Java SE Project

Training And Placement Application

***PtTakeQuizFrame Documentation***

1. **PtTakeQuizFrame -**



**WRITING THE CODE IN QuestionDAO**

**Returning Question Paper Of JobID**

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

**The Code For getAllQuestionPaperByJobId()**

**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;**

**}**

**Setting The Result In ResultDAO**

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

**The Code For setResult()**

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

**{**

**PreparedStatement ps;**

**ps=DBConnection.getConnection().prepareStatement("update results set percentage=? where participantid=? and jobid=?");**

**ps.setDouble(1,rs.getPercentage());**

**ps.setString(2,rs.getParticipantId());**

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

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

**}**

**The Code For PtTakeQuizFrame**

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

**List<QuestionPojo> questionList;**

**private String jobId,jobTitle;**

**private QuestionPojo currentQuestion;**

**int counter=0;**

**List<Integer> userAnswerList;**

**PtTakeQuizFrame pt=this;**

**public PtTakeQuizFrame() {**

**initComponents();**

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

**this.setLocationRelativeTo(null);**

**}**

**public PtTakeQuizFrame(String jobId,String jobTitle) {**

**this();**

**this.jobId=jobId;**

**this.jobTitle=jobTitle;**

**loadQuestionPaper();**

**}**

**private void loadQuestionPaper(){**

**try{**

**questionList=QuestionDAO.getQuestionPaperByJobId(jobId);**

**userAnswerList=new ArrayList<Integer>(questionList.size());**

**System.out.println("Size of QL:"+questionList.size());**

**System.out.println("Size of Al:"+userAnswerList.size());**

**for(int i=0;i<questionList.size();i++)**

**userAnswerList.add(i,0);**

**System.out.println("Size of Al:"+userAnswerList.size());**

**lblTitle.setText("Paper For "+this.jobTitle+" Total Questions:"+questionList.size());**

**showNextQuestionInFrame();**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**}**

**}**

**private void showNextQuestionInFrame(){**

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

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

**jrbOption1.setText(currentQuestion.getOption1());**

**jrbOption2.setText(currentQuestion.getOption2());**

**jrbOption3.setText(currentQuestion.getOption3());**

**jrbOption4.setText(currentQuestion.getOption4());**

**int answer=userAnswerList.get(counter);**

**switch(answer){**

**case 1:**

**jrbOption1.setSelected(true);**

**break;**

**case 2:**

**jrbOption2.setSelected(true);**

**break;**

**case 3:**

**jrbOption3.setSelected(true);**

**break;**

**case 4:**

**jrbOption4.setSelected(true);**

**break;**

**default:**

**buttonGroup1.clearSelection();**

**}**

**}**

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

**int res=validateChooseOption();**

**if(res==0){**

**JOptionPane.showMessageDialog(null,"Please Select An Option","No Option Selected!",JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**answerList.set(counter, res);**

**--counter;**

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

**counter=questionList.size()-1;**

**}**

**showNextQuestionInFrame();**

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

**}**

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

**int res=validateChooseOption();**

**if(res==0){**

**JOptionPane.showMessageDialog(null,"Please Select An Option","No Option Selected!",JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**answerList.set(counter, res);**

**++counter;**

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

**counter=0;**

**}**

**showNextQuestionInFrame();**

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

**}**

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

**int res=validateChooseOption();**

**if(res==0){**

**JOptionPane.showMessageDialog(null,"Please Select An Option","No Option Selected!",JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**answerList.set(counter, res);**

**double per=getPercentage();**

**ResultsPojo result=new ResultsPojo();**

**result.setJobId(jobId);**

**result.setpId(CurrentUser.getId());**

**if(per<5)**

**per=5.0;**

**result.setPercentage(per);**

**result.setSelectedByHr("No");**

**try{**

**boolean resp=ResultDAO.setResult(result);**

**if(resp==true){**

**JOptionPane.showMessageDialog(null,"You Got "+per+"% marks+\nAnd Result Has Been Sent On Your Mail","Result!",JOptionPane.INFORMATION\_MESSAGE);**

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

**this.dispose();**

**}**

**}**

**catch(SQLException ex){**

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

**ex.printStackTrace();**

**}**

**}**

**private int validateChooseOption()**

**{**

**if(jrbOption1.isSelected())**

**return 1;**

**else if(jrbOption2.isSelected())**

**return 2;**

**else if(jrbOption3.isSelected())**

**return 3;**

**else if(jrbOption4.isSelected())**

**return 4;**

**else**

**return 0;**

**}**

**private double getPercentage(){**

**int marksObtained=0;**

**for(int i=0;i<questionList.size();i++){**

**QuestionPojo question=questionList.get(i);**

**if(question.getCorrectOption()==answerList.get(i)){**

**++marksObtained;**

**}**

**}**

**double per=((double)marksObtained/questionList.size())\*100;**

**return per;**

**}**