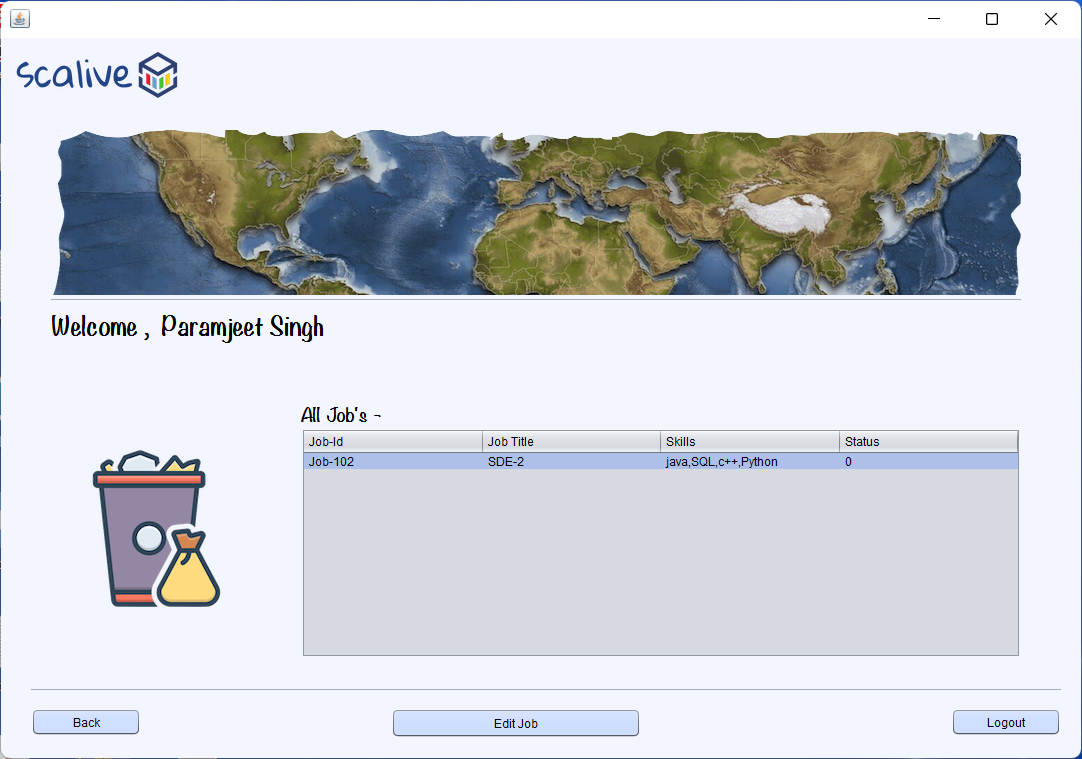
Java SE Project

Training And Placement Application

***HrEditJobFrame Documentation***

1. **HrChooseJobForEditingFrame -**



1. **HrEditJobFrame -**



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

In the **HrEditJobFrame** 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 loading all **editable jobs** by **current Hr**
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 loading Editable Jobs**

**WRITING THE CODE IN JobDAO**

**Returning All Editable JobList**

To return a **list of all editable Jobs** we must define a method in **JobDAO** called **getAllEditbleJobsByCurrentHr()**

This method should return JOBID,JOBTITLE,TAGS and STATUS of all EDITABLE JOBS i.e Jobs whose status is 0 , by the currently logged in HR from the DB

So the prototype of the method is:

**public static List<JobPojo> getAllEditableJobsByCurrentHr(String hrId)throws SQLException**

**The Code For getAllEditableJobsByCurrentHr()**

**public static List<JobPojo> getAllEditableJobsByCurrentHr(String hrId)throws SQLException{**

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

**PreparedStatement ps=conn.prepareStatement("Select jobid,jobtitle,tags,status from jobs where hrid=? and status=0");**

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

**ResultSet rs=ps.executeQuery();**

**List<JobPojo> allJobsList=new ArrayList<>();**

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

**JobPojo obj=new JobPojo();**

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

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

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

**obj.setStatus(rs.getInt(4));**

**allJobsList.add(obj);**

**}**

**return allJobsList;**

**}**

**Coding editJobByJobId() method in JobDAO**

**public static boolean editJobByJobId(JobPojo job)throws SQLException{**

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

**PreparedStatement ps=conn.prepareStatement("update jobs set jobtitle=?, tags=? where jobid=?");**

**ps.setString(1,job.getJobTitle());**

**ps.setString(2, job.getTags());**

**ps.setString(3,job.getJobId());**

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

**}**

**The Code For HrChooseJobForEditingFrame**

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

**DefaultTableModel model;**

**public HrChooseJobForEditingFrame() {**

**initComponents();**

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

**this.setLocationRelativeTo(null);**

**model=(DefaultTableModel)tblAllJobs.getModel();**

**loadJobDetails();**

**}**

**private void loadJobDetails(){**

**try{**

**List <JobPojo> allJobsList=JobDAO.getAllEditableJobsByCurrentHr(CurrentUser.getId());**

**for(JobPojo job: allJobsList){**

**Vector<String> row=new Vector<>();**

**row.add(job.getJobId());**

**row.add(job.getJobTitle());**

**row.add(job.getTags());**

**row.add(String.valueOf(job.getStatus()));**

**model.addRow(row);**

**}**

**if(allJobsList.isEmpty()){**

**JOptionPane.showMessageDialog(null,"No Editable Job Present!", "Editable Job Not Present", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**}**

**}**

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

**int row=tblAllJobs.getSelectedRow();**

**if(row==-1){**

**JOptionPane.showMessageDialog(null,"Please Select A Row First!", "No Selections!!", JOptionPane.INFORMATION\_MESSAGE);**

**return;**

**}**

**JobPojo job=new JobPojo();**

**job.setJobId(tblAllJobs.getValueAt(row, 0).toString().trim());**

**job.setJobTitle(tblAllJobs.getValueAt(row, 1).toString().trim());**

**job.setTags(tblAllJobs.getValueAt(row, 2).toString().trim());**

**job.setHrId(CurrentUser.getId());**

**job.setStatus(0);**

**new HrEditJobFrame(job).setVisible(true);**

**this.dispose();**

**}**

**The Code For HrEditJobFrame**

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

**private JobPojo job;**

**public HrEditJobFrame() {**

**initComponents();**

**this.setLocationRelativeTo(null);**

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

**}**

**public HrEditJobFrame(JobPojo job){**

**this();**

**this.job=job;**

**showJobDetails();**

**}**

**private void showJobDetails(){**

**String companyName="";**

**try{**

**companyName=HrDAO.getCompanyNameById(job.getHrId());**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**return;**

**}**

**lblJobId.setText(job.getJobId());**

**txtJobTitle.setText(job.getJobTitle());**

**txtCompanyName.setText(companyName);**

**String []allSkills=job.getTags().split(",");**

**txtSkill1.setText(allSkills[0]);**

**txtSkill2.setText(allSkills[1]);**

**txtSkill3.setText(allSkills[2]);**

**txtSkill4.setText(allSkills[3]);**

**}**

**private boolean validateInputs(){**

**if(txtJobTitle.getText().trim().isEmpty()||txtSkill1.getText().trim().isEmpty()||txtSkill2.getText().trim().isEmpty()||txtSkill3.getText().trim().isEmpty()||txtSkill4.getText().trim().isEmpty())**

**return false;**

**return true;**

**}**

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

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

**JOptionPane.showMessageDialog(null,"Please Fill All The Fields","Empty Data!!",JOptionPane.INFORMATION\_MESSAGE);**

**return;**

**}**

**job.setJobTitle(txtJobTitle.getText().trim());**

**job.setTags(txtSkill1.getText().trim()+","+txtSkill2.getText().trim()+","+txtSkill3.getText().trim()+","+txtSkill4.getText().trim());**

**try{**

**boolean result=JobDAO.editJobByJobId(job);**

**if(result){**

**JOptionPane.showMessageDialog(null,"Job Successfully Edited","Success",JOptionPane.INFORMATION\_MESSAGE);**

**return;**

**}**

**JOptionPane.showMessageDialog(null,"Job Not Edited","Editing Failed!",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**return;**

**}**

**}**