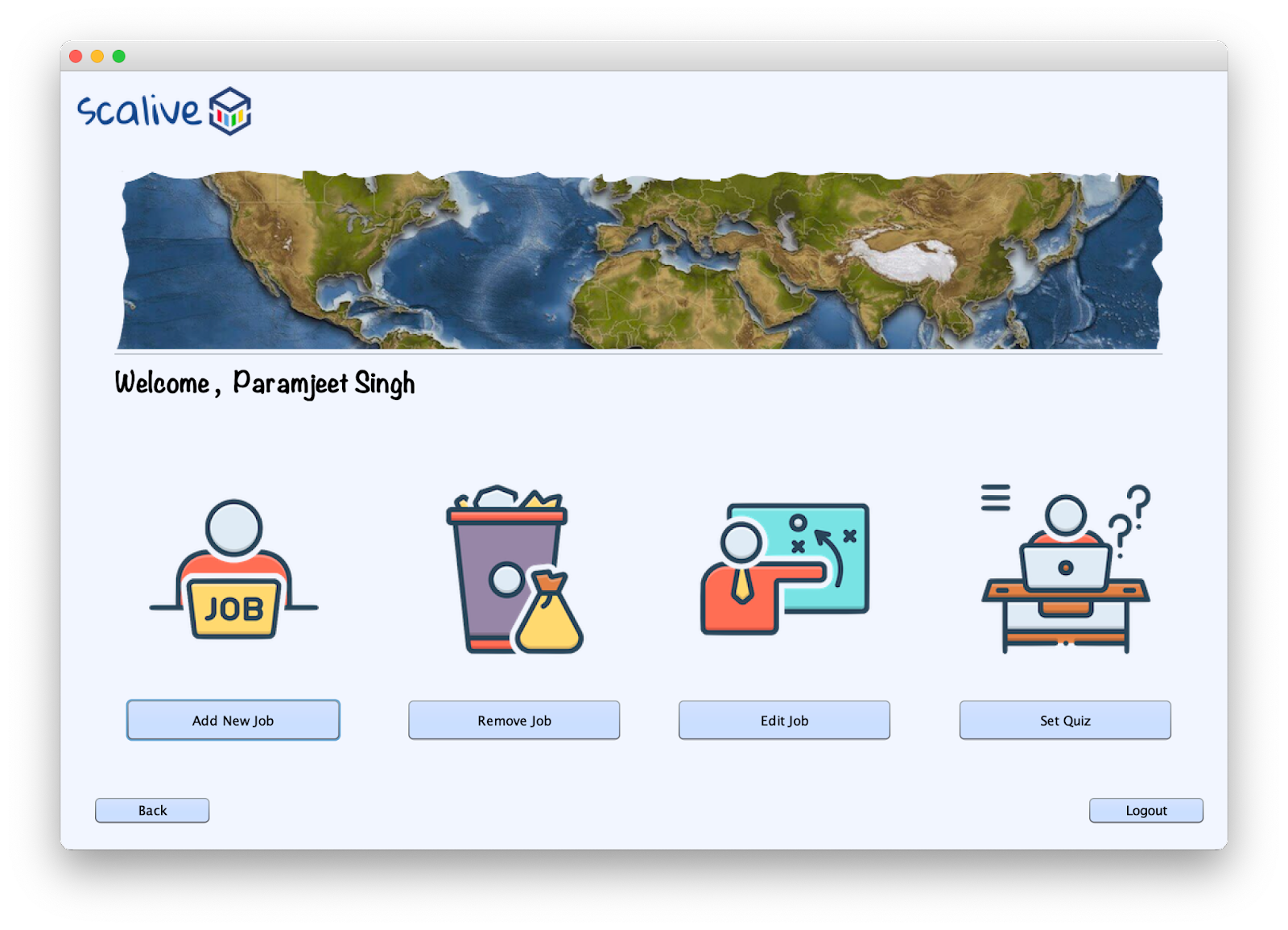
Java SE Project

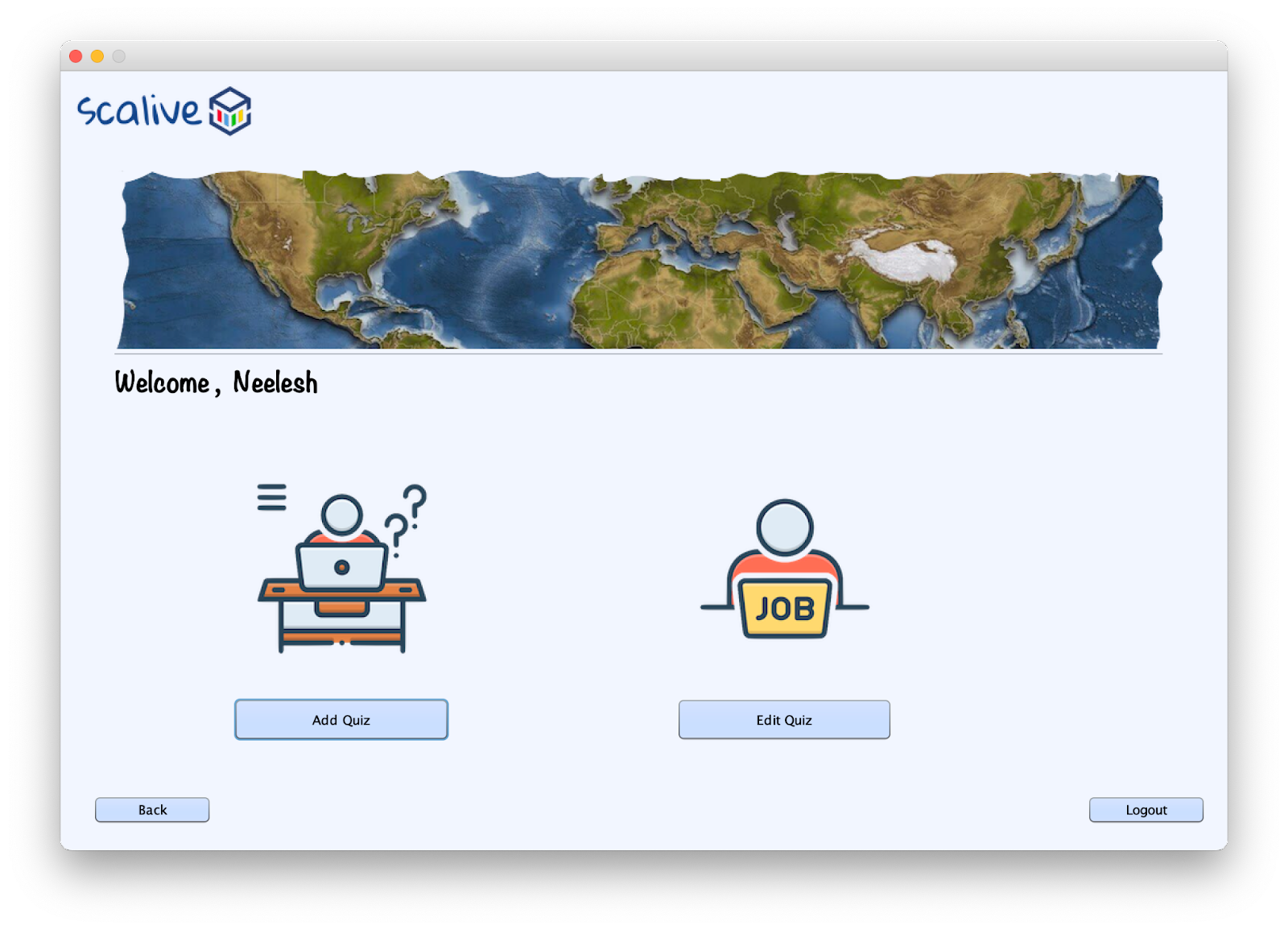
Training And Placement Application

***HrEditJobFrame Documentation***

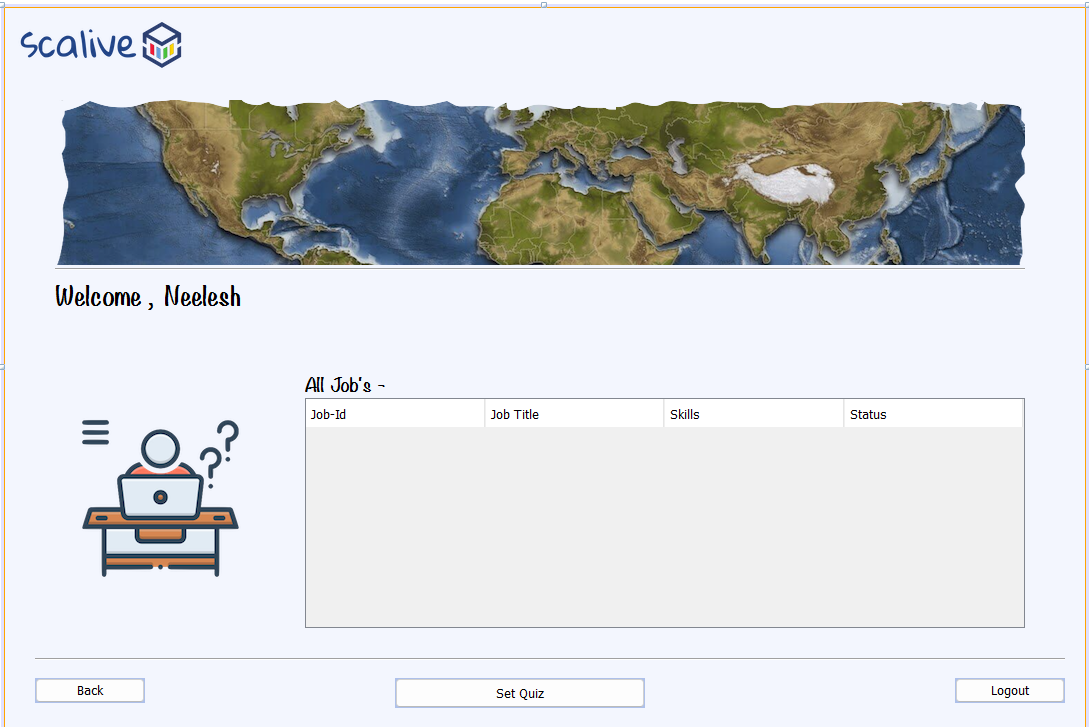
1. **HrJobOptionsFrame -**



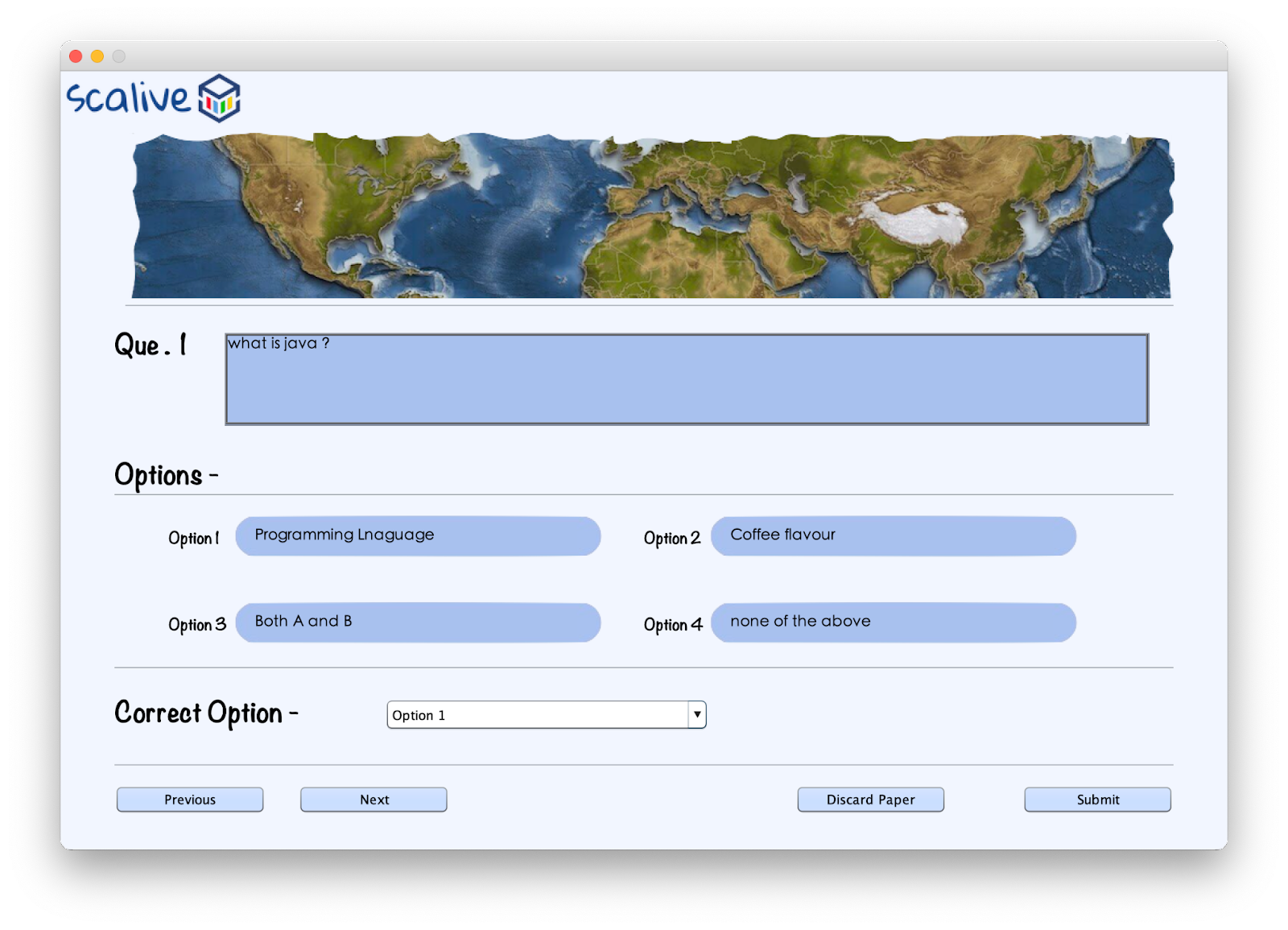
1. **HrQuizOptionsFrame –**



1. **HrChooseJobForSettingQuizFrame –**

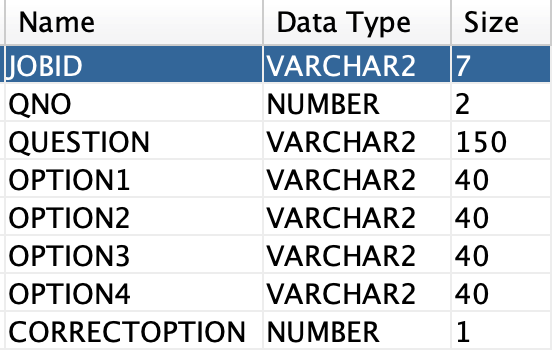


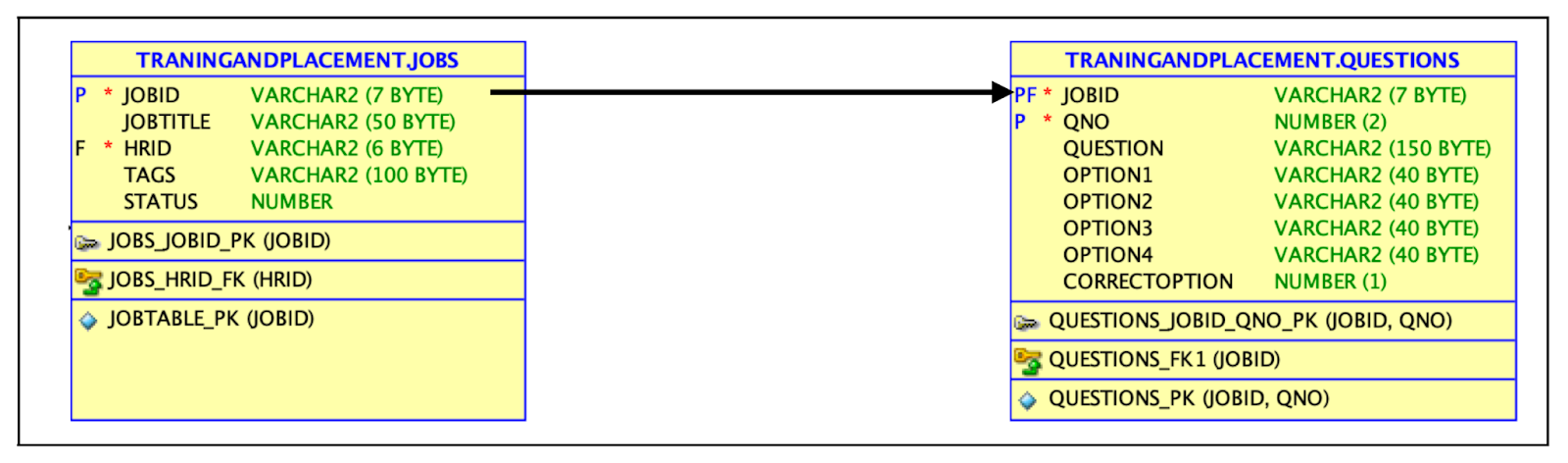
1. **HrSetQuizFrame –**

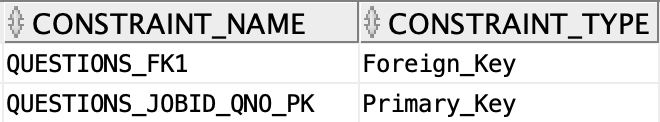


For Quiz functionality, we have to create a table in the database by name of “**Questions**”.

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** **HrChooseJobForSettingQuizFrame**

In the **HrChooseJobForSettingQuizFrame** 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 **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;**

**}**

**The Code For HrChooseJobForSettingQuizFrame**

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

**DefaultTableModel model;**

**public HrChooseJobForSettingQuizFrame() {**

**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 btnSetQuizActionPerformed(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;**

**}**

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

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

**this.dispose();**

**}**

**Coding setPaper() method in QuestionDAO**

**public static boolean setPaper(List<QuestionPojo> questionList)throws SQLException{**

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

**int count=0;**

**PreparedStatement ps=conn.prepareStatement("Insert into questions values(?,?,?,?,?,?,?,?");**

**for(QuestionPojo question:questionList){**

**ps.setString(1, question.getJobId());**

**ps.setInt(2, question.getQno());**

**ps.setString(3, question.getQuestion());**

**ps.setString(4, question.getOption1());**

**ps.setString(5,question.getOption2());**

**ps.setString(6, question.getOption3());**

**ps.setString(7, question.getOption4());**

**ps.setInt(8, question.getCorrectOption());**

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

**if(res==1)**

**++count;**

**}**

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

**return true;**

**return false;**

**}**

**Coding setJobStatus() method in JobDAO**

**public static void setJobStatus(String jobId)throws SQLException**

**{**

**PreparedStatement ps;**

**ps=DBConnection.getConnection().prepareStatement("update jobs set status=1 where jobId=?");**

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

**ps.executeUpdate();**

**}**

**The Code For HrSetQuizFrame**

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

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

**private String jobId;**

**int counter=0;**

**public HrSetQuizFrame(){**

**initComponents();**

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

**this.setLocationRelativeTo(null);**

**}**

**public HrSetQuizFrame(String jobId) {**

**this();**

**this.jobId=jobId;**

**}**

**private boolean validateInput(){**

**if(txtQuestion.getText().trim().isEmpty()||txtOption1.getText().trim().isEmpty()||txtOption2.getText().trim().isEmpty()||txtOption3.getText().trim().isEmpty()||txtOption4.getText().trim().isEmpty()||jcCorrectOption.getSelectedIndex()==0)**

**return false;**

**return true;**

**}**

**private void clearText(){**

**txtQuestion.setText("");**

**txtOption1.setText("");**

**txtOption2.setText("");**

**txtOption3.setText("");**

**txtOption4.setText("");**

**jcCorrectOption.setSelectedIndex(0);**

**txtQuestion.requestFocus();**

**}**

**private QuestionPojo setQuestion(){**

**QuestionPojo question=new QuestionPojo();**

**question.setJobId(jobId);**

**question.setQno(counter+1);**

**question.setQuestion(txtQuestion.getText().trim());**

**question.setOption1(txtOption1.getText().trim());**

**question.setOption2(txtOption2.getText().trim());**

**question.setOption3(txtOption3.getText().trim());**

**question.setOption4(txtOption4.getText().trim());**

**question.setCorrectOption(jcCorrectOption.getSelectedIndex());**

**return question;**

**}**

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

**if(counter==0)**

**return;**

**if(counter==questionList.size()) {**

**if(validateInput()){**

**int result=JOptionPane.showConfirmDialog(null,"You have an unsaved question. Do you want to save it ?","Confirm ?",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);**

**if(result==JOptionPane.YES\_OPTION){**

**QuestionPojo question=setQuestion();**

**questionList.add(question);**

**clearText();**

**}**

**}**

**--counter;**

**getQuestionFromList();**

**}else{**

**if(validateInput()){**

**QuestionPojo question=setQuestion();**

**questionList.set(counter,question);**

**--counter;**

**getQuestionFromList();**

**}else{**

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

**}**

**}**

**lblQno.setText(String.valueOf(counter+1));**

**}**

**private void getQuestionFromList(){**

**QuestionPojo question;**

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

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

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

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

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

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

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

**}**

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

**if(validateInput()){**

**QuestionPojo question=setQuestion();**

**if(counter==questionList.size()){**

**questionList.add(question);**

**clearText();**

**++counter;**

**}else{**

**questionList.set(counter, question);**

**++counter;**

**if(counter==questionList.size()){**

**clearText();**

**}else{**

**getQuestionFromList();**

**}**

**}**

**lblQno.setText(String.valueOf(counter+1));**

**}else{**

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

**}**

**}**

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

**if(validateInput()){**

**QuestionPojo que=setQuestion();**

**if(counter==questionList.size()){**

**questionList.add(que);**

**}else{**

**questionList.set(counter, que);**

**}**

**try{**

**boolean result=QuestionDAO.setPaper(questionList);**

**if(result){**

**JOptionPane.showMessageDialog(null, "Question Paper Added Successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);**

**JobDAO.setJobStatus(jobId);**

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

**this.dispose();**

**}else{**

**JOptionPane.showMessageDialog(null, "Could Not Add The Paper", "Failure!!", JOptionPane.WARNING\_MESSAGE);**

**}**

**}catch(SQLException ex){**

**JOptionPane.showMessageDialog(null, "DB Error", "Error In DB Communication", JOptionPane.WARNING\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**else{**

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

**}**

**}**