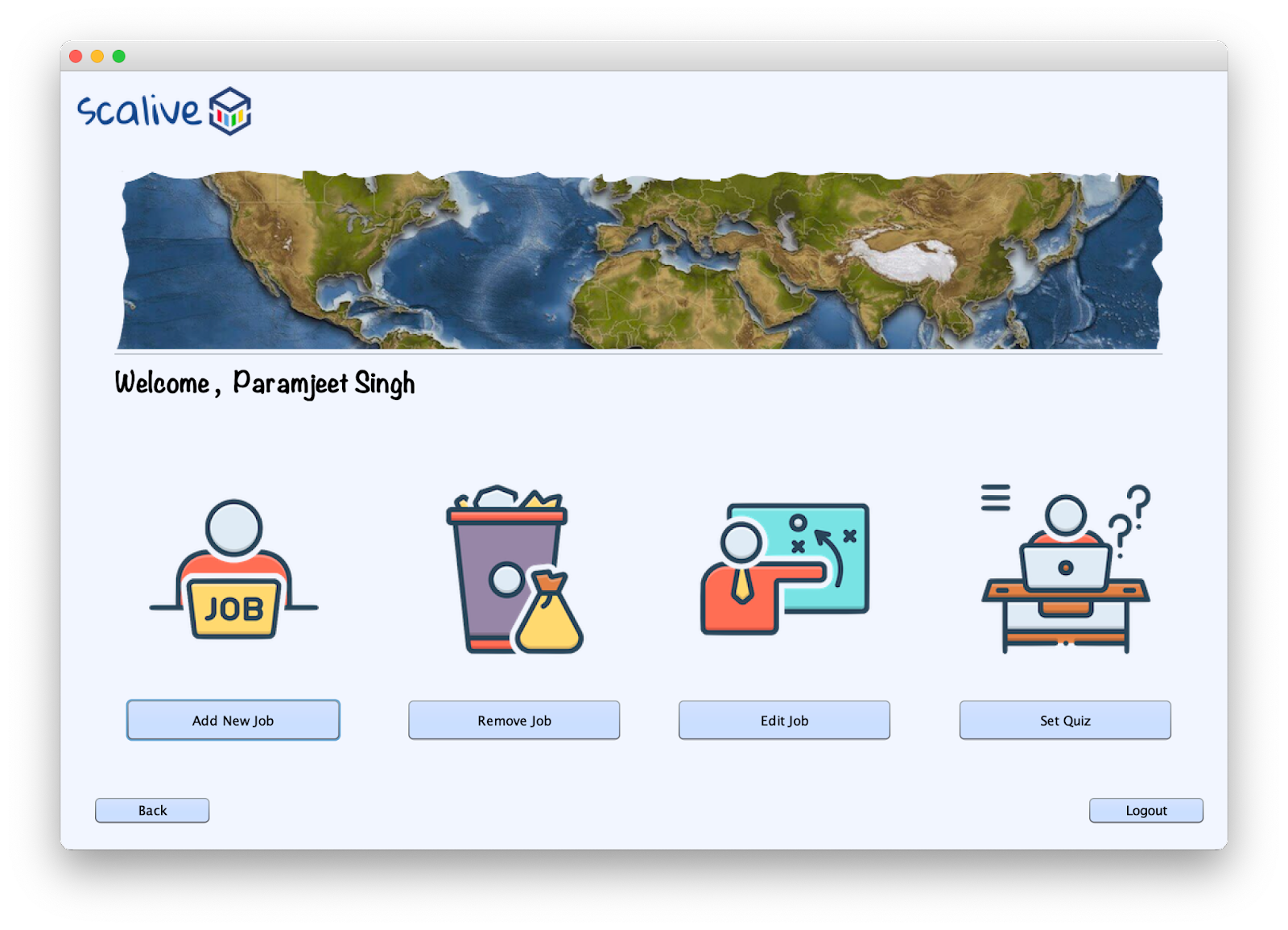
Java SE Project

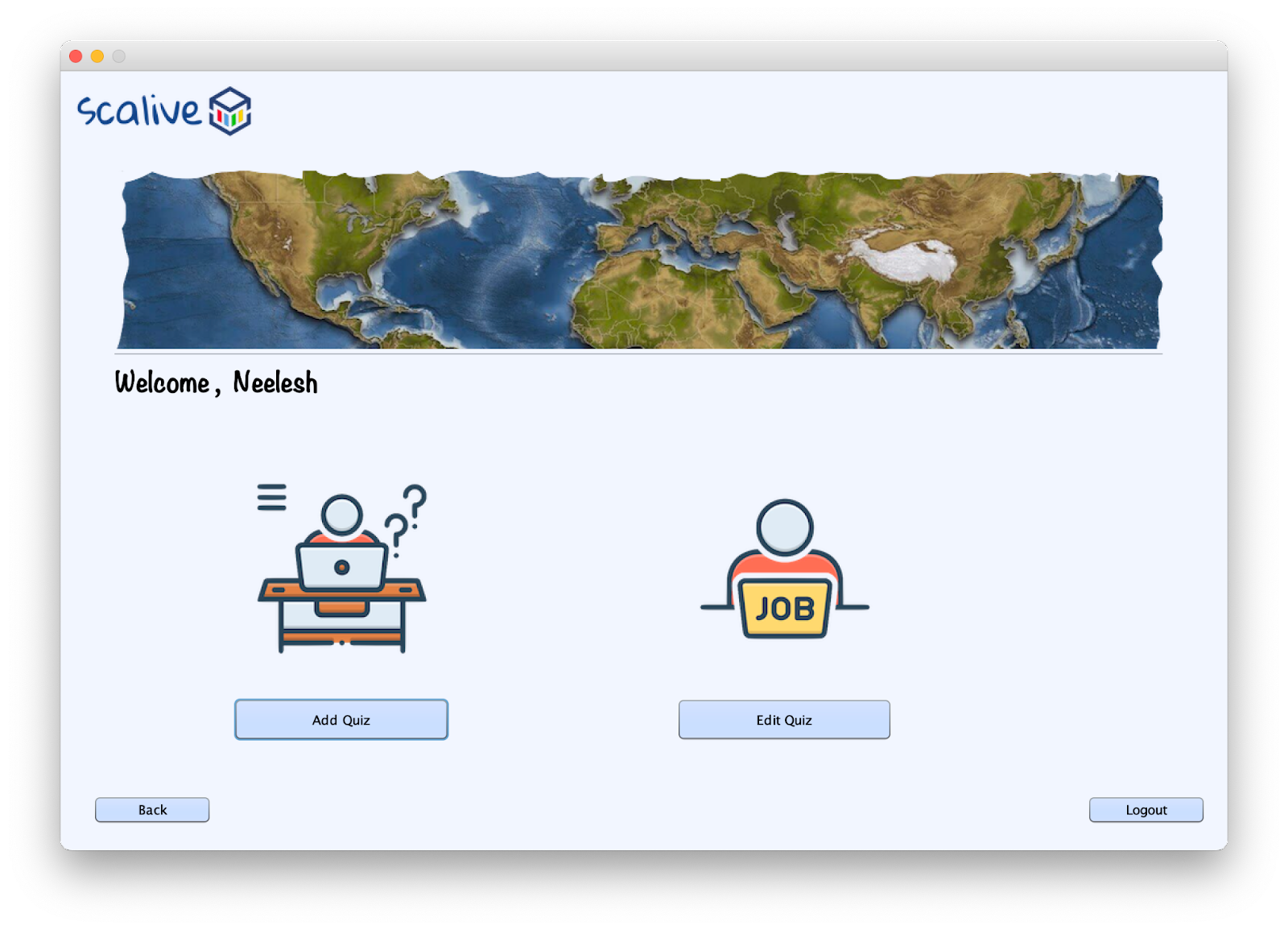
Training And Placement Application

***HrEditJobFrame Documentation***

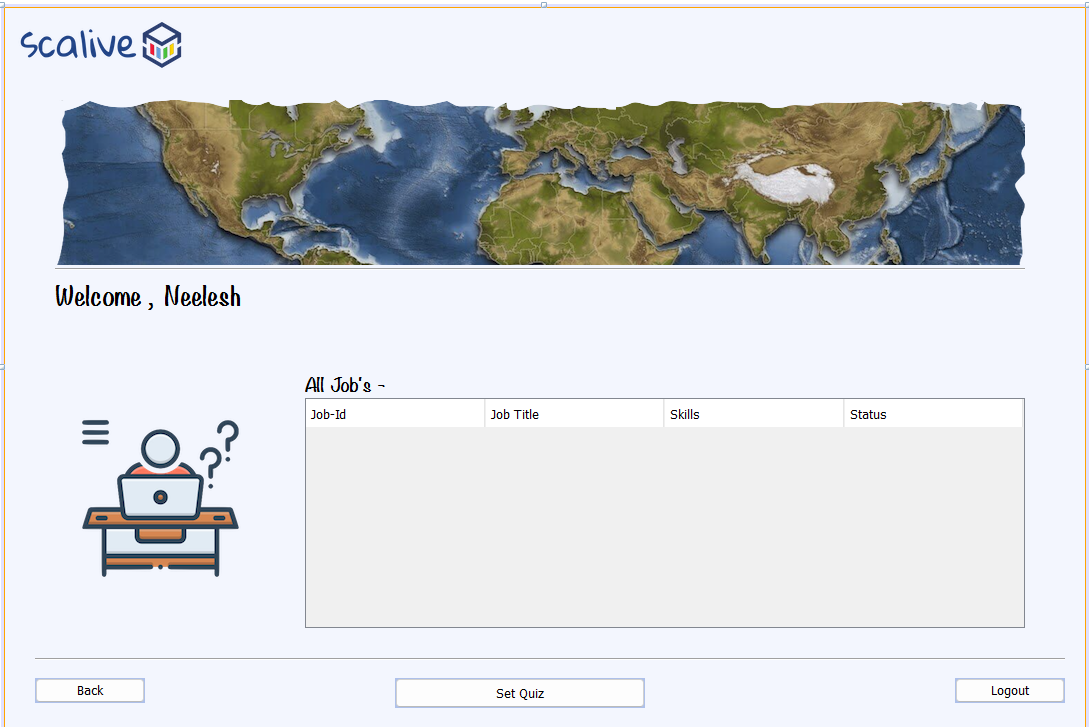
1. **HrJobOptionsFrame -**



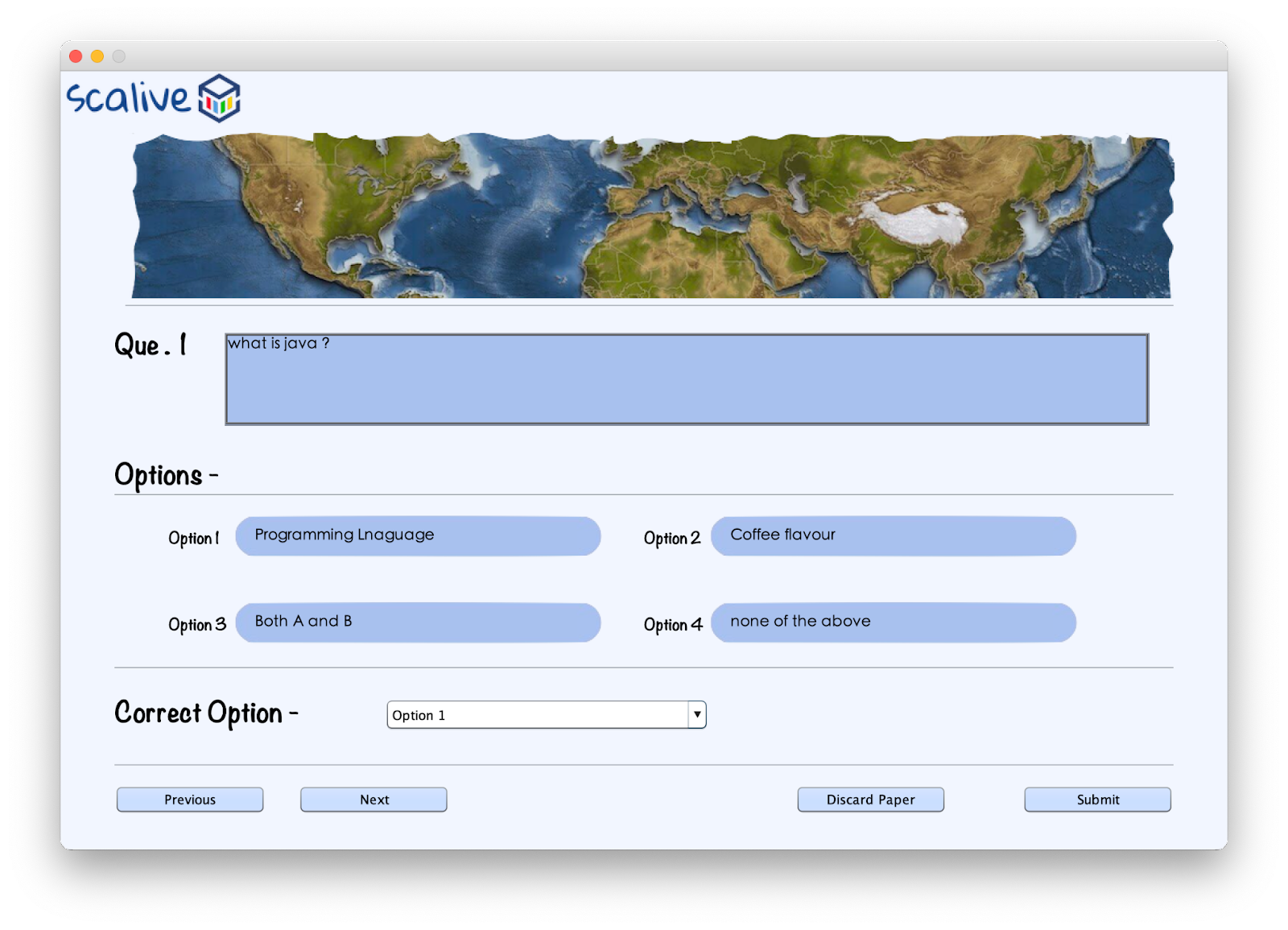
1. **HrQuizOptionsFrame –**



1. **HrChooseJobForSettingQuizFrame –**

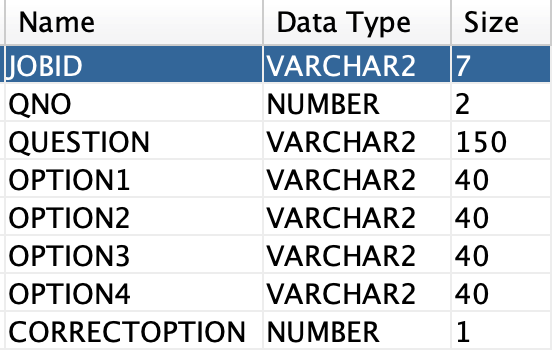


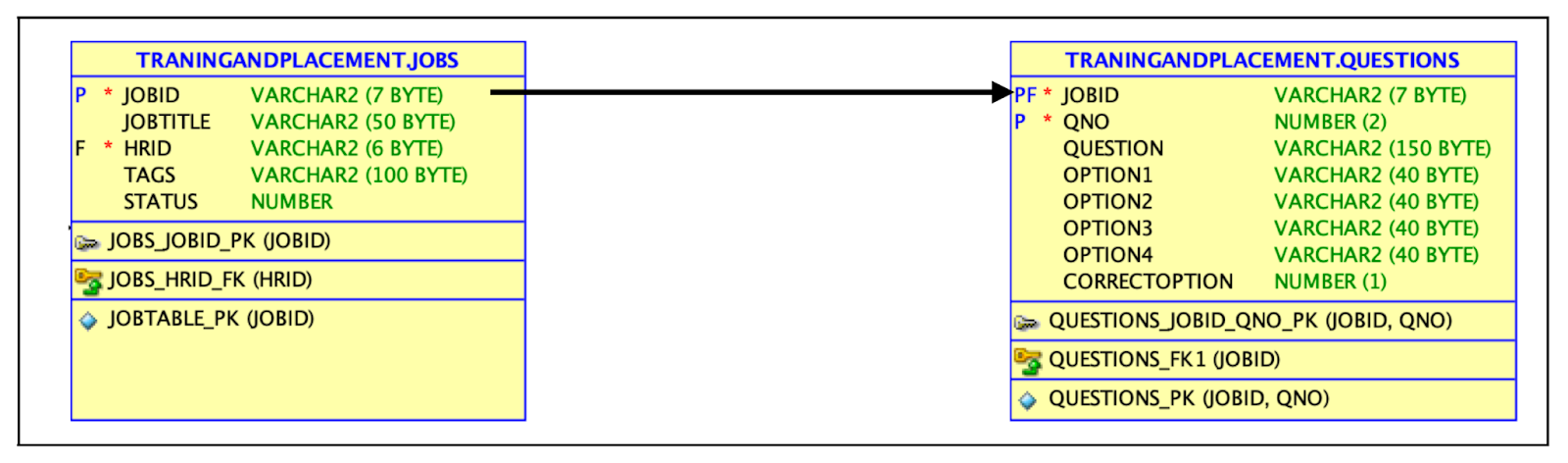
1. **HrSetQuizFrame –**

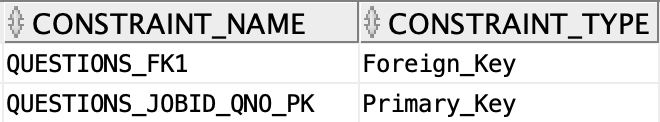


For Quiz functionality, we have to create a table in the database by name of “**Questions**”.

Column name and data type are shown below snapshot :







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

1. create a class inside the package by the name of **“QuestionPojo.java”**

**public class QuestionPojo {**

**public String getJobId() {**

**return jobId;**

**}**

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

**this.jobId = jobId;**

**}**

**public int getQno() {**

**return qno;**

**}**

**public void setQno(int qno) {**

**this.qno = qno;**

**}**

**public String getQuestion() {**

**return question;**

**}**

**public void setQuestion(String question) {**

**this.question = question;**

**}**

**public String getOption1() {**

**return option1;**

**}**

**public void setOption1(String option1) {**

**this.option1 = option1;**

**}**

**public String getOption2() {**

**return option2;**

**}**

**public void setOption2(String option2) {**

**this.option2 = option2;**

**}**

**public String getOption3() {**

**return option3;**

**}**

**public void setOption3(String option3) {**

**this.option3 = option3;**

**}**

**public String getOption4() {**

**return option4;**

**}**

**public void setOption4(String option4) {**

**this.option4 = option4;**

**}**

**public int getCorrectOption() {**

**return correctOption;**

**}**

**public void setCorrectOption(int correctOption) {**

**this.correctOption = correctOption;**

**}**

**private String jobId;**

**private int qno;**

**private String question;**

**private String option1;**

**private String option2;**

**private String option3;**

**private String option4;**

**private int correctOption;**

**}**

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

In the **HrChooseJobForEditingQuizFrame** 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 **jobs** by **current Hr** for which Quiz can be set
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 **getAllOpenJobsByCurrentHr()**

This method should return JOBID,JOBTITLE,TAGS and STATUS of all OPEN JOBS i.e Jobs whose status is 1 , by the currently logged in HR from the DB

So the prototype of the method is:

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

**The Code For getAllOpenJobsByCurrentHr()**

**public static List<JobPojo> getAllOpenJobsByCurrentHr( String hrId ) throws SQLException // Edit Job Dao Method**

**{**

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

**PreparedStatement ps=conn.prepareStatement("select jobid , jobtitle , tags , status from jobs where hrid=? and status=1 order by jobid");**

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

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

**List<JobPojo> allJobList=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));**

**allJobList.add(obj);**

**}**

**return allJobList;**

**}**

**The Code For HrChooseJobForEditingQuizFrame**

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

**DefaultTableModel model;**

**public HrChooseJobForEditingQuizFrame() {**

**initComponents();**

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

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

**this.setLocationRelativeTo(null);**

**this.setVisible(true);**

**loadJobDetails();**

**}**

**private void loadJobDetails(){**

**try{**

**List<JobPojo> allJobsList=JobDAO.getAllOpenJobByCurrentHr(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 Job Available For Setting Quiz", "Job Not Found!",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**}**

**}**

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

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

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

**JOptionPane.showMessageDialog(null,"Please Select A Row Firs!!", "No Row Selected!!",JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

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

**new HrEditQuizFrame(jobId).setVisible(true);**

**this.dispose();**

**}**

**Coding getQuestionPaperByJobId() Method In QuestionDAO**

**public static List<QuestionPojo> getQuestionPaperByJobId(String jobId)throws SQLException**

**{**

**PreparedStatement ps;**

**ps=DBConnection.getConnection().prepareStatement("select \* from questions where jobid=?");**

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

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

**List<QuestionPojo> questionsList = new ArrayList<QuestionPojo>();**

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

**{**

**QuestionPojo que=new QuestionPojo();**

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

**que.setqNo(rs.getInt(2));**

**que.setQuestion(rs.getString(3));**

**que.setOption1(rs.getString(4));**

**que.setOption2(rs.getString(5));**

**que.setOption3(rs.getString(6));**

**que.setOption4(rs.getString(7));**

**que.setCorrectOption(rs.getInt(8));**

**questionsList.add(que);**

**}**

**return questionsList;**

**}**

**Coding editQuestionPaper() method in QuestionDAO**

**public static boolean editQuestionPaper(List<QuestionPojo> questionsList)throws SQLException**

**{**

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

**PreparedStatement ps=conn.prepareStatement("update questions set question=?,option1=?,option2=?,option3=?,option4=?,correctoption=? where jobid=? and qno=?");**

**int count=0;**

**for(QuestionPojo que:questionsList)**

**{**

**ps.setString(1,que.getQuestion());**

**ps.setString(2,que.getOption1());**

**ps.setString(3,que.getOption2());**

**ps.setString(4,que.getOption3());**

**ps.setString(5,que.getOption4());**

**ps.setInt(6,que.getCorrectOption());**

**ps.setString(7,que.getJobId());**

**ps.setInt(8,que.getqNo());**

**int res=ps.executeUpdate();**

**if(res==1)**

**++count;**

**}**

**if(count==questionsList.size())**

**return true;**

**return false;**

**}**

**The Code For HrEditQuizFrame**

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

**List<QuestionPojo> questionsList;**

**private String jobId;**

**private QuestionPojo currentQuestion;**

**int counter=0,total=0;**

**public HrEditQuizFrame() {**

**initComponents();**

**lblQno.setText(""+(counter+1));**

**this.setLocationRelativeTo(null);**

**}**

**public HrEditQuizFrame(String jobId) {**

**this();**

**this.jobId=jobId;**

**lblQno.setText(""+(counter+1));**

**loadQuestionPaper();**

**}**

**boolean validateInput()**

**{**

**if(txtQuestion.getText().equals("") || op1Txt.getText().equals("") || txtOption2.getText().equals("") || txtOption3.getText().equals("") || txtOption4.getText().equals("") || jcCorrectOption.getSelectedIndex()==0)**

**return false;**

**return true;**

**}**

**private void loadQuestionPaper(){**

**try**

**{**

**questionsList=QuestionDao.getQuestionPaperByJobId( jobId );**

**total=questionsList.size( );**

**showNextQuestionInFrame();**

**}**

**catch(SQLException ex)**

**{**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(null,"DB-Error While editing quiz...","Error",JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**}**

**private void showNextQuestionInFrame(){**

**currentQuestion=questionList.get(counter);**

**txtQuestion.setText(currentQuestion.getQuestion());**

**txtOption1.setText(currentQuestion.getOption1());**

**txtOption2.setText(currentQuestion.getOption2());**

**txtOption3.setText(currentQuestion.getOption3());**

**txtOption4.setText(currentQuestion.getOption4());**

**jcCorrectOption.setSelectedIndex(currentQuestion.getCorrectOption());**

**}**