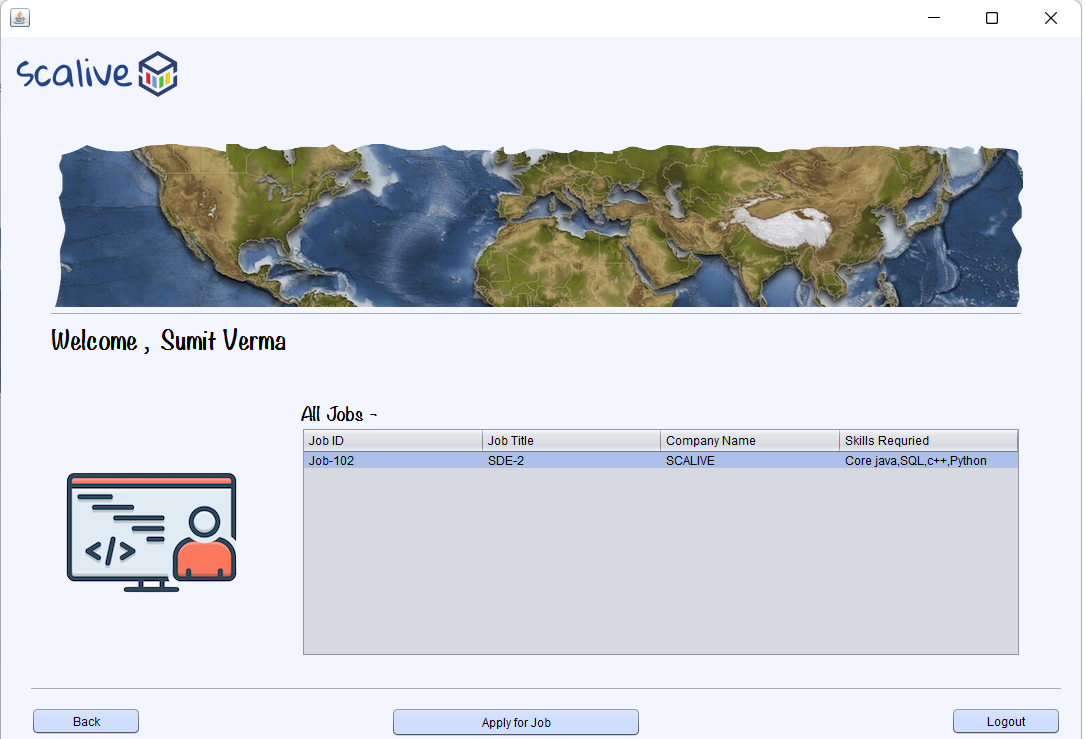
Java SE Project

Training And Placement Application

***PtViewAllAvailableJobsFrame Documentation***

1. **PtViewAllAvailableJobsFrame -**



**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;**

**}**

**}**

1. Create a class called **ResultPojo** which corresponds to **RESULTS** table

**public class ResultsPojo {**

**private String pId;**

**private String jobId;**

**private double percentage;**

**private String selectedByHr;**

**public String getpId() {**

**return pId;**

**}**

**public void setpId(String pId) {**

**this.pId = pId;**

**}**

**public String getJobId() {**

**return jobId;**

**}**

**public void setJobId(String jobId) {**

**this.jobId = jobId;**

**}**

**public double getPercentage() {**

**return percentage;**

**}**

**public void setPercentage(double percentage) {**

**this.percentage = percentage;**

**}**

**public String getSelectedByHr() {**

**return selectedByHr;**

**}**

**public void setSelectedByHr(String selectedByHr) {**

**this.selectedByHr = selectedByHr;**

**}**

**}**

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

In the **PtViewAllAvailableJobsFrame** 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 **JobDAO** called **getAllOpenJobs()**

This method should return JOBID,JOBTITLE,COMPANYNAME and TAGS of all OPEN JOBS i.e Jobs whose status is 1

So the prototype of the method is:

**public static List< ParticipantJobPojo> getAllAvailableJobs()throws SQLException**

**The Code For getAllAvailableJobs()**

**public static List<ParticipantJobPojo> getAllAvailableJobs()throws SQLException**

**{**

**Statement st=DBConnection.getConnection().createStatement();**

**ResultSet rs=st.executeQuery("select jobid,jobtitle,companyname,tags,jobs.hrid from jobs,hr where jobs.hrid=hr.hrid and status=1 order by jobid ");**

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

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

**ParticipantJobPojo obj=new ParticipantJobPojo();**

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

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

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

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

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

**allOpenJobsList.add(obj);**

**}**

**return allOpenJobsList;**

**}**

**WRITING THE CODE IN ResultDAO**

**Applying For JOB By Participant**

To apply for JOB by participant we must define a method in **ResultDAO** called **applyForJob()**

**public static boolean applyForJob(ResultPojo rs)throws SQLException**

**The Code For applyForJob()**

**public static boolean applyForJob(ResultsPojo result)throws SQLException**

**{**

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

**PreparedStatement ps=conn.prepareStatement("Select \* from results where participantid=? and jobid=?");**

**ps.setString(1,result.getpId());**

**ps.setString(2,result.getJobId());**

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

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

**return false;**

**}**

**ps=conn.prepareStatement("insert into results values(?,?,?,?)");**

**ps.setString(1, result.getpId());**

**ps.setString(2, result.getJobId());**

**ps.setDouble(3, result.getPercentage());**

**ps.setString(4, result.getSelectedByHr());**

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

**}**

**The Code For PtViewAllAvailableJobsFrame**

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

**private DefaultTableModel model;**

**private List<ParticipantJobPojo> allOpenJobsList;**

**public PtViewAllAvailableJobsFrame() {**

**initComponents();**

**this.setLocationRelativeTo(null);**

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

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

**loadOpenJobsDetails();**

**}**

**private void loadOpenJobsDetails(){**

**try{**

**allOpenJobsList=JobDAO.getAllAvailableJobs();**

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

**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(allOpenJobsList.isEmpty()){**

**JOptionPane.showMessageDialog(null,"No Jobs Available For Applying", "Job Not Found!",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**}**

**}**

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

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

**if(row>-1)**

**{**

**try**

**{**

**ResultPojo rs=new ResultPojo();**

**rs.setParticipantId(CurrentUser.id);**

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

**rs.setPercentage(-1.0);**

**if(ResultDAO.applyForJob(rs))**

**{**

**JOptionPane.showMessageDialog(null,"Applied for Job successFully\n"**

**+ "Now go to view all applied job and take Test !!!","Success...",JOptionPane.INFORMATION\_MESSAGE);**

**return;**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null,"You already applied for this job !!!","Warning",JOptionPane.INFORMATION\_MESSAGE);**

**return;**

**}**

**}**

**catch(SQLException ex)**

**{**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(null,"DB-Error While applied for job !!!","Error",JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null,"Please select any row first !!!","Warning",JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**}**