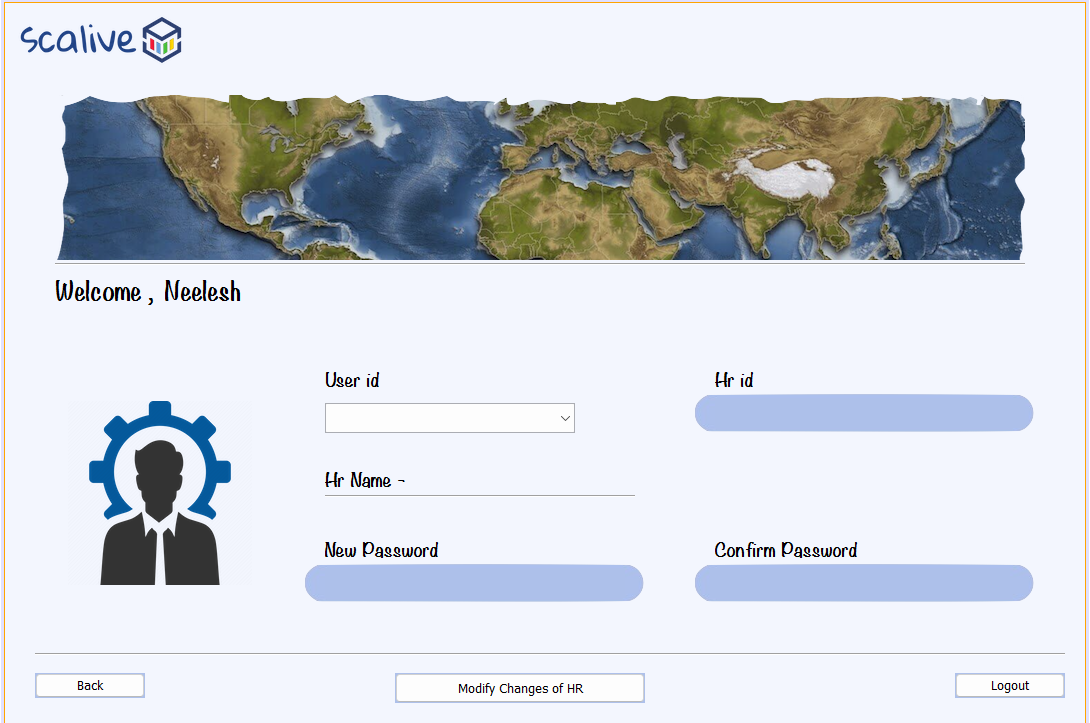
Java SE Project

Training And Placement Application

***AdminModifyHRFrame Documentation***

1. **AdminModifyHrFrame -**



**2. STEPS TO BE DONE IN** **AdminModifyHrFrame**

In the **AdminModifyHrFrame** we need to do following steps:

1. Displaying the **Username**
2. Allow the user to **logout**
3. Allow the user to **go back**
4. Write code for modifying the **HR** credentials;
5. **Displaying The Username**

Same as before

1. **Allowing The User To Logout**

Same as before

1. **Allowing The User To Go Back**

Same as before

1. **Writing Code For Modifying HR**

**WRITING THE CODE IN UserDAO**

**Returning All Userid-HrPojo as Key-Value Pair**

**public static List <HrPojo> getAllHr()throws SQLException{**

**Connection conn=DBConnection.getConnection();**

**Statement st=conn.createStatement();**

**ResultSet rs=st.executeQuery("Select hr.hrid,users.userid,users.name,hr.phone,hr.companyname from hr,users where hr.hrid=users.id and users.active='Y' order by hr.hrid");**

**List<HrPojo> allHrList=new ArrayList<>();**

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

**HrPojo obj=new HrPojo();**

**obj.setHrId(rs.getString(1));**

**obj.setUserId(rs.getString(2));**

**obj.setHrName(rs.getString(3));**

**obj.setPhone(rs.getString(4));**

**obj.setCompanyName(rs.getString(5));**

**allHrList.add(obj);**

**}**

**return allHrList;**

**}**

**Changing HR Password Code**

**public static boolean updatePwd(String userId,String password)throws SQLException{**

**Connection conn=DBConnection.getConnection();**

**PreparedStatement ps=conn.prepareStatement("update users set password=? ,active='Y' where userid=?");**

**ps.setString(1,password);**

**ps.setString(2,userId);**

**return 1==ps.executeUpdate();**

**}**

**The Code For Frame**

**public class AdminModifyHrFrame extends javax.swing.JFrame {**

**private Map<String,HrPojo> allHr;**

**public AdminModifyHrFrame() {**

**initComponents();**

**this.setLocationRelativeTo(null);**

**lblName.setText(CurrentUser.getName());**

**loadHrDetails();**

**}**

**private void loadHrDetails(){**

**try{**

**allHr=UserDao.getHrList();**

**Set<String> keys=allHr.keySet();**

**for(String id: keys){**

**jcUserId.addItem(id);**

**}**

**}**

**catch(SQLException e)**

**{**

**JOptionPane.showMessageDialog(null,"Some Error in DB","Error",JOptionPane.ERROR\_MESSAGE);**

**e.printStackTrace();**

**}**

**}**

**private void jcAllUserIdActionPerformed(java.awt.event.ActionEvent evt) {**

**String userId=(String)jcAllUserId.getSelectedItem();**

**HrPojo hr=allHr.get(userId);**

**txtHrId.setText(hr.getHrId());**

**lblHrName.setText(hr.getHrName());**

**}**

**private boolean validateInputs(){**

**char []pwd=txtPassword.getPassword();**

**char []repwd=txtRePassword.getPassword();**

**if(pwd.length <4 || repwd.length<4){**

**JOptionPane.showMessageDialog(null,"Passwords must be atleast 4 characters long");**

**return false;**

**}**

**password=String.valueOf(pwd);**

**String repassword=String.valueOf(repwd);**

**if(password.equals(repassword)==false){**

**JOptionPane.showMessageDialog(null,"Passwords not matching!");**

**return false;**

**}**

**return true;**

**}**

**private void clearText(){**

**txtPassword.setText("");**

**txtRePassword.setText("");**

**jcAllUserId.setSelectedIndex(0);**

**}**

**private void btnModifyHrActionPerformed(java.awt.event.ActionEvent evt) {**

**if(!validateInputs()){**

**return;**

**}**

**try{**

**boolean result=UserDAO.updatePwd(jcAllUserId.getSelectedItem().toString(), password);**

**if(result){**

**JOptionPane.showMessageDialog(null,"Password changed succesfully","Done!",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**else{**

**JOptionPane.showMessageDialog(null,"Sorry! Could not change the password","Unsuccessfull",JOptionPane.ERROR\_MESSAGE);**

**}**

**clearText();**

**}**

**catch(SQLException ex){**

**JOptionPane.showMessageDialog(null,"DB Error","Error",JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**private void backBtnActionPerformed(java.awt.event.ActionEvent evt) {**

**new AdminHrModule().setVisible(true);**

**this.dispose();**

**}**

**private void logOutBtnActionPerformed(java.awt.event.ActionEvent evt) {**

**new LoginFrame().setVisible(true);**

**this.dispose();**

**}**