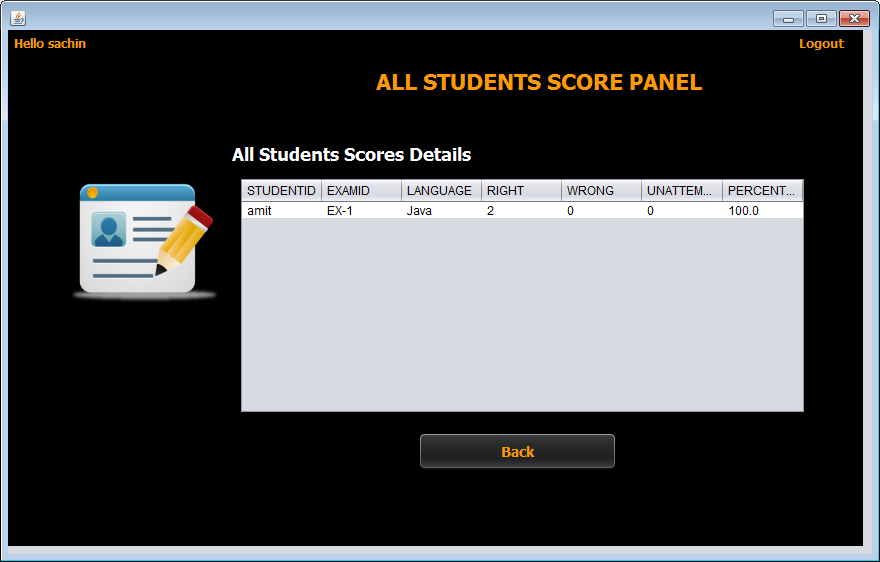
**Designing The ViewAllScoreFrame**

****

**STEPS TO BE DONE IN** **ViewAllScoreFrame**

In the **ViewAllScoreFrame** we need to do following steps:

1. Display **username** on the top left

2. Allow the user to **logout**

3. Handle the constructor so that it loads complete details of all the students who have appeared for test from the database. To do this we need to call the method **getAllData( )** of the class **PerformanceDAO**.

4. Handle the "**Back**" Button

**THE TABLES USED IN ViewAllScoreFrame**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **USERID** | **Varchar2(10)** | **Contains Admin Provided User ID** |
| **EXAMID** | **Varchar2(10)** | **Contains id of the paper** |
| **RIGHT** | **Number(4)** | **Number of correct answers** |
| **WRONG** | **Number(4)** | **Number of wrong answers** |
| **UNATTEMPTED** | **Number(5)** | **Number of unattempted questions** |
| **PER** | **Number(5,2)** | **Percentage obtained** |
| **LANGUAGE** | **Varchar2(10)** | **Programming language name** |

**THE POJO CLASSES USED IN ViewAllScoreFrame**

1. The **Performance** POJO

***public class Performance {***

***private String examId;***

***private String language;***

***private String userId;***

***private int right;***

***private int wrong;***

***private int unattempted;***

***private double per;***

***public String getExamId() {***

***return examId;***

***}***

***public void setExamId(String examId) {***

***this.examId = examId;***

***}***

***public String getLanguage() {***

***return language;***

***}***

***public void setLanguage(String language) {***

***this.language = language;***

***}***

***public String getUserId() {***

***return userId;***

***}***

***public void setUserId(String userId) {***

***this.userId = userId;***

***}***

***public int getRight() {***

***return right;***

***}***

***public void setRight(int right) {***

***this.right = right;***

***}***

***public int getWrong() {***

***return wrong;***

***}***

***public void setWrong(int wrong) {***

***this.wrong = wrong;***

***}***

***public int getUnattempted() {***

***return unattempted;***

***}***

***public void setUnattempted(int unattempted) {***

***this.unattempted = unattempted;***

***}***

***public double getPer() {***

***return per;***

***}***

***public void setPer(double per) {***

***this.per = per;***

***}***

***public Performance(String examId, String language, String userId, int right, int wrong, int unattempted, double per) {***

***this.examId = examId;***

***this.language = language;***

***this.userId = userId;***

***this.right = right;***

***this.wrong = wrong;***

***this.unattempted = unattempted;***

***this.per = per;***

***}***

***}***

**THE DAO CLASSES USED IN ViewAllScoreFrame**

1. The **PerformanceDAO**

**HOW TO LOAD AND SHOW ALL STUDENT EXAM DETAILS:**

To do this we need to create a method called **getAllData( )** in the class **PerformanceDAO**. This method should fetch and return details all the students from the table **PERFORMANCE.** This return value is a list of **Performance** POJO objects . So the prototype of this method will be:

***public static ArrayList<Performance> getAllData()throws SQLException***

Following is it's code:

***public static ArrayList<Performance> getAllData()throws SQLException{***

***String qry="Select \* from Performance";***

***ArrayList<Performance> performanceList=new ArrayList<Performance>();***

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

***Statement st=conn.createStatement();***

***ResultSet rs=st.executeQuery(qry);***

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

***String userId=rs.getString("userid");***

***String examId=rs.getString("examid");***

***String language=rs.getString("language");***

***int right=rs.getInt("right");***

***int wrong=rs.getInt("wrong");***

***int unattempted=rs.getInt("unattempted");***

***double per=rs.getDouble("per");***

***Performance p=new Performance(examId,language,userId,right,wrong,unattempted,per);***

***performanceList.add(p);***

***}***

***return performanceList;***

***}***

**DISPLAYING STUDENT DETAILS IN JTABLE:**

To display details of all the students we have to use another UI control called **JTable**. The **JTable** control allows us to display multiple rows of data. So we need to create a method called **showDataInTable()** in the **ViewAllScoreFrame** which does the following:

**a. Calls the method getAllData( ) of the class PerformanceDAO.**

**b. If no id is returned , it displays the message "No student has yet appeared for the exam"**

**c. Otherwise it iterates over the student list and displays it in the JTable**

**d. It also handles SQLException which the method getAllStudentId( ) of the class PerformanceDAO can throw**.

Following is it's code:

***public void showDataInTable(){***

***try***

***{***

***ArrayList<Performance> performanceList=PerformanceDAO.getAllData();***

***if(performanceList.isEmpty()==true)***

***JOptionPane.showMessageDialog(null, "Sorry! No student has yet given any exam","Error!",JOptionPane.INFORMATION\_MESSAGE);***

***else***

***{***

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

***Object[] rows=new Object[7];***

***for(Performance pobj: performanceList)***

***{***

***rows[0]=pobj.getUserId();***

***rows[1]=pobj.getExamId();***

***rows[2]=pobj.getLanguage();***

***rows[3]=pobj.getRight();***

***rows[4]=pobj.getWrong();***

***rows[5]=pobj.getUnattempted();***

***rows[6]=pobj.getPer();***

***model.addRow(rows);***

***}***

***}***

**HANDLING THE CONSTRUCTOR:**

From the body of constructor we would call the above method so as to show the students details as s soon as the frame opens

***public ViewAllScoresFrame() {***

***initComponents();***

***this.setLocationRelativeTo(null);***

***lblUsername.setText("Hello "+UserProfile.getUSername());***

***showDataInTable();***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN ViewAllScoresFrame**

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

***ViewScoresFrame viewScoresFrame =new ViewScoresFrame();***

***viewScoresFrame.setVisible(true);***

***this.dispose();***

***}***