**User.java**

**package** com.example.railways;

**public** **class** User {

**private** **int** id;

**private** String username;

**private** String email;

**private** String password;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**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;

}

}

**RailwayCrossing.java**

**package** com.example.railways;

**public** **class** RailwayCrossing {

**private** **int** id;

**private** String name;

**private** String address;

**private** String landmark;

**private** String 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 getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

}

**DatabaseUtil.java**

**package** com.example.railways;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** DatabaseUtil {

**private** **static** **final** String ***DB\_URL*** = "jdbc:mysql://localhost:3306/railway\_app";

**private** **static** **final** String ***DB\_USERNAME*** = "root";

**private** **static** **final** String ***DB\_PASSWORD*** = "P6ya5mx@5132001";

**public** **static** Connection getConnection() **throws** SQLException {

**return** DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway\_app, root, P6ya5mx@5132001");

}

**public** **static** **void** closeConnection(Connection conn) {

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

**try** {

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

**UserDAO.java**

package com.example.railways;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class UserDAO {

public boolean registerUser(User user) {

try (Connection conn = DatabaseUtil.getConnection();

PreparedStatement stmt = conn.prepareStatement(

"INSERT INTO users (username, email, password) VALUES (?, ?, ?)")) {

stmt.setString(1, user.getUsername());

stmt.setString(2, user.getEmail());

stmt.setString(3, user.getPassword());

int rowsAffected = stmt.executeUpdate();

return rowsAffected > 0;

} catch (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

return false;

}

public User loginUser(String username, String password) {

try (Connection conn = DatabaseUtil.getConnection();

PreparedStatement stmt = conn.prepareStatement(

"SELECT \* FROM users WHERE username = ? AND password = ?")) {

stmt.setString(1, username);

stmt.setString(2, password);

try (ResultSet rs = stmt.executeQuery()) {

if (rs.next()) {

User user = new User();

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

user.setUsername(rs.getString("username"));

user.setEmail(rs.getString("email"));

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

return user;

}

}

} catch (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

return null;

}

public User getUserById(int userId) {

try (Connection conn = DatabaseUtil.getConnection();

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM users WHERE id = ?")) {

stmt.setInt(1, userId);

try (ResultSet rs = stmt.executeQuery()) {

if (rs.next()) {

User user = new User();

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

user.setUsername(rs.getString("username"));

user.setEmail(rs.getString("email"));

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

return user;

}

}

} catch (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

return null;

}

// Implement other user-related methods as needed

}

**RailwayCrossingDAO.java**

**package** com.example.railways;

**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 {

**public** **boolean** addRailwayCrossing(RailwayCrossing crossing) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement(

"INSERT INTO railway\_crossings (name, address, landmark, status) VALUES (?, ?, ?, ?)")) {

stmt.setString(1, crossing.getName());

stmt.setString(2, crossing.getAddress());

stmt.setString(3, crossing.getLandmark());

stmt.setString(4, crossing.getStatus());

**int** rowsAffected = stmt.executeUpdate();

**return** rowsAffected > 0;

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

**public** List<RailwayCrossing> getAllRailwayCrossings() {

List<RailwayCrossing> crossings = **new** ArrayList<>();

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM railway\_crossings");

ResultSet rs = stmt.executeQuery()) {

**while** (rs.next()) {

RailwayCrossing crossing = **new** RailwayCrossing();

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

crossing.setName(rs.getString("name"));

crossing.setAddress(rs.getString("address"));

crossing.setLandmark(rs.getString("landmark"));

crossing.setStatus(rs.getString("status"));

crossings.add(crossing);

}

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** crossings;

}

**public** **boolean** updateRailwayCrossing(RailwayCrossing crossing) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement(

"UPDATE railway\_crossings SET name = ?, address = ?, landmark = ?, status = ? WHERE id = ?")) {

stmt.setString(1, crossing.getName());

stmt.setString(2, crossing.getAddress());

stmt.setString(3, crossing.getLandmark());

stmt.setString(4, crossing.getStatus());

stmt.setInt(5, crossing.getId());

**int** rowsAffected = stmt.executeUpdate();

**return** rowsAffected > 0;

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

**public** **boolean** deleteRailwayCrossing(**int** crossingId) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement("DELETE FROM railway\_crossings WHERE id = ?")) {

stmt.setInt(1, crossingId);

**int** rowsAffected = stmt.executeUpdate();

**return** rowsAffected > 0;

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

}

**FavoriteCrossingDAO.java**

**package** com.example.railways;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**import** java.sql.ResultSet;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** FavoriteCrossingDAO {

**public** **boolean** addFavoriteCrossing(**int** userId, **int** crossingId) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement(

"INSERT INTO favorites (user\_id, crossing\_id) VALUES (?, ?)")) {

stmt.setInt(1, userId);

stmt.setInt(2, crossingId);

**int** rowsAffected = stmt.executeUpdate();

**return** rowsAffected > 0;

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

**public** **boolean** removeFavoriteCrossing(**int** userId, **int** crossingId) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement(

"DELETE FROM favorites WHERE user\_id = ? AND crossing\_id = ?")) {

stmt.setInt(1, userId);

stmt.setInt(2, crossingId);

**int** rowsAffected = stmt.executeUpdate();

**return** rowsAffected > 0;

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

**public** List<RailwayCrossing> getFavoriteCrossingsByUserId(**int** userId) {

List<RailwayCrossing> favoriteCrossings = **new** ArrayList<>();

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement(

"SELECT rc.\* FROM railway\_crossings rc INNER JOIN favorites f ON rc.id = f.crossing\_id WHERE f.user\_id = ?")) {

stmt.setInt(1, userId);

**try** (ResultSet rs = stmt.executeQuery()) {

**while** (rs.next()) {

RailwayCrossing crossing = **new** RailwayCrossing();

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

crossing.setName(rs.getString("name"));

crossing.setAddress(rs.getString("address"));

crossing.setLandmark(rs.getString("landmark"));

crossing.setStatus(rs.getString("status"));

favoriteCrossings.add(crossing);

}

}

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** favoriteCrossings;

}

}

**RegisterServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

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

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

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

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

**import** java.security.MessageDigest;

**import** java.security.NoSuchAlgorithmException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

@WebServlet("/register")

**public** **class** RegisterServlet **extends** HttpServlet {

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

**private** **static** **final** String ***DB\_URL*** = "jdbc:mysql://localhost:3306/railway\_app";

**private** **static** **final** String ***DB\_USERNAME*** = "root";

**private** **static** **final** String ***DB\_PASSWORD*** = "P6ya5mx@5132001";

**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");

// Hash the password for security

String hashedPassword = hashPassword(password);

// Store the user in the database

**try** {

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway\_app, root, P6ya5mx@5132001");

PreparedStatement stmt = conn.prepareStatement("INSERT INTO users (name, email, password) VALUES (?, ?, ?)");

stmt.setString(1, name);

stmt.setString(2, email);

stmt.setString(3, hashedPassword);

stmt.executeUpdate();

stmt.close();

conn.close();

response.sendRedirect("login.jsp");

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

// Redirect or show an error message to the user

}

}

**private** String hashPassword(String password) {

**try** {

MessageDigest md = MessageDigest.*getInstance*("SHA-256");

**byte**[] hashedBytes = md.digest(password.getBytes());

// Convert the byte array to hexadecimal string

StringBuilder sb = **new** StringBuilder();

**for** (**byte** b : hashedBytes) {

sb.append(String.*format*("%02x", b));

}

**return** sb.toString();

} **catch** (NoSuchAlgorithmException e) {

e.printStackTrace();

// Handle any errors related to password hashing

// Return an empty string or throw an exception

}

**return** "";

}

}

**LoginServlet.java**

**package** com.example.railways;

**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;

**import** java.security.MessageDigest;

**import** java.security.NoSuchAlgorithmException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

@WebServlet("/login")

**public** **class** LoginServlet **extends** HttpServlet {

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

**private** **static** **final** String ***DB\_URL*** = "jdbc:mysql://localhost:3306/railway\_app";

**private** **static** **final** String ***DB\_USERNAME*** = "root";

**private** **static** **final** String ***DB\_PASSWORD*** = "P6ya5mx@5132001";

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

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

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

// Hash the password for comparison

String hashedPassword = hashPassword(password);

// Check if the user exists in the database

**try** {

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway\_app, root, P6ya5mx@5132001");

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM users WHERE email = ? AND password = ?");

stmt.setString(1, email);

stmt.setString(2, hashedPassword);

ResultSet rs = stmt.executeQuery();

**if** (rs.next()) {

// User login successful

HttpSession session = request.getSession();

session.setAttribute("email", email);

response.sendRedirect("dashboard.jsp"); // Redirect to the user's dashboard

} **else** {

// Invalid email or password

response.sendRedirect("login.jsp"); // Redirect back to the login page

}

rs.close();

stmt.close();

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

// Redirect or show an error message to the user

}

}

**private** String hashPassword(String password) {

**try** {

MessageDigest md = MessageDigest.*getInstance*("SHA-256");

**byte**[] hashedBytes = md.digest(password.getBytes());

// Convert the byte array to hexadecimal string

StringBuilder sb = **new** StringBuilder();

**for** (**byte** b : hashedBytes) {

sb.append(String.*format*("%02x", b));

}

**return** sb.toString();

} **catch** (NoSuchAlgorithmException e) {

e.printStackTrace();

// Handle any errors related to password hashing

// Return an empty string or throw an exception

}

**return** "";

}

}

**AdminLoginServlet.java**

**package** com.example.railways;

**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("/adminLogin")

**public** **class** AdminLoginServlet **extends** HttpServlet {

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

**private** **static** **final** String ***ADMIN\_USERNAME*** = "admin";

**private** **static** **final** String ***ADMIN\_PASSWORD*** = "admin";

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

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

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

**if** (username.equals(***ADMIN\_USERNAME***) && password.equals(***ADMIN\_PASSWORD***)) {

// Admin authentication successful

HttpSession session = request.getSession();

session.setAttribute("isAdminLoggedIn", **true**);

response.sendRedirect("adminDashboard.jsp");

} **else** {

// Admin authentication failed

response.sendRedirect("adminLogin.jsp?error=true");

}

}

}

**AdminDashboardServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** javax.servlet.ServletException;

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

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

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

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

@WebServlet("/adminDashboard")

**public** **class** AdminDashboardServlet **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");

**if** (authenticateAdmin(email, password)) {

request.getRequestDispatcher("adminDashboard.jsp").forward(request, response);

} **else** {

response.sendRedirect("index.jsp?adminError=true");

}

}

**private** **boolean** authenticateAdmin(String email, String password) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM admin WHERE email = ?")) {

stmt.setString(1, email);

**try** (ResultSet rs = stmt.executeQuery()) {

**if** (rs.next()) {

String storedPassword = rs.getString("password");

// Here, we compare the plain text password for simplicity

**if** (password.equals(storedPassword)) {

**return** **true**;

}

}

}

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

}

**FavoriteCrossingsServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** javax.servlet.ServletException;

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

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

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

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

@WebServlet("/adminDashboard")

**public** **class** AdminDashboardServlet **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");

**if** (authenticateAdmin(email, password)) {

request.getRequestDispatcher("adminDashboard.jsp").forward(request, response);

} **else** {

response.sendRedirect("index.jsp?adminError=true");

}

}

**private** **boolean** authenticateAdmin(String email, String password) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM admin WHERE email = ?")) {

stmt.setString(1, email);

**try** (ResultSet rs = stmt.executeQuery()) {

**if** (rs.next()) {

String storedPassword = rs.getString("password");

// Here, we compare the plain text password for simplicity

**if** (password.equals(storedPassword)) {

**return** **true**;

}

}

}

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

}

**SearchServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**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("/search")

**public** **class** SearchServlet **extends** HttpServlet {

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

**private** **static** **final** String ***DB\_URL*** = "jdbc:mysql://localhost:3306/railway\_app";

**private** **static** **final** String ***DB\_USERNAME*** = "root";

**private** **static** **final** String ***DB\_PASSWORD*** = "P6ya5mx@5132001";

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

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

// Perform search query on the database

List<RailwayCrossing> searchResults = searchRailwayCrossings(query);

request.setAttribute("searchResults", searchResults);

request.getRequestDispatcher("searchResults.jsp").forward(request, response);

}

**private** List<RailwayCrossing> searchRailwayCrossings(String query) {

List<RailwayCrossing> searchResults = **new** ArrayList<>();

**try** {

Connection conn = DriverManager.*getConnection*(***DB\_URL***, ***DB\_USERNAME***, ***DB\_PASSWORD***);

PreparedStatement stmt = conn

.prepareStatement("SELECT \* FROM railway\_crossings WHERE name LIKE ?");

stmt.setString(1, "%" + query + "%");

ResultSet rs = stmt.executeQuery();

**while** (rs.next()) {

RailwayCrossing crossing = **new** RailwayCrossing();

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

crossing.setName(rs.getString("name"));

crossing.setAddress(rs.getString("address"));

crossing.setLandmark(rs.getString("landmark"));

crossing.setStatus(rs.getString("status"));

searchResults.add(crossing);

}

rs.close();

stmt.close();

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

// Redirect or show an error message to the user

}

**return** searchResults;

}

}

**ManageCrossingServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**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("/manageCrossings")

**public** **class** ManageCrossingServlet **extends** HttpServlet {

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

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

List<RailwayCrossing> crossings = getAllCrossings();

request.setAttribute("crossings", crossings);

request.getRequestDispatcher("manageCrossings.jsp").forward(request, response);

}

**private** List<RailwayCrossing> getAllCrossings() {

List<RailwayCrossing> crossings = **new** ArrayList<>();

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM railway\_crossings");

ResultSet rs = stmt.executeQuery()) {

**while** (rs.next()) {

RailwayCrossing crossing = **new** RailwayCrossing();

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

crossing.setName(rs.getString("name"));

crossing.setAddress(rs.getString("address"));

crossing.setLandmark(rs.getString("landmark"));

crossing.setStatus(rs.getString("status"));

crossings.add(crossing);

}

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** crossings;

}

}

**AddCrossingServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**import** javax.servlet.ServletException;

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

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

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

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

@WebServlet("/addCrossing")

**public** **class** AddCrossingServlet **extends** HttpServlet {

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

**private** **static** **final** String ***DB\_URL*** = "jdbc:mysql://localhost:3306/railway\_app";

**private** **static** **final** String ***DB\_USERNAME*** = "root";

**private** **static** **final** String ***DB\_PASSWORD*** = "P6ya5mx@5132001";

**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 status = request.getParameter("status");

// Insert the crossing into the database

addCrossing(name, address, landmark, status);

response.sendRedirect("adminDashboard.jsp");

}

**private** **void** addCrossing(String name, String address, String landmark, String status) {

**try** {

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway\_app, root, P6ya5mx@5132001");

PreparedStatement stmt = conn.prepareStatement(

"INSERT INTO railway\_crossings (name, address, landmark, status) VALUES (?, ?, ?, ?)");

stmt.setString(1, name);

stmt.setString(2, address);

stmt.setString(3, landmark);

stmt.setString(4, status);

stmt.executeUpdate();

stmt.close();

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

// Redirect or show an error message to the user

}

}

}

**DeleteCrossingServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**import** javax.servlet.ServletException;

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

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

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

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

@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("crossingId"));

**if** (deleteCrossing(crossingId)) {

response.sendRedirect("deleteCrossings.jsp?success=true");

} **else** {

response.sendRedirect("deleteCrossings.jsp?error=true");

}

}

**private** **boolean** deleteCrossing(**int** crossingId) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement("DELETE FROM railway\_crossings WHERE id = ?")) {

stmt.setInt(1, crossingId);

**int** rowsAffected = stmt.executeUpdate();

**return** rowsAffected > 0;

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

}

**UpdateCrossingServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** javax.servlet.ServletException;

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

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

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

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

@WebServlet("/updateCrossings")

**public** **class** UpdateCrossingServlet **extends** HttpServlet {

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

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

**throws** ServletException, IOException {

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

RailwayCrossing crossing = getCrossingById(crossingId);

request.setAttribute("crossing", crossing);

request.getRequestDispatcher("updateCrossings.jsp").forward(request, response);

}

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

**throws** ServletException, IOException {

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

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

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

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

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

RailwayCrossing crossing = **new** RailwayCrossing();

crossing.setId(crossingId);

crossing.setName(name);

crossing.setAddress(address);

crossing.setLandmark(landmark);

crossing.setStatus(status);

**if** (updateCrossing(crossing)) {

response.sendRedirect("manageCrossings");

} **else** {

response.sendRedirect("updateCrossings?id=" + crossingId + "&error=true");

}

}

**private** RailwayCrossing getCrossingById(**int** crossingId) {

RailwayCrossing crossing = **null**;

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM railway\_crossings WHERE id = ?")) {

stmt.setInt(1, crossingId);

**try** (ResultSet rs = stmt.executeQuery()) {

**if** (rs.next()) {

crossing = **new** RailwayCrossing();

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

crossing.setName(rs.getString("name"));

crossing.setAddress(rs.getString("address"));

crossing.setLandmark(rs.getString("landmark"));

crossing.setStatus(rs.getString("status"));

}

}

}

**catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** crossing;

}

**private** **boolean** updateCrossing(RailwayCrossing crossing) {

**try** (Connection conn = DatabaseUtil.*getConnection*();

PreparedStatement stmt = conn.prepareStatement(

"UPDATE railway\_crossings SET name = ?, address = ?, landmark = ?, status = ? WHERE id = ?")) {

stmt.setString(1, crossing.getName());

stmt.setString(2, crossing.getAddress());

stmt.setString(3, crossing.getLandmark());

stmt.setString(4, crossing.getStatus());

stmt.setInt(5, crossing.getId());

**int** rowsAffected = stmt.executeUpdate();

**return** rowsAffected > 0;

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

}

**return** **false**;

}

}

**RailwayCrossingsServlet.java**

**package** com.example.railways;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**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("/railwayCrossings")

**public** **class** RailwayCrossingsServlet **extends** HttpServlet {

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

**private** **static** **final** String ***DB\_URL*** = "jdbc:mysql://localhost:3306/railway\_app";

**private** **static** **final** String ***DB\_USERNAME*** = "root";

**private** **static** **final** String ***DB\_PASSWORD*** = "P6ya5mx@5132001";

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

List<RailwayCrossing> crossings = **new** ArrayList<>();

**try** {

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway\_app, root, P6ya5mx@5132001");

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM railway\_crossings");

ResultSet rs = stmt.executeQuery();

**while** (rs.next()) {

RailwayCrossing crossing = **new** RailwayCrossing();

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

crossing.setName(rs.getString("name"));

crossing.setAddress(rs.getString("address"));

crossing.setLandmark(rs.getString("landmark"));

crossing.setStatus(rs.getString("status"));

crossings.add(crossing);

}

rs.close();

stmt.close();

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

// Redirect or show an error message to the user

}

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

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

**try** {

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway\_app, root, P6ya5mx@5132001");

PreparedStatement stmt = conn.prepareStatement("UPDATE railway\_crossings SET status = ? WHERE id = ?");

stmt.setString(1, newStatus);

stmt.setString(2, crossingId);

stmt.executeUpdate();

stmt.close();

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

// Handle any errors that occur during database interaction

// Redirect or show an error message to the user

}

request.setAttribute("crossings", crossings);

request.getRequestDispatcher("railwayCrossings.jsp").forward(request, response);

}

}

**Hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection properties -->

<property name=*"hibernate.connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/your\_database\_name</property>

<property name=*"hibernate.connection.username"*>your\_username</property>

<property name=*"hibernate.connection.password"*>your\_password</property>

<!-- Hibernate dialect for MySQL -->

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL8Dialect</property>

<!-- Enable Hibernate's automatic session context management -->

<property name=*"hibernate.current\_session\_context\_class"*>thread</property>

<!-- Show SQL statements in console for debugging (optional) -->

<property name=*"hibernate.show\_sql"*>true</property>

<!-- Update the database schema automatically (optional) -->

<property name=*"hibernate.hbm2ddl.auto"*>update</property>

<!-- Mapping files -->

<mapping class=*"com.example.model.RailwayCrossing"*/>

<!-- Add more mapping classes here if needed -->

</session-factory>

</hibernate-configuration>

**Register.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>User Registration</title>

</head>

<body>

<h1>User Registration</h1>

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

<label for=*"name"*>Name:</label>

<input type=*"text"* id=*"name"* name=*"name"* required><br><br>

<label for=*"email"*>Email:</label>

<input type=*"email"* id=*"email"* name=*"email"* required><br><br>

<label for=*"password"*>Password:</label>

<input type=*"password"* id=*"password"* name=*"password"* required><br><br>

<input type=*"submit"* value=*"Register"*>

</form>

</body>

</html>

**Login.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>User Login</title>

</head>

<body>

<h1>User Login</h1>

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

<label for=*"email"*>Email:</label>

<input type=*"email"* id=*"email"* name=*"email"* required><br><br>

<label for=*"password"*>Password:</label>

<input type=*"password"* id=*"password"* name=*"password"* required><br><br>

<input type=*"submit"* value=*"Login"*>

</form>

</body>

</html>

**AdminLogin.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin Login</title>

</head>

<body>

<h1>Admin Login</h1>

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

<input type=*"text"* name=*"username"* placeholder=*"Username"* required><br>

<input type=*"password"* name=*"password"* placeholder=*"Password"* required><br>

<input type=*"submit"* value=*"Log in"*>

</form>

</body>

</html>

**adminDashboard.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ page session=*"false"* %>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<c:if test=*"*${sessionScope.isAdminLoggedIn **eq true**}*"*>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin Dashboard</title>

</head>

<body>

<h1>Welcome, Admin</h1>

<ul>

<li><a href=*"addCrossing.jsp"*>Add Crossing</a></li>

<li><a href=*"manageCrossings.jsp"*>Manage Crossings</a></li>

<li><a href=*"logout"*>Log out</a></li>

</ul>

</body>

</html>

</c:if>

**railwayCrossings.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Railway Crossings</title>

</head>

<body>

<body>

<h1>Railway Crossings</h1>

<table>

<thead>

<tr>

<th>Name</th>

<th>Address</th>

<th>Landmark</th>

<th>Status</th>

</tr>

</thead>

<tbody>

<c:forEach var=*"crossing"* items=*"*${crossings}*"*>

<tr>

<td>${crossing.name}</td>

<td>${crossing.address}</td>

<td>${crossing.landmark}</td>

<td>${crossing.status}</td>

</tr>

</c:forEach>

</tbody>

</table>

</body>

</html>

**manageCrossings.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ page session=*"false"* %>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>Manage Crossings</h1>

<table>

<tr>

<th>Name</th>

<th>Address</th>

<th>Landmark</th>

<th>Status</th>

<th>Action</th>

</tr>

<c:forEach items=*"*${crossings}*"* var=*"crossing"*>

<tr>

<td>${crossing.name}</td>

<td>${crossing.address}</td>

<td>${crossing.landmark}</td>

<td>${crossing.status}</td>

<td>

<a href=*"updateCrossings?id=*${crossing.id}*"*>Edit</a>

</td>

</tr>

</c:forEach>

</table>

</body>

</html>

**searchResults.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Railway Crossing Search Results</title>

</head>

<body>

<h1>Railway Crossing Search Results</h1>

<table>

<thead>

<tr>

<th>Name</th>

<th>Address</th>

<th>Landmark</th>

<th>Status</th>

</tr>

</thead>

<tbody>

<c:forEach var=*"crossing"* items=*"*${searchResults}*"*>

<tr>

<td>${crossing.name}</td>

<td>${crossing.address}</td>

<td>${crossing.landmark}</td>

<td>${crossing.status}</td>

</tr>

</c:forEach>

</tbody>

</table>

</body>

</html>

**addCrossings.java**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Add Crossing</title>

</head>

<body>

<h1>Add Crossing</h1>

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

<!-- Input fields for crossing details -->

<input type=*"text"* name=*"name"* placeholder=*"Name"* required><br>

<input type=*"text"* name=*"address"* placeholder=*"Address"* required><br>

<input type=*"text"* name=*"landmark"* placeholder=*"Landmark"* required><br>

<input type=*"text"* name=*"status"* placeholder=*"Status"* required><br>

<input type=*"submit"* value=*"Add Crossing"*>

</form>

</body>

</html>

**favoriteCrossings.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ page session=*"false"* %>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>Favorite Crossings</h1>

<c:if test=*"*${**empty** favoriteCrossings}*"*>

<p>No favorite crossings found.</p>

</c:if>

<c:if test=*"*${**not empty** favoriteCrossings}*"*>

<table>

<tr>

<th>Name</th>

<th>Address</th>

<th>Landmark</th>

<th>Status</th>

</tr>

<c:forEach items=*"*${favoriteCrossings}*"* var=*"crossing"*>

<tr>

<td>${crossing.name}</td>

<td>${crossing.address}</td>

<td>${crossing.landmark}</td>

<td>${crossing.status}</td>

</tr>

</c:forEach>

</table>

</c:if>

</body>

</html>

**deleteCrossing.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>Delete Crossings</h1>

<c:if test="${param.success == '**true**'}">

<p>Successfully deleted the crossing.</p>

</c:if>

<c:if test="${param.error == '**true**'}">

<p>Error occurred while deleting the crossing.</p>

</c:if>

<table>

<tr>

<th>Name</th>

<th>Address</th>

<th>Landmark</th>

<th>Status</th>

<th>Action</th>

</tr>

<c:forEach items="${crossings}" var=*"crossing"*>

<tr>

<td>${crossing.name}</td>

<td>${crossing.address}</td>

<td>${crossing.landmark}</td>

<td>${crossing.status}</td>

<td>

<form action=*"deleteCrossing"* method=*"post"*>

<input type=*"hidden"* name=*"crossingId"* value=*"*${crossing.id}*"*>

<button type=*"submit"*>Delete</button>

</form>

</td>

</tr>

</c:forEach>

</table>

</body>

</html>

**updateCrossings.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ page session=*"false"* %>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Update Crossing</title>

</head>

<body>

<h1>Update Crossing</h1>

<c:if test=*"*${param.error == '**true**'}*"*>

<p>Error occurred while updating the crossing.</p>

</c:if>

<form action=*"updateCrossings"* method=*"post"*>

<input type=*"hidden"* name=*"id"* value=*"*${crossing.id}*"*>

<label>Name:</label>

<input type=*"text"* name=*"name"* value=*"*${crossing.name}*"* required>

<br>

<label>Address:</label>

<input type=*"text"* name=*"address"* value=*"*${crossing.address}*"* required>

<br>

<label>Landmark:</label>

<input type=*"text"* name=*"landmark"* value=*"*${crossing.landmark}*"* required>

<br>

<label>Status:</label>

<input type=*"text"* name=*"status"* value=*"*${crossing.status}*"* required>

<br>

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

</form>

</body>

</html>

**Index.jsp**

<html>

<head>

<title>Railway Crossing Application</title>

</head>

<body>

<h1>Welcome to Railway Crossing Application</h1>

<p>Developer: Java FSD</p>

<h2>User Interface</h2>

<p>Select an option:</p>

<ul>

<li><a href=*"register.jsp"*>Register</a></li>

<li><a href=*"login.jsp"*>Login</a></li>

</ul>

<h2>Administrator Interface</h2>

<form action=*"adminLogin"* method=*"post"*>

<label>Email:</label>

<input type=*"email"* name=*"email"* required>

<br>

<label>Password:</label>

<input type=*"password"* name=*"password"* required>

<br>

<button type=*"submit"*>Login as Administrator</button>

</form>

</body>

</html>

**Web.xml**

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Archetype Created Web Application</display-name>

<servlet>

<servlet-name>RegisterServlet</servlet-name>

<display-name>RegisterServlet</display-name>

<description></description>

<servlet-class>com.example.railways.RegisterServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>LoginServlet</servlet-name>

<display-name>LoginServlet</display-name>

<description></description>

<servlet-class>com.example.railways.LoginServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>RailwayCrossingsServlet</servlet-name>

<display-name>RailwayCrossingsServlet</display-name>

<description></description>

<servlet-class>com.example.railways.RailwayCrossingsServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>SearchServlet</servlet-name>

<display-name>SearchServlet</display-name>

<description></description>

<servlet-class>com.example.railways.SearchServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>FavoriteCrossingsServlet</servlet-name>

<display-name>FavoriteCrossingsServlet</display-name>

<description></description>

<servlet-class>com.example.railways.FavoriteCrossingsServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminLoginServlet</servlet-name>

<display-name>AdminLoginServlet</display-name>

<description></description>

<servlet-class>com.example.railways.AdminLoginServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>AddCrossingServlet</servlet-name>

<display-name>AddCrossingServlet</display-name>

<description></description>

<servlet-class>com.example.railways.AddCrossingServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>LogoutServlet</servlet-name>

<display-name>LogoutServlet</display-name>

<description></description>

<servlet-class>com.example.railways.LogoutServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>DeleteCrossingServlet</servlet-name>

<display-name>DeleteCrossingServlet</display-name>

<description></description>

<servlet-class>com.example.railways.DeleteCrossingServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>ManageCrossingServlet</servlet-name>

<display-name>ManageCrossingServlet</display-name>

<description></description>

<servlet-class>com.example.railways.ManageCrossingServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>UpdateCrossingServlet</servlet-name>

<display-name>UpdateCrossingServlet</display-name>

<description></description>

<servlet-class>com.example.railways.UpdateCrossingServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminDashboardServlet</servlet-name>

<display-name>AdminDashboardServlet</display-name>

<description></description>

<servlet-class>com.example.railways.AdminDashboardServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>RegisterServlet</servlet-name>

<url-pattern>/RegisterServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>LoginServlet</servlet-name>

<url-pattern>/LoginServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>RailwayCrossingsServlet</servlet-name>

<url-pattern>/RailwayCrossingsServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>SearchServlet</servlet-name>

<url-pattern>/SearchServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>FavoriteCrossingsServlet</servlet-name>

<url-pattern>/FavoriteCrossingsServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminLoginServlet</servlet-name>

<url-pattern>/AdminLoginServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AddCrossingServlet</servlet-name>

<url-pattern>/AddCrossingServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>LogoutServlet</servlet-name>

<url-pattern>/LogoutServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>DeleteCrossingServlet</servlet-name>

<url-pattern>/DeleteCrossingServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>ManageCrossingServlet</servlet-name>

<url-pattern>/ManageCrossingServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>UpdateCrossingServlet</servlet-name>

<url-pattern>/UpdateCrossingServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminDashboardServlet</servlet-name>

<url-pattern>/AdminDashboardServlet</url-pattern>

</servlet-mapping>

</web-app>

**Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.railways</groupId>

<artifactId>RailwayCrossingApplication</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>RailwayCrossingApplication Maven Webapp</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>6.2.5.Final</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>RailwayCrossingApplication</finalName>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>

**Railway\_app database schema:**

**CREATE DATABASE railway\_app;**

**USE railway\_app;**

**CREATE TABLE users (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**name VARCHAR(255) NOT NULL,**

**email VARCHAR(255) NOT NULL UNIQUE,**

**password VARCHAR(255) NOT NULL**

**);**

**CREATE TABLE railway\_crossings (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**name VARCHAR(255) NOT NULL,**

**address VARCHAR(255) NOT NULL,**

**landmark VARCHAR(255),**

**train\_schedules TEXT,**

**person\_in\_charge VARCHAR(255),**

**status ENUM('open', 'closed') DEFAULT 'open'**

**);**

**CREATE TABLE favorites (**

**user\_id INT,**

**crossing\_id INT,**

**FOREIGN KEY (user\_id) REFERENCES users(id),**

**FOREIGN KEY (crossing\_id) REFERENCES railway\_crossings(id),**

**PRIMARY KEY (user\_id, crossing\_id)**

**);**