open" : "status-closed";
63 %>
64 <div class="container">
65 <h3><%=crossing.getName()%></h3>
66 <p>
67 Status: <strong><%=crossing.getStatusClass()%><%=status%></span></strong>
68 </p>
69 <p>
70 Person in Charge: <strong><%=crossing.getPersonInCharge()%></strong>
71 </p>
72 <p>
73 Train Schedules: <strong><%=crossing.getTrainSchedule()%></strong>
74 </p>
75 <p>

```

```

74 </p>
75 <p>
76 Landmark: <strong><%=crossing.getLandmark()%></strong>
77 </p>
78 <p>
79 Address: <strong><%=crossing.getAddress()%></strong>
80 </p>
81 <form action="removeFromFavorites" method="post"
82 style="display: inline;">
83 <input type="hidden" name="crossingId" value="<%=crossing.getId()%>">
84 <button type="submit"
85 style="border-radius: 10px; font-weight: bold; margin-top: 10px;">REMOVE
86 FROM FAVORITES</button>
87 </form>
88 </div>
89 <%
90 }
91 %>
92 </div>
93 </body>
94 </html>
95

```

```
*userLogin.jsp × userRegister.jsp userSearchCrossing.jsp
1  <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2  <pageEncoding="UTF-8"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="UTF-8"%>
7  <title>User Login</title>
8  <style>
9  body {
10     display: flex;
11     justify-content: center;
12     align-items: center;
13     height: 100vh;
14     text-align: center;
15     background-color: #f2f2f2;
16 }
17
18 .panel {
19     width: 400px;
20     padding: 40px;
21     background-color: #ffffff;
22     border-radius: 5px;
23     box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
24 }
25
26 .panel label {
27     display: block;
28     margin-bottom: 10px;
29     text-align: left;
30 }
31
32 .panel input[type="text"], .panel input[type="password"] {
33     width: 100%;
34     padding: 10px;
35     border: 1px solid #ccc;
36     border-radius: 4px;
37 }
38
39 .panel button[type="submit"] {
40     width: 100%;
41     padding: 10px;
42     background-color: #4CAF50;
43     color: white;
44     border: none;
45     border-radius: 4px;
46     cursor: pointer;
47 }
48
49 .panel button[type="submit"]:hover {
50     background-color: #45a049;
51 }
52
53 .panel .signup-link {
54     display: block;
55     margin-top: 20px;
56 }
57 </style>
58 </head>
59 <body>
60     <div class="panel">
61         <h2>Railway Crossing</h2>
62         <h4>User Login</h4>
63         <form action="login" method="post">
64             <label for="email">Email</label> <input type="text" id="email"
65             name="email" required><br> <label for="password">Password</label>
66             <input type="password" id="password" name="password" required><br>
67             <button type="submit">Login</button>
68         </form>
69         <div class="signup-link">
70             <p>
71                 Don't have an account? <a href="userRegister.jsp">Create New
72                 Account</a>
73             </p>
74         </div>
75     </div>
76 </body>
77 </html>
```

```
*userRegister.jsp × userSearchCrossing.jsp
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2  pageEncoding="UTF-8"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="UTF-8"%>
7  <title>User Registration</title>
8  <style>
9  body {
10     display: flex;
11     justify-content: center;
12     align-items: center;
13     height: 100vh;
14     text-align: center;
15     background-color: #f2f2f2;
16 }
17 panel {
18     width: 500px;
19     padding: 40px;
20     background-color: #ffffff;
21     border-radius: 5px;
22     box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
23 }
24 panel label {
25     display: block;
26     margin-bottom: 10px;
27     text-align: left;
28 }
29 panel input[type="text"], panel input[type="password"] {
30     width: 100%;
31     padding: 10px;
32     border: 1px solid #ccc;
33     border-radius: 4px;
34 }
35 panel button[type="submit"] {
36     width: 100%;
37     padding: 10px;
38     background-color: #4CAF50;
39     color: white;
40     border: none;
41     border-radius: 4px;
42     cursor: pointer;
43 }
44 panel button[type="submit"]:hover {
45     background-color: #45a049;
46 }
47 panel .signup-link {
48     display: block;
49     margin-top: 20px;
50 }
51 }
52 </style>
53 </head>
54 <body>
55     <div class="panel">
56         <h2>Railway Crossing</h2>
57         <h4>User Register</h4>
58         <form action="register" method="post">
59             <label for="name">Name:</label> <input type="text" id="name"
60             name="name" required><br> <label for="email">Email:</label>
61             <input type="text" id="email" name="email" required><br>
62             <label for="password">Password:</label> <input type="password"
63             id="password" name="password" required><br>
64             <button type="submit">Register</button>
65         </form>
66         <div class="signup-link">
67             <p>
68                 Already have an account? <a href="userLogin.jsp">Sign in</a>
69             </p>
70         </div>
71     </div>
72 </body>
73 </html>
74
```

```

userSearchCrossing.jsp ×
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="com.simpli.RailwayCrossing"%>
4 <%@ page import="com.simpli.RailwayCrossingDAO"%>
5 <%@ page import="java.util.List"%>
6 <!DOCTYPE html>
7 <html>
8 <head>
9   <meta charset="UTF-8">
10  <title>Railway Crossings - Search Crossings</title>
11 <style>
12 body {
13   font-family: Arial, sans-serif;
14   margin: 0;
15   padding: 20px;
16 }
17
18 .container {
19   border: 1px solid #ccc;
20   border-radius: 10px;
21   padding: 20px;
22   margin-bottom: 20px;
23   margin-top: 20px;
24 }
25
26 .container h3 {
27   margin-top: 0;
28 }
29
30 .container p {
31   margin: 0;
32 }
33
34 .search-container {
35   margin-top: 60px;
36   margin-bottom: 20px;
37 }
38
39 .search-container input[type="text"] {
40   padding: 10px;
41   width: 200px;
42 }
43
44 .logout-button {
45   position: absolute;
46   padding: 10px;

```

```

userSearchCrossing.jsp ×
46   padding: 10px;
47   top: 50px;
48   right: 30px;
49   border-radius: 10px;
50   font-weight: bold;
51   top: 50px;
52 }
53
54 .allCrossing-button {
55   position: absolute;
56   padding: 5px;
57   top: 75px;
58   left: 10px;
59 }
60
61 .favouriteCrossing-button {
62   position: absolute;
63   padding: 5px;
64   top: 75px;
65   left: 130px;
66 }
67
68 .status {
69   display: inline-block;
70   padding: 5px;
71   font-weight: bold;
72 }
73
74 .status-open {
75   color: white;
76   border-radius: 30%;
77   background-color: green;
78 }
79
80 .status-closed {
81   color: white;
82   background-color: red;
83 }
84
85 .crossing {
86   margin-bottom: 10px;
87 }
88
89 .crossing button {
90   margin-left: 10px;
91 }

```



```

userSearchCrossing.jsp ×
91 }
92 </style>
93 </script>
94 function searchCrossings() {
95     var input = document.getElementById("searchInput");
96     var filter = input.value.toUpperCase();
97     var crossings = document.getElementsByClassName("crossing");
98
99     for (var i = 0; i < crossings.length; i++) {
100         var crossing = crossings[i];
101         var name = crossing.getElementsByTagName("h3")[0].innerText;
102         if (name.toUpperCase().indexOf(filter) > -1) {
103             crossing.style.display = "";
104         } else {
105             crossing.style.display = "none";
106         }
107     }
108 }
109 </script>
110 </head>
111 <body>
112     <h2>Search Crossings</h2>
113     <!-- Display All button -->
114     <button onclick="location.href='userHome.jsp'"
115         class="allCrossing-button"
116         style="border-radius: 10px; font-weight: bold;">All Crossings</button>
117
118     <!-- Display Favorite Crossings button -->
119     <button onclick="location.href='userHomeFavoriteCrossing.jsp'"
120         class="favouriteCrossing-button"
121         style="border-radius: 10px; font-weight: bold;">Favorite
122         Crossings</button>
123
124     <!-- Search input -->
125     <div class="search-container">
126         <input type="text" id="searchInput" placeholder="Search by name"
127             onkeyup="searchCrossings()">
128         <!-- Logout button -->
129         <button onclick="location.href='userLogin.jsp'" class="logout-button"
130             style="border-radius: 10px; font-weight: bold;">Logout</button>
131     </div>
132
133     <!-- Search Results Container -->
134     <div class="search-results">
135         <%

```

```

userSearchCrossing.jsp ×
128     <!-- Logout button -->
129     <button onclick="location.href='userLogin.jsp'" class="logout-button"
130         style="border-radius: 10px; font-weight: bold;">Logout</button>
131 </div>
132
133
134 <!-- Search Results Container -->
135 <div class="search-results">
136     <%
137     RailwayCrossingDAO crossingDAO = new RailwayCrossingDAO();
138     List<RailwayCrossing> allCrossings = crossingDAO.getAllCrossings();
139     for (RailwayCrossing crossing : allCrossings) {
140         String status = crossing.getStatus();
141         String statusClass = status.equalsIgnoreCase("Open") ? "status-open" : "status-closed";
142     >%
143     <div class="container crossing">
144         <h3><%=crossing.getName()%></h3>
145         <p>
146             Status: <strong><span class="status <%=statusClass%>"><%=crossing.getStatus()%></span></strong>
147         </p>
148         <p>
149             Person in Charge: <strong><%=crossing.getPersonInCharge()%></strong>
150         </p>
151         <p>
152             Train Schedules: <strong><%=crossing.getTrainSchedule()%></strong>
153         </p>
154         <p>
155             Landmark: <strong><%=crossing.getLandmark()%></strong>
156         </p>
157         <p>
158             Address: <strong><%=crossing.getAddress()%></strong>
159         </p>
160         <form action="addToFavorites" method="post" style="display: inline;">
161             <input type="hidden" name="crossingId" value="<%=crossing.getId()%>">
162             <button type="submit"
163                 style="border-radius: 10px; font-weight: bold; margin-top: 10px; margin-left: 0px;">ADD
164                 TO FAVORITES</button>
165         </form>
166     </div>
167     <%
168     }
169     <%
170 </div>
171 </body>
172 </html>
173

```

# RailwayCrossingStatus/pom.xml × web.xml

```

1  https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
5    <modelVersion>4.0.0</modelVersion>
6    <groupId>com.simpli.railwaycrossing</groupId>
7    <artifactId>RailwayCrossingStatus</artifactId>
8    <version>0.0.1-SNAPSHOT</version>
9    <packaging>war</packaging>
10   <name>Building an application to communicate the status of railway crossings
11     in advance</name>
12   <properties>
13     <maven.compiler.source>1.8</maven.compiler.source>
14     <maven.compiler.target>1.8</maven.compiler.target>
15   </properties>
16   <dependencies>
17     <!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
18     <dependency>
19       <groupId>javax.servlet</groupId>
20       <artifactId>jstl</artifactId>
21       <version>1.2</version>
22     </dependency>
23
24     <!-- MySQL Connector -->
25     <dependency>
26       <groupId>mysql</groupId>
27       <artifactId>mysql-connector-java</artifactId>
28       <version>8.0.33</version>
29     </dependency>
30
31     <!-- Jakarta Servlet API -->
32     <dependency>
33       <groupId>javax.servlet</groupId>
34       <artifactId>javax.servlet-api</artifactId>
35       <version>4.0.0</version>
36       <scope>provided</scope>
37     </dependency>
38   </dependencies>
39
40   <build>
41     <plugins>
42       <plugin>
43         <artifactId>maven-compiler-plugin</artifactId>
44         <version>3.8.1</version>
45         <configuration>
46           <release>8</release>
47         </configuration>
48       </plugin>
49       <plugin>
50         <artifactId>maven-war-plugin</artifactId>
51         <version>3.2.3</version>
52       </plugin>
53     </plugins>
54   </build>
55 </project>
56

```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://jakarta.ee/xml/ns/jakartaee"
4         xmlns:web="http://xmlns.jcp.org/xml/ns/javaee"
5         xsi:schemaLocation="http://jakarta.ee/xml/ns/jakartaee http://jakarta.ee/xml/ns/jakartaee/web-app_5_0.xsd"
6         id="WebApp_ID" version="5.0">
7     <display-name>RailwayCrossingStatus</display-name>
8     <welcome-file-list>
9         <welcome-file>HomePage.jsp</welcome-file>
10        <welcome-file>index.jsp</welcome-file>
11        <welcome-file>index.htm</welcome-file>
12        <welcome-file>default.html</welcome-file>
13        <welcome-file>default.jsp</welcome-file>
14        <welcome-file>default.htm</welcome-file>
15    </welcome-file-list>
16    <servlet>
17        <description></description>
18        <display-name>GovernmentServlet</display-name>
19        <servlet-name>GovernmentServlet</servlet-name>
20        <servlet-class>com.simpli.GovernmentServlet</servlet-class>
21    </servlet>
22    <servlet-mapping>
23        <servlet-name>GovernmentServlet</servlet-name>
24        <url-pattern>/GovernmentServlet</url-pattern>
25    </servlet-mapping>
26    <servlet>
27        <description></description>
28        <display-name>LoginServlet</display-name>
29        <servlet-name>LoginServlet</servlet-name>
30        <servlet-class>com.simpli.LoginServlet</servlet-class>
31    </servlet>
32    <servlet-mapping>
33        <servlet-name>LoginServlet</servlet-name>
34        <url-pattern>/LoginServlet</url-pattern>
35    </servlet-mapping>
36    <servlet>
37        <description></description>
38        <display-name>RegisterServlet</display-name>
39        <servlet-name>RegisterServlet</servlet-name>
40        <servlet-class>com.simpli.RegisterServlet</servlet-class>
41    </servlet>
42    <servlet-mapping>
43        <servlet-name>RegisterServlet</servlet-name>
44        <url-pattern>/RegisterServlet</url-pattern>
45    </servlet-mapping>
46    <servlet>
47        <description></description>
48        <display-name>AdminRegisterServlet</display-name>
49        <servlet-name>AdminRegisterServlet</servlet-name>
50        <servlet-class>com.simpli.AdminRegisterServlet</servlet-class>
51    </servlet>
52    <servlet-mapping>
53        <servlet-name>AdminRegisterServlet</servlet-name>
54        <url-pattern>/AdminRegisterServlet</url-pattern>
55    </servlet-mapping>
56    <servlet>
57        <description></description>
58        <display-name>AdminLoginServlet</display-name>
59        <servlet-name>AdminLoginServlet</servlet-name>
60        <servlet-class>com.simpli.AdminLoginServlet</servlet-class>
61    </servlet>
62    <servlet-mapping>
63        <servlet-name>AdminLoginServlet</servlet-name>
64        <url-pattern>/AdminLoginServlet</url-pattern>
65    </servlet-mapping>
66    <servlet>
67        <description></description>
68        <display-name>AdminHomePageServlet</display-name>
69        <servlet-name>AdminHomePageServlet</servlet-name>
70        <servlet-class>com.simpli.AdminHomePageServlet</servlet-class>
71    </servlet>
```

```
66    <servlet>
67        <description></description>
68        <display-name>AdminHomePageServlet</display-name>
69        <servlet-name>AdminHomePageServlet</servlet-name>
70        <servlet-class>com.simpli.AdminHomePageServlet</servlet-class>
71    </servlet>
72    <servlet-mapping>
73        <servlet-name>AdminHomePageServlet</servlet-name>
74        <url-pattern>/AdminHomePageServlet</url-pattern>
75    </servlet-mapping>
76    <servlet>
77        <description></description>
78        <display-name>UpdateCrossingServlet</display-name>
79        <servlet-name>UpdateCrossingServlet</servlet-name>
80        <servlet-class>com.simpli.UpdateCrossingServlet</servlet-class>
81    </servlet>
82    <servlet-mapping>
83        <servlet-name>UpdateCrossingServlet</servlet-name>
84        <url-pattern>/UpdateCrossingServlet</url-pattern>
85    </servlet-mapping>
86    <servlet>
87        <description></description>
88        <display-name>DeleteCrossingServlet</display-name>
89        <servlet-name>DeleteCrossingServlet</servlet-name>
90        <servlet-class>com.simpli.DeleteCrossingServlet</servlet-class>
91    </servlet>
92    <servlet-mapping>
93        <servlet-name>DeleteCrossingServlet</servlet-name>
94        <url-pattern>/DeleteCrossingServlet</url-pattern>
95    </servlet-mapping>
96    <servlet>
97        <description></description>
98        <display-name>SearchCrossingServlet</display-name>
99        <servlet-name>SearchCrossingServlet</servlet-name>
100        <servlet-class>com.simpli.SearchCrossingServlet</servlet-class>
101    </servlet>
102    <servlet-mapping>
103        <servlet-name>SearchCrossingServlet</servlet-name>
104        <url-pattern>/searchCrossing</url-pattern>
105    </servlet-mapping>
106    <servlet>
107        <description></description>
108        <display-name>AddCrossingServlet</display-name>
109        <servlet-name>AddCrossingServlet</servlet-name>
110        <servlet-class>com.simpli.AddCrossingServlet</servlet-class>
111    </servlet>
112    <servlet-mapping>
113        <servlet-name>AddCrossingServlet</servlet-name>
114        <url-pattern>/addCrossing</url-pattern>
115    </servlet-mapping>
116    <servlet>
117        <description></description>
118        <display-name>AddToFavoritesServlet</display-name>
119        <servlet-name>AddToFavoritesServlet</servlet-name>
120        <servlet-class>com.simpli.AddToFavoritesServlet</servlet-class>
121    </servlet>
122    <servlet-mapping>
123        <servlet-name>AddToFavoritesServlet</servlet-name>
124        <url-pattern>/addToFavorites</url-pattern>
125    </servlet-mapping>
126    <servlet>
127        <description></description>
128        <display-name>RemoveFromFavoritesServlet</display-name>
129        <servlet-name>RemoveFromFavoritesServlet</servlet-name>
130        <servlet-class>com.simpli.RemoveFromFavoritesServlet</servlet-class>
131    </servlet>
132    <servlet-mapping>
133        <servlet-name>RemoveFromFavoritesServlet</servlet-name>
134        <url-pattern>/removeFromFavorites</url-pattern>
135    </servlet-mapping>
136 </web-app>
```