Writeup Flow of project:

Step 1: Set up the project

1. Create a new Java web project in your preferred IDE (e.g., Eclipse, IntelliJ).

2. Include the necessary libraries for JDBC, Servlets, and JSP.

Step 2: Create the database and table

1. Set up a database of your choice (e.g., MySQL).

2. Create a table called "railway\_crossing" with the following columns:

- id (auto-increment)

- name

- address

- landmark

- train\_schedules

- person\_in\_charge

- status

Step 3: Create the JSP form for adding a railway crossing

1. Create an JSP form that collects the necessary details for adding a railway crossing.

2. Set the form action to the appropriate Servlet URL.

Step 4: Create the Servlet for adding a railway crossing

1. Create a Servlet to handle the form submission and add the railway crossing details to the database.

2. Use JDBC to establish a connection to the database.

3. Retrieve the form parameters and insert them into the "railway\_crossing" table.

4. Handle exceptions and display appropriate messages.

Step 5: Create the JSP for displaying railway crossings

1. Create a JSP page to display a table view of existing railway crossings.

2. Retrieve the railway crossing data from the database using JDBC.

3. Display the data in an HTML table format.

4. Include options to delete and update the status of each railway crossing.

Step 6: Create the Servlet for deleting a railway crossing

1. Create a Servlet to handle the deletion of a railway crossing.

2. Retrieve the ID of the railway crossing to be deleted from the request parameters.

3. Use JDBC to delete the corresponding row from the "railway\_crossing" table.

4. Handle exceptions and display appropriate messages.

Step 7: Implement exception handling, collections, and searching techniques

1. Handle exceptions in the Servlets using try-catch blocks and provide appropriate error messages.

2. Utilize collections (e.g., ArrayList) to store and manipulate railway crossing data.

3. Implement search functionality by allowing users to search for railway crossings based on specific criteria (e.g., name, address).

4. Optimize the source code by following best practices, such as using prepared statements, closing database connections properly, and minimizing database queries.

Step 8: Implement login functionality

1. Create a login form using HTML and JSP.

2. Retrieve the email and password from the form submission.

3. Use JDBC to validate the email and password against the pre-created email and password stored in the database.

4. Upon successful login, navigate the user to the dashboard page with railway crossing functionalities.

configure the web.xml file to map the Servlets and JSPs appropriately.