

Phase 2 Assessment Project

RAILWAY CROSSING STATUS

Creating Database and table for Project

create database railway_crossing_status;

use railway_crossing_status;

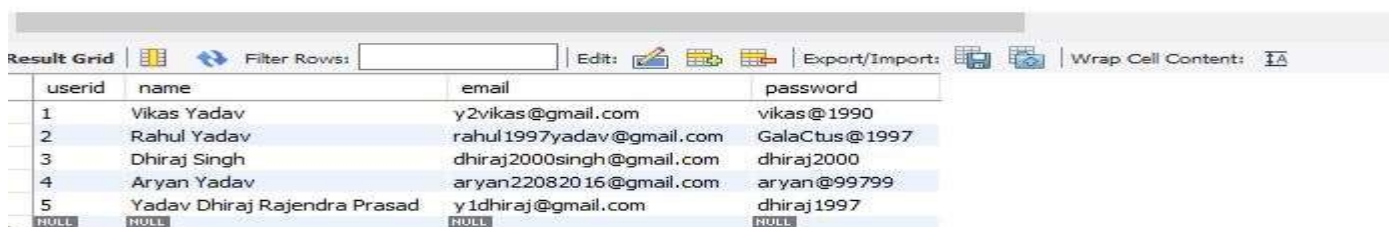
```
CREATE TABLE railway_crossing (  
    railway_crossing_id INT PRIMARY KEY auto_increment,  
    name varchar(255),  
    address varchar(255),  
    landmark varchar(100),  
    train_schedule varchar(50),  
    person_in_charge varchar(50),  
    status varchar(10)  
);
```



railway_crossing_id	name	address	landmark	train_schedule	person_in_charge	status
9	Allahabad Junction Crossing	Station Road, Prayagraj	Allahabad Junction Railway Station	Daily 6:00 AM and 9:00 PM	Rajesh Verma	Open
10	Sangam Marg Crossing	Sangam Marg, Prayagraj	Triveni Sangam	Weekends 8:00 AM	Priya Singh	Closed
12	Yamuna Bridge Crossing	Yamuna Bridge Road, Prayagraj	Yamuna Bridge	Mondays 3:30 PM	Amit Patel	Open
14	Civil Lines Crossing	Civil Lines, Prayagraj	Civil Lines Market	Daily 7:30 AM and 5:30 PM	Neha Gupta	Closed
17	Kurla Crossing	LBS Road	Kanakia Zillion Business Point	Daily 5:15 PM	Adil Khan	Closed
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE user_signup(  
    userid INT PRIMARY KEY auto_increment,  
    name varchar(255),  
    email varchar(50),  
    password varchar(50)  
);
```

```
1 • SELECT * FROM railway_crossing_status.user_signup;
```



userid	name	email	password
1	Vikas Yadav	y2vikas@gmail.com	vikas@1990
2	Rahul Yadav	rahul1997yadav@gmail.com	GalaCtus@1997
3	Dhiraj Singh	dhiraj2000singh@gmail.com	dhiraj2000
4	Aryan Yadav	aryan22082016@gmail.com	aryan@99799
5	Yadav Dhiraj Rajendra Prasad	y1dhiraj@gmail.com	dhiraj1997
NULL	NULL	NULL	NULL

```

CREATE TABLE user_signup(
    id INT PRIMARY KEY auto_increment,
    railway_crossing_id INT
    userid INT,
    FOREIGN KEY (railway_crossing_id) REFERENCES railway_crossing(railway_crossing_id)
);

```

pom.xml (Maven) :

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.railwaycrossing.railwaycrossing</groupId>
    <artifactId>RailwayCrossingStatus</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>Building an application to communicate the status of railway crossings
        in advance</name>
    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
    <dependencies>

        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>

        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>4.0.0</version>
            <scope>provided</scope>
        </dependency>

    </dependencies>
    <build>
        <plugins>
            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>

```

```

        <version>3.8.1</version>
        <configuration>
            <release>8</release>
        </configuration>
    </plugin>
    <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.2.3</version>
    </plugin>
</plugins>
</build>
</project>

```

AddCrossingServlet.java

```

package com.railwaycrossing;

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 = "AddCrossingServlet", urlPatterns = { "/addCrossing" })
public class AddCrossingServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

        String name = request.getParameter("name");
        String address = request.getParameter("address");
        String landmark = request.getParameter("landmark");
        String trainSchedules = request.getParameter("trainSchedules");
        String personInCharge = request.getParameter("personInCharge");
        String status = request.getParameter("status");

        RailwayCrossing crossing = new RailwayCrossing();
        crossing.setName(name);
        crossing.setAddress(address);
        crossing.setLandmark(landmark);
        crossing.setTrainSchedule(trainSchedules);
        crossing.setPersonInCharge(personInCharge);
        crossing.setStatus(status);

        RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();
        crossingDAO.addCrossing(crossing);

        response.sendRedirect("adminHome.jsp");
    }
}

```

AddToFavoritesServlet.java

```

package com.railwaycrossing;

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

public class AddToFavoritesServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String crossingId = request.getParameter("crossingId");
        if (crossingId != null && !crossingId.isEmpty()) {
            int railwayCrossingId = Integer.parseInt(crossingId);
            RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();
            crossingDAO.addToFavorites(railwayCrossingId);
        }
        response.sendRedirect("userHome.jsp");
    }
}

```

AdminHomePageServlet.java

```

package com.railwaycrossing;

import java.io.IOException;
import java.util.List;

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("/adminHomePage")
public class AdminHomePageServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private RailwayCrossingDatabaseOp crossingDAO;

    public void init() {
        crossingDAO = new RailwayCrossingDatabaseOp();
    }

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

        List<RailwayCrossing> crossings = crossingDAO.getAllCrossings();

        request.setAttribute("crossings", crossings);
        request.getRequestDispatcher("adminHome.jsp").forward(request, response);
    }
}

```

AdminLoginServlet.java

```

package com.railwaycrossing;

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("/adminLogin")
public class AdminLoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private static final String ADMIN_ID = "yadavdhiraj.admin@gmail.com";
    private static final String ADMIN_PASSWORD = "dhiraj1997";

    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);
        }
    }
}

```

DatabaseConnection.java

```

package com.railwaycrossing;

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_crossing_status";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "Dhiraj@1997";

    public static Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
        return connection;
    }
}

```

DeleteCrossingServlet.java

```

package com.railwaycrossing;

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("/deleteCrossing")
public class DeleteCrossingServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

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

```

```

        RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();
        crossingDAO.deleteCrossing(crossingId);

        response.sendRedirect("adminHome.jsp");
    }
}

```

LoginServlet.java

```

package com.railwaycrossing;

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

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

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

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

        UserDatabaseOp userDatabaseOp = new UserDatabaseOp();
        User user = userDatabaseOp.getUserByEmail(email);

        if (user != null && user.getPassword().equals(password)) {

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

            response.sendRedirect("userHome.jsp");
        } else {

            response.sendRedirect("userRegister.jsp?error=1");
        }
    }
}

```

RegisterServlet.java

```

package com.railwaycrossing;

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;

/**
 * Servlet implementation class RegisterServlet
 */
@WebServlet("/register")
public class RegisterServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

```

```

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

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

            User user = new User(name, email, password);

            UserDatabaseOp userDatabaseOp = new UserDatabaseOp();
            userDatabaseOp.registerUser(user);

            response.sendRedirect("userLogin.jsp");
        }
    }
}

```

RemoveFromFavouritesServlet.java

```

package com.railwaycrossing;

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

/**
 * Servlet implementation class RemoveFromFavoritesServlet
 */
public class RemoveFromFavoritesServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        String crossingId = request.getParameter("crossingId");
        if (crossingId != null && !crossingId.isEmpty()) {
            int railwayCrossingId = Integer.parseInt(crossingId);

            RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();
            crossingDAO.removeFromFavorites(railwayCrossingId);
        }

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

```

SearchCrossingServlet.java

```

package com.railwaycrossing;

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 {
    private static final long serialVersionUID = 1L;

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

        int searchId = Integer.parseInt(request.getParameter("searchId"));

        RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();
        RailwayCrossing crossing = crossingDAO.getCrossingById(searchId);

        request.setAttribute("crossing", crossing);

        request.getRequestDispatcher("adminHome.jsp").forward(request, response);
    }
}

```

UpdateCrossingServlet.java

```

package com.railwaycrossing;

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("/updateCrossing")
public class UpdateCrossingServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

            String idParam = request.getParameter("crossingId");

            if (idParam != null && !idParam.isEmpty()) {
                int crossingId = Integer.parseInt(idParam);

                String name = request.getParameter("name");
                String address = request.getParameter("address");
                String landmark = request.getParameter("landmark");
                String trainSchedules = request.getParameter("trainSchedules");
                String personInCharge = request.getParameter("personInCharge");
                String status = request.getParameter("status");

                RailwayCrossing updatedCrossing = new RailwayCrossing();
                updatedCrossing.setId(crossingId);
                updatedCrossing.setName(name);
                updatedCrossing.setAddress(address);
                updatedCrossing.setLandmark(landmark);
                updatedCrossing.setTrainSchedule(trainSchedules);
                updatedCrossing.setPersonInCharge(personInCharge);
                updatedCrossing.setStatus(status);

                RailwayCrossingDatabaseOp crossingDAO = new
RailwayCrossingDatabaseOp();
                crossingDAO.updateCrossing(updatedCrossing);
            }
        }
    }
}

```



```

        response.sendRedirect("adminHome.jsp");
    } else {
        throw new ServletException("Crossing ID is missing.");
    }

} catch (NumberFormatException e) {
    throw new ServletException("Invalid Crossing ID format.", e);
} catch (Exception e) {
    throw new ServletException("An error occurred during the crossing update.",
e);
}
}
}

```

RailwayCrossing.java

```

package com.railwaycrossing;

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();
        // TODO Auto-generated constructor stub
    }

    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;
    }

    public void setId(String string) {
        // TODO Auto-generated method stub
    }
}

```

RailwayCrossingDatabaseOp.java

```

package com.railwaycrossing;

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 RailwayCrossingDatabaseOp {
    private Connection connection;

    public RailwayCrossingDatabaseOp() {

```

```

        connection = DatabaseConnection.getConnection();
    }

    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("railway_crossing_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"));
                crossings.add(crossing);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return crossings;
    }

    public RailwayCrossing getCrossingById(int crossingId) {
        RailwayCrossing crossing = null;
        try {
            String query = "SELECT * FROM railway_crossing WHERE railway_crossing_id = ?";

            PreparedStatement statement = connection.prepareStatement(query);
            statement.setInt(1, crossingId);
            ResultSet resultSet = statement.executeQuery();
            if (resultSet.next()) {
                crossing = new RailwayCrossing();
                crossing.setId(resultSet.getInt("railway_crossing_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;
    }

    public void updateCrossing(RailwayCrossing crossing) {
        try {
            String query = "UPDATE railway_crossing SET name=?, address=?, landmark=?, train_schedule=?, person_in_charge=?, status=? WHERE railway_crossing_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();
        }
    }

    public void deleteCrossing(int crossingId) {
        try {
            connection.setAutoCommit(false);

            String deleteFavoriteCrossingsQuery = "DELETE FROM favorite_crossing WHERE
railway_crossing_id = ?";
            try (PreparedStatement deleteFavoriteCrossingsStatement = connection
                .prepareStatement(deleteFavoriteCrossingsQuery)) {
                deleteFavoriteCrossingsStatement.setInt(1, crossingId);
                deleteFavoriteCrossingsStatement.executeUpdate();
            }
            String deleteRailwayCrossingQuery = "DELETE FROM railway_crossing WHERE
railway_crossing_id = ?";
            try (PreparedStatement deleteRailwayCrossingStatement = connection
                .prepareStatement(deleteRailwayCrossingQuery)) {
                deleteRailwayCrossingStatement.setInt(1, crossingId);
                deleteRailwayCrossingStatement.executeUpdate();
            }

            connection.commit();
            connection.setAutoCommit(true);
        } catch (SQLException e) {
            try {
                connection.rollback();
            } catch (SQLException rollbackException) {
                rollbackException.printStackTrace();
            }
            e.printStackTrace();
        }
    }

    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.railway_crossing_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("railway_crossing_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();
    }
}

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();
    }
}
}
}

```

User.java

```

package com.railwaycrossing;

public class User {
    private String name;
    private String email;
    private String password;

    public User() {
        super();
    }

    public User(String name, String email, String password) {
        this.name = name;
        this.email = email;
        this.password = password;
    }

    public String getName() {
        return name;
    }
}

```

```

    }

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

    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;
    }
}

```

UserDatabaseOp.java

```

package com.railwaycrossing;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class UserDatabaseOp {
    private Connection connection;

    public UserDatabaseOp() {
        connection = DatabaseConnection.getConnection();
    }

    public void registerUser(User user) {
        try {
            String query = "INSERT INTO user_signup (name, email, password) VALUES (?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(query);
            statement.setString(1, user.getName());
            statement.setString(2, user.getEmail());
            statement.setString(3, user.getPassword());
            statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public boolean loginUser(String email, String password) {
        try {
            String query = "SELECT * FROM user_signup WHERE email=? AND password=?";
            PreparedStatement statement = connection.prepareStatement(query);
            statement.setString(1, email);
            statement.setString(2, password);
            ResultSet resultSet = statement.executeQuery();
            return resultSet.next();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
}

```

```

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.setName(resultSet.getString("name"));
            user.setEmail(resultSet.getString("email"));
            user.setPassword(resultSet.getString("password"));
            return user;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}
}

```

addRailCrossingForm.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<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(1, 1, 1, 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: 50%;
    padding: 10px;
}

```

```

        background-color: gray;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }

.panel button[type="submit"]:hover {
    background-color: #000000;
}
</style>
</head>
<body>
    <div class="panel">
        <h2>Railway Crossings</h2>
        <h4>Add Railway Crossing Information</h4>
        <form action="addCrossing" 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="trainSchedules">Train Schedules</label> <input type="text"
                id="trainSchedules" name="trainSchedules" required><br>
            <label for="personInCharge">Person in Charge</label> <input
                type="text" id="personInCharge" name="personInCharge" required><br>
            <label for="status">Crossing Status</label> <select id="status"
                name="status">
                <option value="Open">Open</option>
                <option value="Closed">Closed</option>
            </select><br><br>
            <button type="submit">Submit</button>
        </form>
    </div>
</body>
</html>

```

adminHome.jsp

```

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

<%

RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();

List<RailwayCrossing> crossings = crossingDAO.getAllCrossings();

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

if (searchId != null && !searchId.isEmpty()) {

    int crossingId = Integer.parseInt(searchId);

    RailwayCrossing searchedCrossing = crossingDAO.getCrossingById(crossingId);

    request.setAttribute("crossing", searchedCrossing);

}
%>
<!DOCTYPE html>
<html>
<head>

```



```

<meta charset="UTF-8">
<title>Admin Home Page</title>
<style>
/* Button Styles */
.button-container {
    display: inline-block;
    justify-content: flex-start;
    margin-bottom: 10px;
    margin-left: 1px;
}

.button-container form {
    margin-right: 100px;
}

.button-container input[type="text"] {
    margin-right: 5px;
    padding: 5px;
    border-radius: 4px;
    border: 1px solid #ccc;
    display: inline-block;
}

.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 {
    border-collapse: collapse;
    width: 100%;
}

th, td {
    text-align: left;
    padding: 8px;
    border-bottom: 1px solid #ddd;
}

th {
    background-color: #f2f2f2;
}

.inline {
    display: inline-block;
}

```

```

</style>
</head>
<body>
    <h1 style="font-family: Times New Roman;" align=center>Admin Home Page</h1>
    <h3>
        Number of Railway Crossing : <%=crossings.size()%>
    </h3>

    <button onclick="location.href='HomePage.jsp'" style="border-radius: 5px; font-weight:
bold; padding: 5px 10px;">
        Home</button>

    <button onclick="location.href='adminHome.jsp'"
        style="border-radius: 5px; margin-left: 50px; font-weight: bold; padding: 5px
10px;">
        Display All Railway Crossings</button>

    <button onclick="location.href='addRailCrossingForm.jsp'" style="border-radius: 5px;
margin-left: 50px; font-weight: bold; padding: 5px 10px;">Add Railway Crossing</button>

    <button onclick="location.href='adminLogin.jsp'"
        style="border-radius: 5px; margin-left: 700px; font-weight: bold; padding: 5px
10px;">Logout</button><br><br>

    <form action="searchCrossing" method="post">
        <button type="submit" style="border-radius: 5px; font-weight: bold;
padding: 5px 10px;">
            Search Crossing</button>
        <input type="text" name="searchId" placeholder="Enter ID" style="padding:
5px 10px" required>
    </form><br>

    <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) {
                RailwayCrossing crossing = (RailwayCrossing)
request.getAttribute("crossing");
            %>
            <tr>
                <td><%=crossing.getId()%></td>
                <td><%=crossing.getName()%></td>
                <td><%=crossing.getAddress()%></td>
                <td><%=crossing.getLandmark()%></td>
                <td><%=crossing.getTrainSchedule()%></td>
                <td><%=crossing.getPersonInCharge()%></td>
                <td><%=crossing.getStatus()%></td>
                <td></td>
            </tr>
            <%
            } else {

```

```

        for (RailwayCrossing anotherCrossing : crossings) {
        %>
        <tr>
            <td><%=anotherCrossing.getId()%></td>
            <td><%=anotherCrossing.getName()%></td>
            <td><%=anotherCrossing.getAddress()%></td>
            <td><%=anotherCrossing.getLandmark()%></td>
            <td><%=anotherCrossing.getTrainSchedule()%></td>
            <td><%=anotherCrossing.getPersonInCharge()%></td>
            <td><%=anotherCrossing.getStatus()%></td>
            <td>
                <form action="updateRailCrossingForm.jsp" method="post">
                    <input type="hidden" name="id"
                        value="<%=anotherCrossing.getId()%>">
                    <button type="submit" class="update-
button">Update</button>
                </form>
                <form action="deleteCrossing" method="post">
                    <input type="hidden" name="id"
                        value="<%=anotherCrossing.getId()%>">
                    <button type="submit" class="delete-
button">Delete</button>
                </form>
            </td>
        </tr>
        <%
        }
        %>
    </tbody>
</table>
</body>
</html>

```

adminLogin.jsp

```

<%@ 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;
    text-align: center;
    background-color: #f2f2f2;
    background-image: url("https://source.unsplash.com/1000x1000/?nature,mountain");
    background-repeat: no-repeat;
    background-size: cover;
    align-items: center;
    height: 90vh;
    background-color: #f2f2f2;
}

.panel {
    width: 500px;
    padding: 40px;
    opacity: 0.8;
    background-color: #ffffff;
    border-radius: 10px;
    box-shadow: 0 2px 5px rgba(1, 1, 1, 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: gray;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

.panel button[type="submit"]:hover {
    background-color: #000000;
}

.panel .signup-link {
    display: block;
    margin-top: 20px;
}
</style>
</head>
<body>
    <div class="panel">
        <h2>Railway Crossing</h2>
        <h4>Admin Login</h4>
        <form action="adminLogin" 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><br>
            <button type="submit">Login</button>
        </form><br>
        <form action="HomePage.jsp" method="post">
            <button type="submit">Home Page</button>
        </form>
    </div>
</body>
</html>

```

HomePage.jsp

```

<%@ 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>
<style>
body {
    background-image:url("https://source.unsplash.com/1000x1000/?nature,mountain");

```

```

        background-repeat: no-repeat;
        background-size: cover;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 90vh;
        text-align: center;
        background-color: #f2f2f2;
    }
    .panel {
        width: 500px;
        padding: 50px;
        background-color: white;
        opacity: 0.5;
        border-radius: 10px;
        box-shadow: 0 2px 5px rgba(100, 10, 10, 1000);
    }

    .panel button[type="submit"] {
        width: 60%;
        padding: 10px;
        background-color: black;

        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }

    .panel button[type="submit"]:hover {
        background-color: #000000;
        opacity: 0.9;
    }
</style>
</head>
<body>
    <div class="panel">
        <h2>Railway Crossings Status</h2>
        <h4>Developed by Yadav Dhiraj Rajendra Prasad</h4><br>

        <form action="adminLogin.jsp" method="post">
            <button type="submit">Government-Admin</button>
        </form><br>
        <form action="userLogin.jsp" method="post">
            <button type="submit">User</button>
        </form>

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

```

searchCrossing.jsp

```

<%@ 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="searchId">Search ID:</label> <input type="number"
    id="searchId" name="searchId" required> <input type="submit"
    value="Search">
</form>

<c:if test="${not empty crossing}">
  <h2>Search Result</h2>
  <table>
    <tr>
      <th>ID</th>
      <th>Name</th>
      <th>Address</th>
      <th>Landmark</th>
      <th>Train Schedule</th>
      <th>Person in Charge</th>
      <th>Status</th>
    </tr>
    <tr>
      <td>${crossing.id}</td>
      <td>${crossing.name}</td>
      <td>${crossing.address}</td>
      <td>${crossing.landmark}</td>
      <td>${crossing.trainSchedule}</td>
      <td>${crossing.personInCharge}</td>
      <td>${crossing.status}</td>
    </tr>
  </table>
</c:if>

<a href="adminHome.jsp">Back to Admin Homepage</a>
</body>
</html>

```

updateRailCrossingForm.jsp

```

<%@ 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">
            <label for="crossingId">Crossing ID</label> <input type="text"
                id="crossingId" name="crossingId" value="${crossing.crossingId}"
                required><br> <label for="name">Enter Name</label> <input
                type="text" id="name" name="name" value="${crossing.name}"
required><br>
            <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>

```

userHome.jsp

```

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

```

```

<%@ page import="com.railwaycrossing.RailwayCrossing"%>
<%@ page import="com.railwaycrossing.RailwayCrossingDatabaseOp"%>
<%@ 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 {
    width: 500px;
    border: 1px solid #ccc;
    border-radius: 10px;
    padding: 20px;
    margin-bottom: 20px;
    margin-top: 20px;
    box-shadow: 0 2px 5px rgba(0, 1, 1, 1);
}

.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: 15%;
    background-color: green;
}

.status-closed {
    color: white;
    border-radius: 15%;
    background-color: red;
}
</style>
</head>
<body>
    <h1 style="font-weight: bold; font-family: Times New Roman;" align=center>User Home
Page</h1>

```



```

        <button onclick="location.href='HomePage.jsp'" style="border-radius: 5px; font-weight:
bold; padding: 5px 10px;">Home</button>

        <button onclick="location.href='userHome.jsp'" style="border-radius: 5px; margin-left:
50px; font-weight: bold; padding: 5px 10px;">All
        Crossings</button>

        <button onclick="location.href='userHomeFavoriteCrossing.jsp'"
        style="border-radius: 5px; margin-left: 50px; font-weight: bold; padding: 5px
10px;">Favorite
        Crossings</button>

        <button onclick="location.href='userSearchCrossing.jsp'"
        style="border-radius: 5px; margin-left: 50px; font-weight: bold; padding: 5px
10px;">SearchCrossings</button>

        <button onclick="location.href='userLogin.jsp'"
        style="border-radius: 5px; margin-left: 600px; font-weight: bold; padding: 5px
10px;">Logout</button>

    <%
    RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();
    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">
        <h3 style="font-family: Times New Roman;"><%=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: <strong><%=crossing.getLandmark()%></strong>
        </p>
        <p>
            Address: <strong><%=crossing.getAddress()%></strong>
        </p>
        <form action="addToFavorites" method="post" style="display: inline;">
            <input type="hidden" name="crossingId" value="<%=crossing.getId()%>">
            <button type="submit"
                style="border-radius: 5px; font-weight: bold; margin-top: 10px;
margin-left: 0px; padding: 3px 7px;">ADD
                TO FAVORITES</button>
        </form>
    </div>
    <%
    }
    %>
</body>
</html>

```



```

        <%
        RailwayCrossingDatabaseOp crossingDAO = new RailwayCrossingDatabaseOp();
        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 style="font-family: Times New Roman;"><%=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: 5px; font-weight: bold; margin-top:
10px; padding: 5px 10px;">REMOVE
                    FROM FAVORITES</button>
            </form>
        </div>
    <%
    }
    %>
</div>
</body>
</html>

```

userLogin.jsp

```

<%@ 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;
    text-align: center;
    background-color: #f2f2f2;
    background-image: url("https://source.unsplash.com/1000x1000/?nature,mountain");
    background-repeat: no-repeat;
    background-size: cover;
    align-items: center;
    height: 90vh;

```

```

        background-color: #f2f2f2;
    }

    .panel {
        width: 500px;
        padding: 40px;
        opacity: 0.9;
        background-color: #ffffff;
        border-radius: 10px;
        box-shadow: 0 2px 5px rgba(1, 1, 1, 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: gray;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }

    .panel button[type="submit"]:hover {
        background-color: #000000;
    }

    .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><br>
            <button type="submit">Login</button>
        </form><br>
        <form action="HomePage.jsp" method="post">
            <button type="submit">Home Page</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>
```

userRegister.jsp

```
<%@ 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;
    text-align: center;
    background-color: #f2f2f2;
    background-image: url("https://source.unsplash.com/1000x1000/?nature,mountain");
    background-repeat: no-repeat;
    background-size: cover;
    align-items: center;
    height: 90vh;
    background-color: #f2f2f2;
}

.panel {
    width: 500px;
    padding: 40px;
    opacity: 0.9;
    background-color: #ffffff;
    border-radius: 10px;
    box-shadow: 0 2px 5px rgba(1, 1, 1, 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: gray;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

.panel button[type="submit"]:hover {
    background-color: #000000;
}

.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><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>

```

userSearchCrossing.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="com.railwaycrossing.RailwayCrossing"%>
<%@ page import="com.railwaycrossing.RailwayCrossingDatabaseOp"%>
<%@ 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 {
    width: 500px;
    border: 1px solid #ccc;
    border-radius: 10px;
    padding: 20px;
    margin-bottom: 20px;
    margin-top: 20px;
    box-shadow: 0 2px 5px rgba(0, 1, 1, 1);
}

.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;
  margin-top: 20px;
  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: 15%;
  background-color: green;
}

.status-closed {
  color: white;
  border-radius: 15%;
  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 {

```



```
                                TO FAVORITES</button>
                                </form>
                            </div>
                        <%
                        }
                        %>
                    </div>
</body>
</html>
```

[GITHUB Linke : https://github.com/dhiraj0803/JAVA-fsd.git](https://github.com/dhiraj0803/JAVA-fsd.git)