Java SE Project

Training And Placement Application

***PtViewAllAppliedJobsFrame Documentation***

1. **PtViewAllAppliedJobsFrame -**



**2. POJO ( Plain Old Java Object ) for JOB’s functionality -**

1. Create a class called **ParticipantJobPojo** which extends **JobPojo** and adds a new data called **companyName**

**public class ParticipantJobPojo extends JobPojo {**

**private String companyName;**

**public String getCompanyName() {**

**return companyName;**

**}**

**public void setCompanyName(String companyName) {**

**this.companyName = companyName;**

**}**

**}**

**3. STEPS TO BE DONE IN** **PtViewAllAppliedJobsFrame**

In the **PtViewAllAppliedJobsFrame** 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 open **jobs**
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 Available Jobs**

**WRITING THE CODE IN JobDAO**

**Returning All Open Jobs JobList**

To return a **list of all open Jobs** we must define a method in **ResultDAO** called **getAllAppliedJobs()**

This method should return JOBID,JOBTITLE,COMPANYNAME and TAGS of all APPLIED JOBS i.e Jobs applied by current user whose percentage is -1

So the prototype of the method is:

**public static List< ParticipantJobPojo> getAllAppliedJobs(String participantId)throws SQLException**

**The Code For getAllAppliedJobs()**

**public static List<ParticipantJobPojo> getAllAppliedJobs(String particpantId)throws SQLException**

**{**

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

**PreparedStatement ps=conn.prepareStatement("Select jobs.jobid,jobs.jobtitle,companyname,tags from jobs,results,hr where jobs.jobid=results.jobid and jobs.hrid=hr.hrid and participantid=? and percentage=-1");**

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

**List<ParticipantJobPojo> allAppliedJobsList=new ArrayList<>();**

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

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

**{**

**ParticipantJobPojo pj=new ParticipantJobPojo();**

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

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

**pj.setCompanyName(rs.getString(3));**

**pj.setTags(rs.getString(4));**

**allAppliedJobsList.add(pj);**

**}**

**return allAppliedJobsList;**

**}**

**The Code For PtViewAllAppliedJobsFrame**

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

**private DefaultTableModel model;**

**private List<ParticipantJobPojo> allAppliedJobsList;**

**public PtViewAllAppliedJobsFrame() {**

**initComponents();**

**this.setLocationRelativeTo(null);**

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

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

**loadAppliedJobsDetails();**

**}**

**private void loadAppliedJobsDetails(){**

**try{**

**allAppliedJobsList=ResultDAO.getAllAppliedJobs(CurrentUser.getId());**

**for(ParticipantJobPojo job:allAppliedJobsList){**

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

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

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

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

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

**model.addRow(row);**

**}**

**if(allAppliedJobsList.isEmpty()){**

**JOptionPane.showMessageDialog(null,"You Haven't Applied For Any Job Yet","Applied Job Not Present",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**}**

**}**

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

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

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

**JOptionPane.showMessageDialog(null,"Please select a row first");**

**return;**

**}**

**String jobId=tblAllJobs.getValueAt(row,0).toString().trim();**

**String jobTitle=tblAllJobs.getValueAt(row,1).toString().trim();**

**PtTakeQuizFrame quizFrame=new PtTakeQuizFrame(jobId,jobTitle);**

**}**