**Objective :** Create a User Management System with the following Features using MySQL database

1) Admin Users can perform add update delete and view users  
 2) End users can perform login and view their details

**CODE :**

**Main.java**

package org.example;

import java.util.List;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

UserService userService = new UserService();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\nUser Management System");

System.out.println("1. Login");

System.out.println("2. Exit");

System.out.print("Choose an option: ");

int option = scanner.nextInt();

scanner.nextLine();

if (option == 2) break;

System.out.print("Enter Username: ");

String username = scanner.nextLine();

System.out.print("Enter Password: ");

String password = scanner.nextLine();

User user = userService.loginUser(username, password);

if (user == null) {

System.out.println("Invalid Credentials!");

continue;

}

if (user.getRole().equals("admin")) {

while (true) {

System.out.println("\nAdmin Panel");

System.out.println("1. Add User");

System.out.println("2. Update User");

System.out.println("3. Delete User");

System.out.println("4. View Users");

System.out.println("5. Logout");

System.out.print("Choose an option: ");

int adminOption = scanner.nextInt();

scanner.nextLine();

if (adminOption == 5) break;

switch (adminOption) {

case 1:

System.out.print("Enter Username: ");

String newUser = scanner.nextLine();

System.out.print("Enter Password: ");

String newPassword = scanner.nextLine();

System.out.print("Enter Role (admin/user): ");

String role = scanner.nextLine();

userService.addUser(newUser, newPassword, role);

break;

case 2:

System.out.print("Enter User ID: ");

int userId = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter New Username: ");

String updatedUser = scanner.nextLine();

System.out.print("Enter New Password: ");

String updatedPass = scanner.nextLine();

System.out.print("Enter New Role: ");

String updatedRole = scanner.nextLine();

userService.updateUser(userId, updatedUser, updatedPass, updatedRole);

break;

case 3:

System.out.print("Enter User ID: ");

int deleteId = scanner.nextInt();

userService.deleteUser(deleteId);

break;

case 4:

List<User> users = userService.getAllUsers();

users.forEach(System.out::println);

break;

}

}

} else {

System.out.println("Welcome, " + user.getUsername() + "! Your details: " + user);

}

}

scanner.close();

}

}

**USER.java**

package org.example;

public class User {

private int id;

private String username;

private String password;

private String role;

public User(int id, String username, String password, String role) {

this.id = id;

this.username = username;

this.password = password;

this.role = role;

}

public int getId() { return id; }

public String getUsername() { return username; }

public String getPassword() { return password; }

public String getRole() { return role; }

@Override

public String toString() {

return "ID: " + id + ", Username: " + username + ", Role: " + role;

}

}

**UserService.java**

package org.example;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class UserService {

private Connection conn;

public UserService() {

conn = DatabaseConnection.getConnection();

}

// Add User

public void addUser(String username, String password, String role) {

String query = "INSERT INTO users (username, password, role) VALUES (?, ?, ?)";

try (PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, username);

stmt.setString(2, password);

stmt.setString(3, role);

stmt.executeUpdate();

System.out.println("User added successfully!");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Update User

public void updateUser(int id, String username, String password, String role) {

String query = "UPDATE users SET username=?, password=?, role=? WHERE id=?";

try (PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, username);

stmt.setString(2, password);

stmt.setString(3, role);

stmt.setInt(4, id);

stmt.executeUpdate();

System.out.println("User updated successfully!");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Delete User

public void deleteUser(int id) {

String query = "DELETE FROM users WHERE id=?";

try (PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setInt(1, id);

stmt.executeUpdate();

System.out.println("User deleted successfully!");

} catch (SQLException e) {

e.printStackTrace();

}

}

// View All Users

public List<User> getAllUsers() {

List<User> users = new ArrayList<>();

String query = "SELECT \* FROM users";

try (Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

while (rs.next()) {

users.add(new User(rs.getInt("id"), rs.getString("username"), rs.getString("password"), rs.getString("role")));

}

} catch (SQLException e) {

e.printStackTrace();

}

return users;

}

// User Login

public User loginUser(String username, String password) {

String query = "SELECT \* FROM users WHERE username=? AND password=?";

try (PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, username);

stmt.setString(2, password);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

return new User(rs.getInt("id"), rs.getString("username"), rs.getString("password"), rs.getString("role"));

}

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

**DatabaseConnection.java**

package org.example;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

private static final String URL = "jdbc:mysql://localhost:3306/UserManagementDB";

private static final String USER = "root";

private static final String PASSWORD = "";

private static Connection connection = null;

public static Connection getConnection() {

try {

if (connection == null || connection.isClosed()) {

// Load MySQL JDBC Driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Establish Connection

connection = DriverManager.getConnection(URL, USER, PASSWORD);

System.out.println("Database Connected Successfully!");

}

} catch (ClassNotFoundException e) {

System.out.println("MySQL JDBC Driver Not Found: " + e.getMessage());

} catch (SQLException e) {

System.out.println("Database Connection Failed: " + e.getMessage());

}

return connection;

}

}

**OUTPUT :**





