**Criterion C (Development of Product)**

**List of Techniques used:**

• MySQL Databases – SQL commands

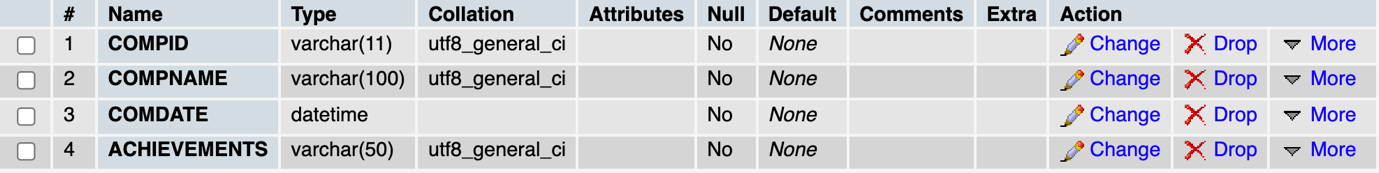
* Competition Details
* Login
* Student Workout
* Training Details

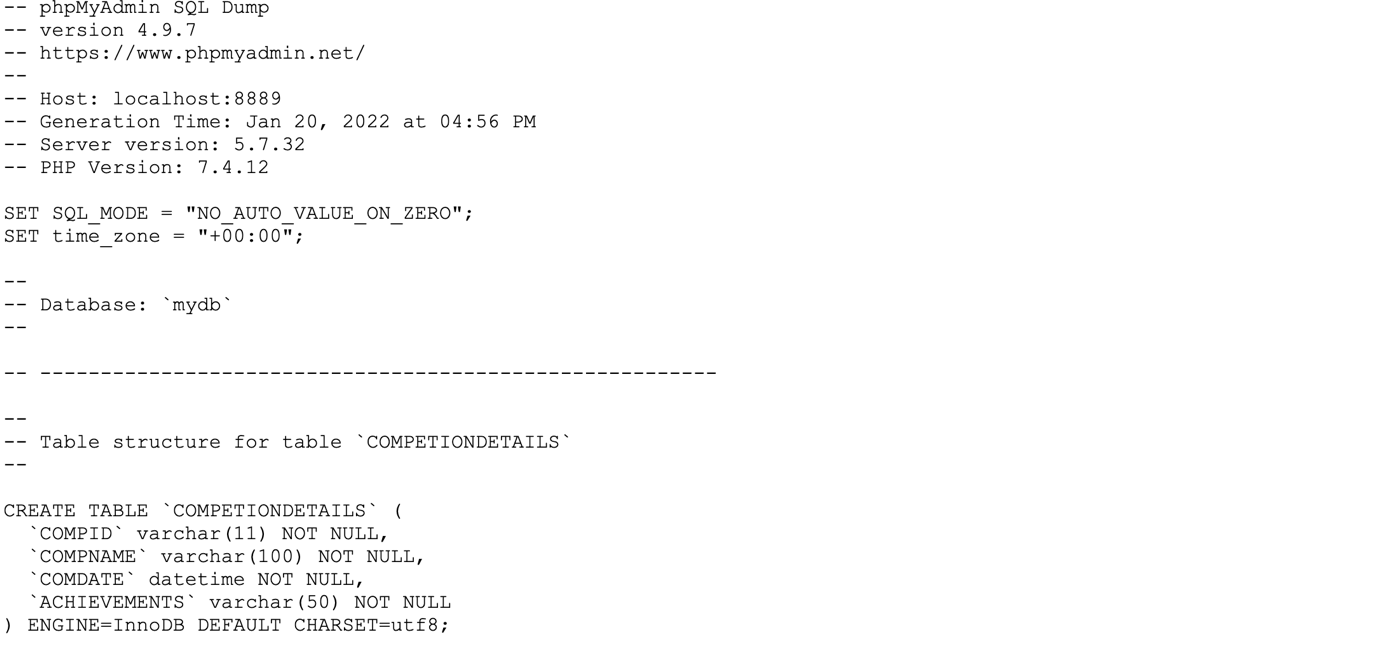
• GUI Programming – Use of Java Swing Controls along with Dialog boxes

* Login
* Register details
* Student Dashboard
* View Timings
* View Comp events
* View Timings
* Teacher Dashboard
* Edit/Add Competition
* Edit Student Details
* Add Home-workout
* Database Connectivity Steps
* External Libraries

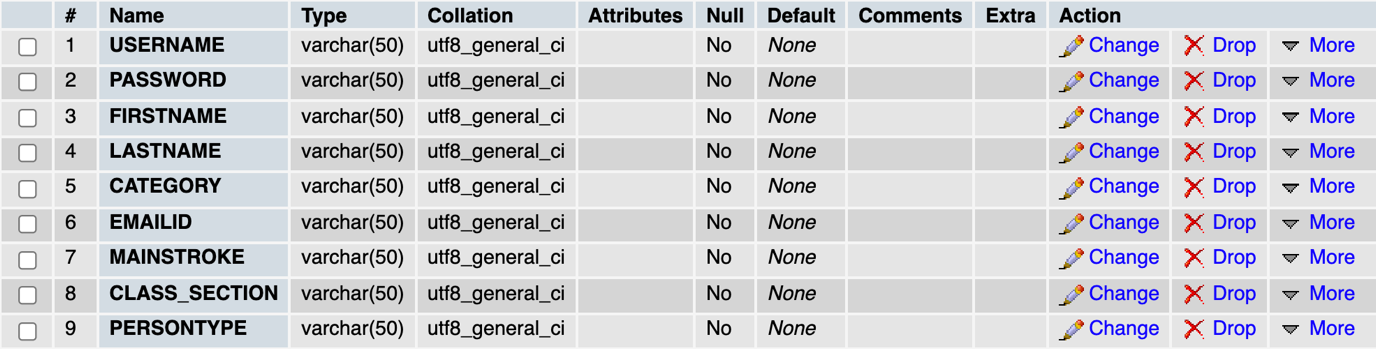
**MYSQL Database**

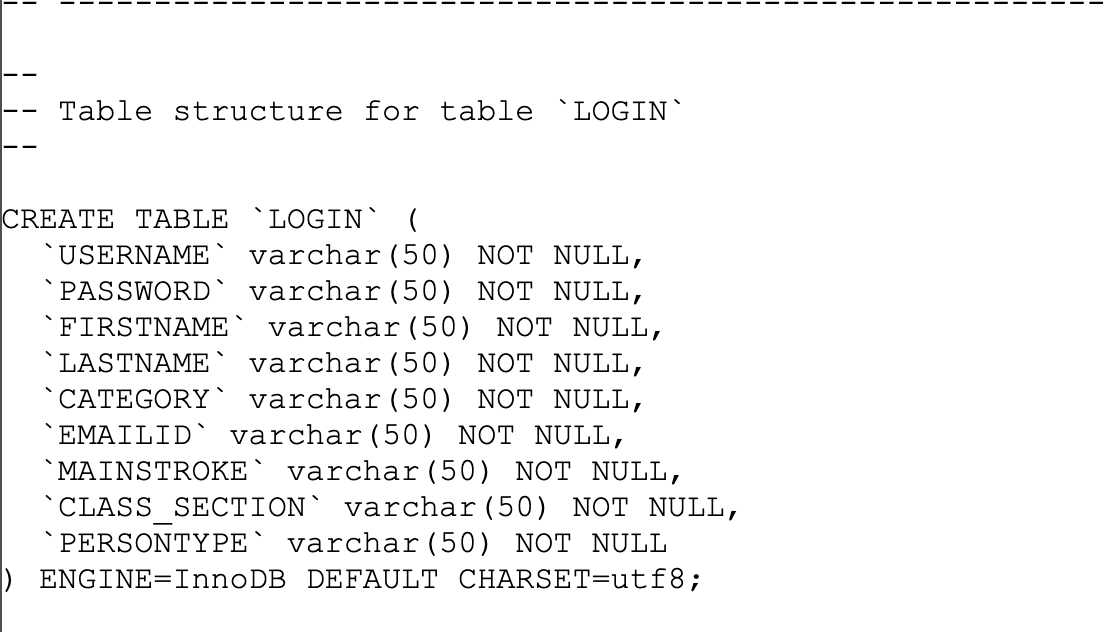
**Competition Details:**



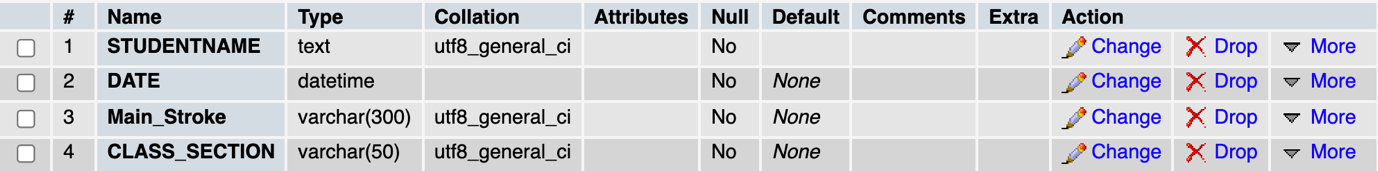


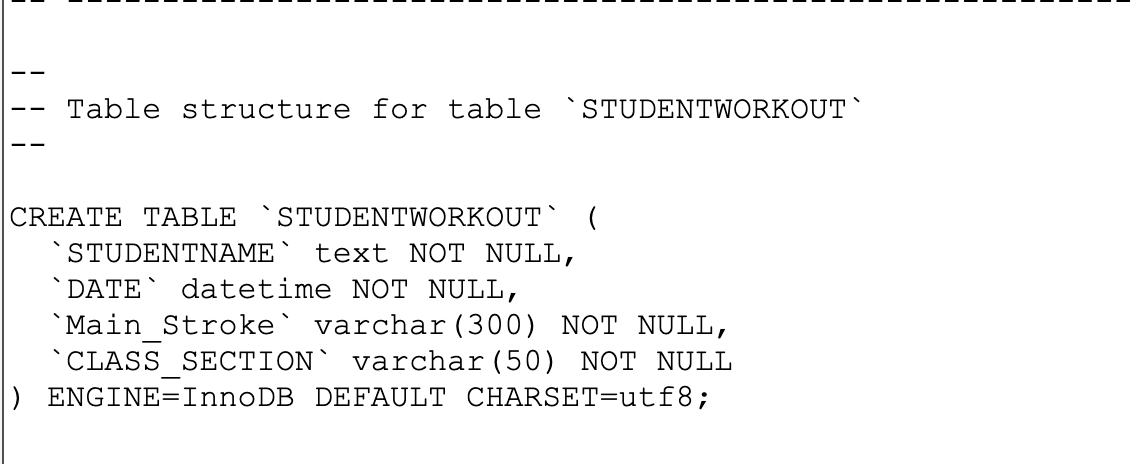
**Login:**



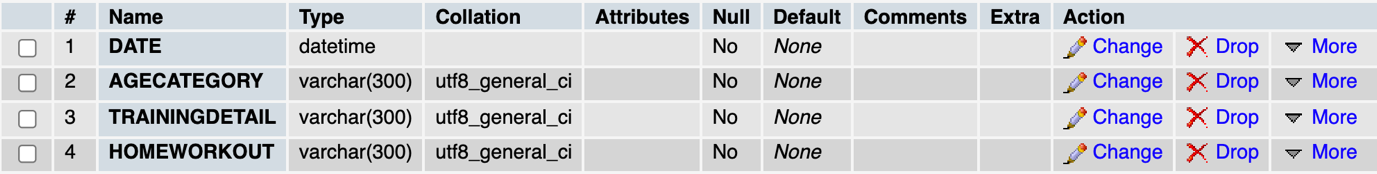


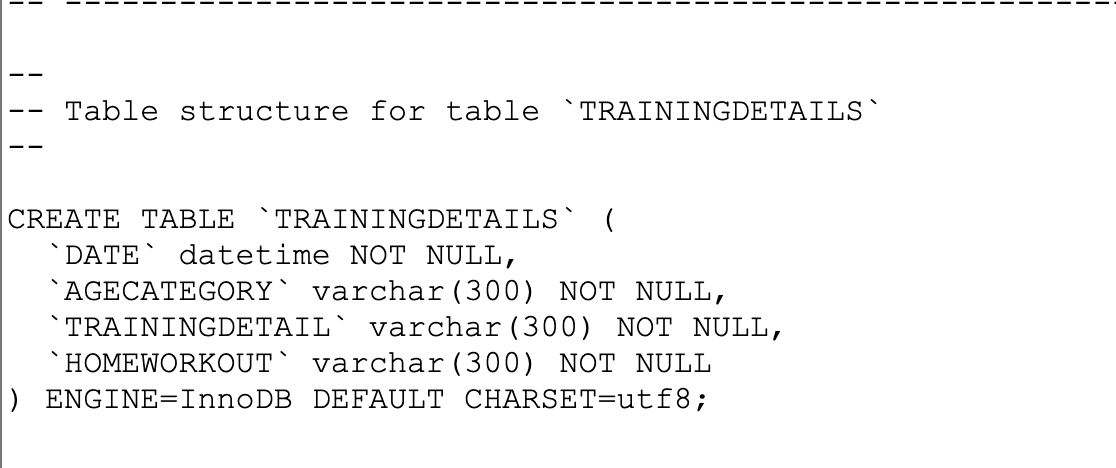
**Student Workout:**





**TrainingDetails:**



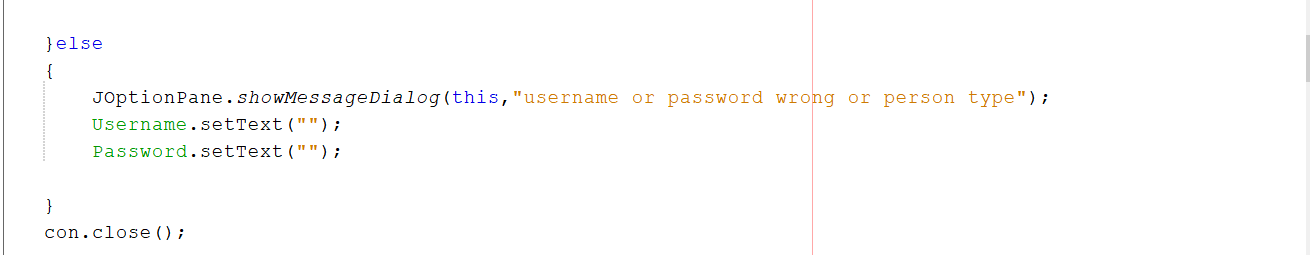
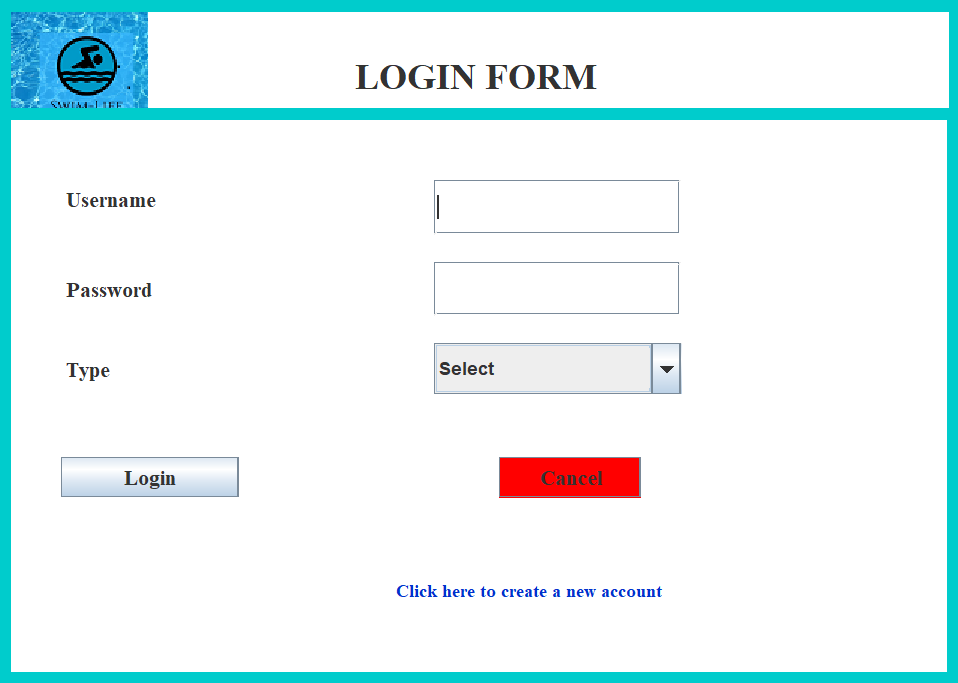


**Front End GUI**

**Logo:**

The logo is a Png image created by through canva and added through **labelcontrol** and **getscaledinstance()**

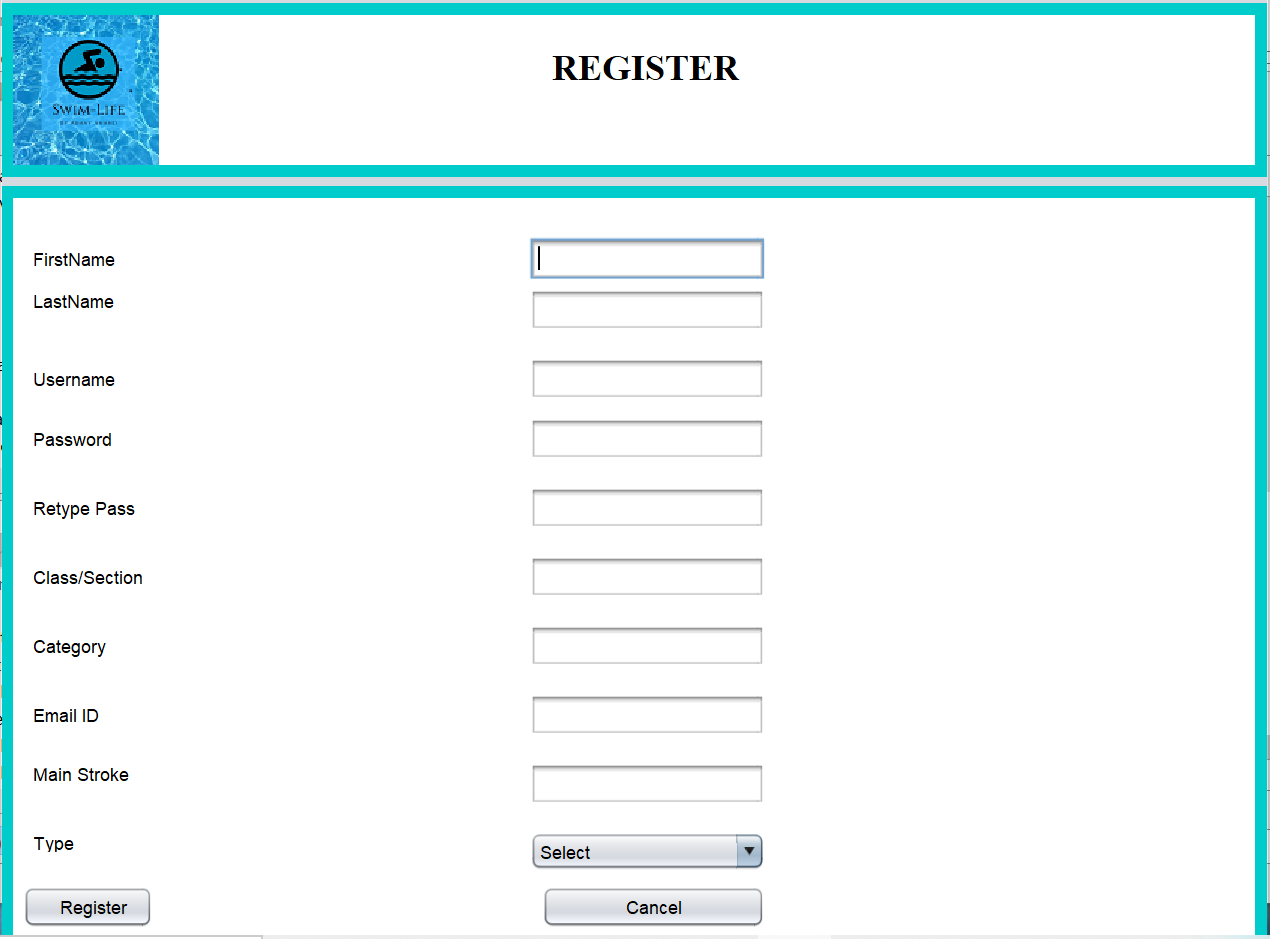
**Login GUI:**

  
**Procedure:**

**Login button**: If statement checks whether the combination of **username** and  **password** exists in the database. If the combination matches, it is checked whether the user is a Student or Coach based on which Coach or Student Dashboard is opened.   
Sign up form opens on the click of hypertexted link “**Click here to create a new account**”

* **username.getText()** , **password.getText()** and **ptype.getSelectedItem().toString()** methods are used to get the Username, Password and the Type(Coach/Student) from the user
* **con.createStatement()** is used to creates a statement object for sending SQL statements to the database
* **SQL** query has been executed to check the inputs with the database entries
* **If statement** is used to check the type and based on the type(Coach/Student) it will navigate to the desired page
* **Dispose()** is used to disappear and remove the resources it is using

**Register details GUI:**



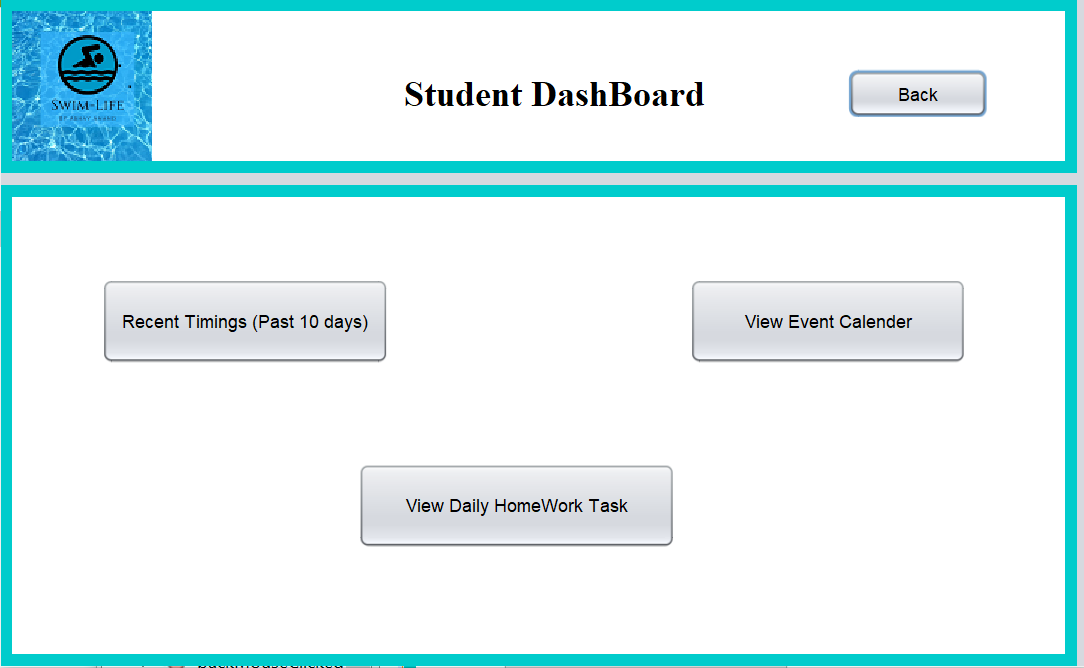


**Procedure:**

Firstly, only if password and re-entered password fields are same the record is added to Table – login. From the drop down Box of User Type, 2 options are provided – Student  or Coach. Accordingly, a value Student or resident will be added in the field of database record. Moreover other fields such as age category (U-17,U-15), class/section and Main Stroke are added to help the instructors select specific students later on. Since  usernameis the primary key for Table – user, if user enters a username that already exists, then  an exception with be encountered and an error message will be shown.

* **firstname.getText(), lastname.getText(), regusername.getText(), regpass.getText(), regcpass.getText(), regcategory.getText(), mailed.getText(), mainstroke.getText(), sptype.getSelectedItem().toString(), regclass.getText()** methods are used to get the required value from the user
* SQL **insert query** has been used to insert all the data into the login table
* **JOptionPane.showMessageDialog**() method is used to display the success and failure message

**Student Dashboard GUI:**





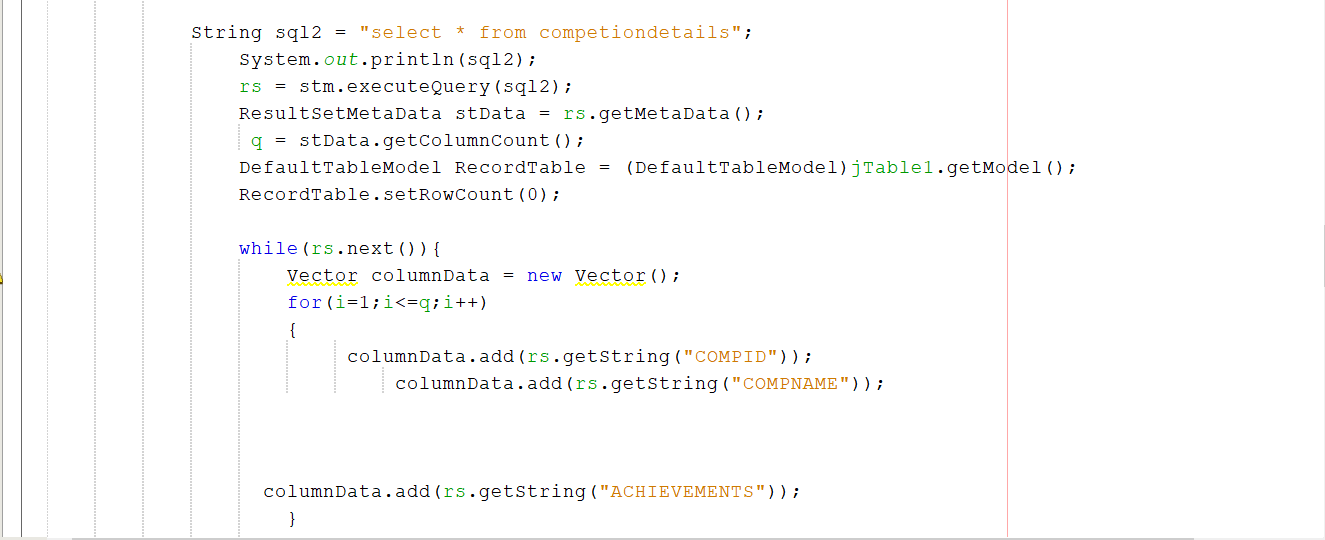
**Procedure:**

The Dashboard contains 3 buttons which each lead to a different screen, since this is the student dashboard it allows for students to view the specifics

* **JButtonMouseClicked()** creates object(event) for **viewevent()** class
* **Event.show()** method displays the Recent Timings, Event Calendar and Daily HomeWork Task

**View Timings GUI:**



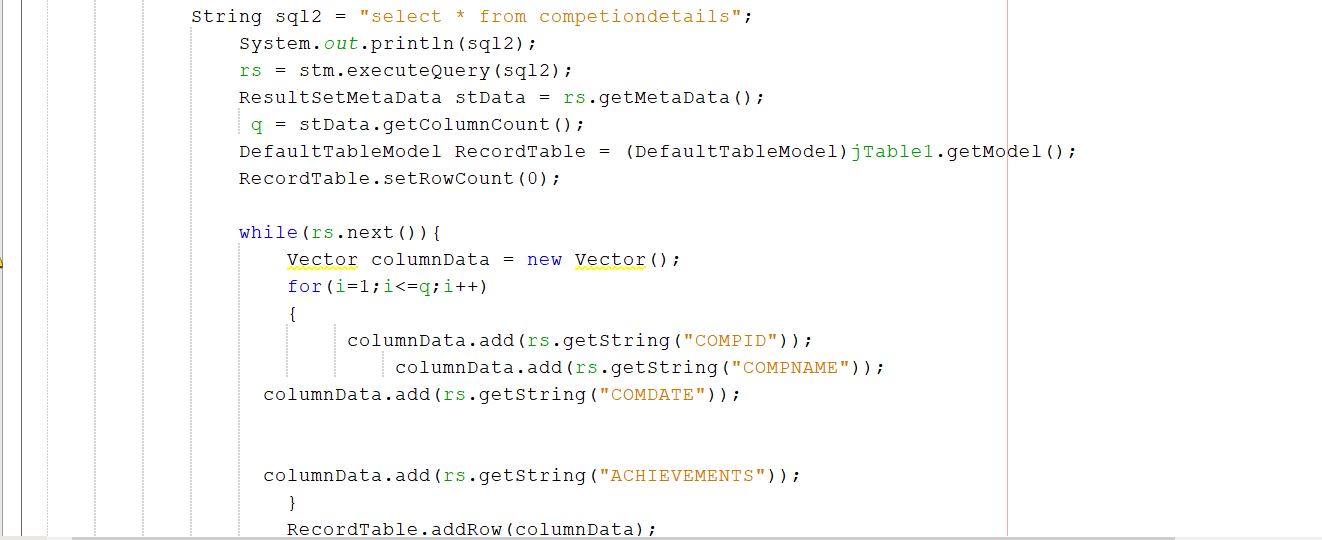
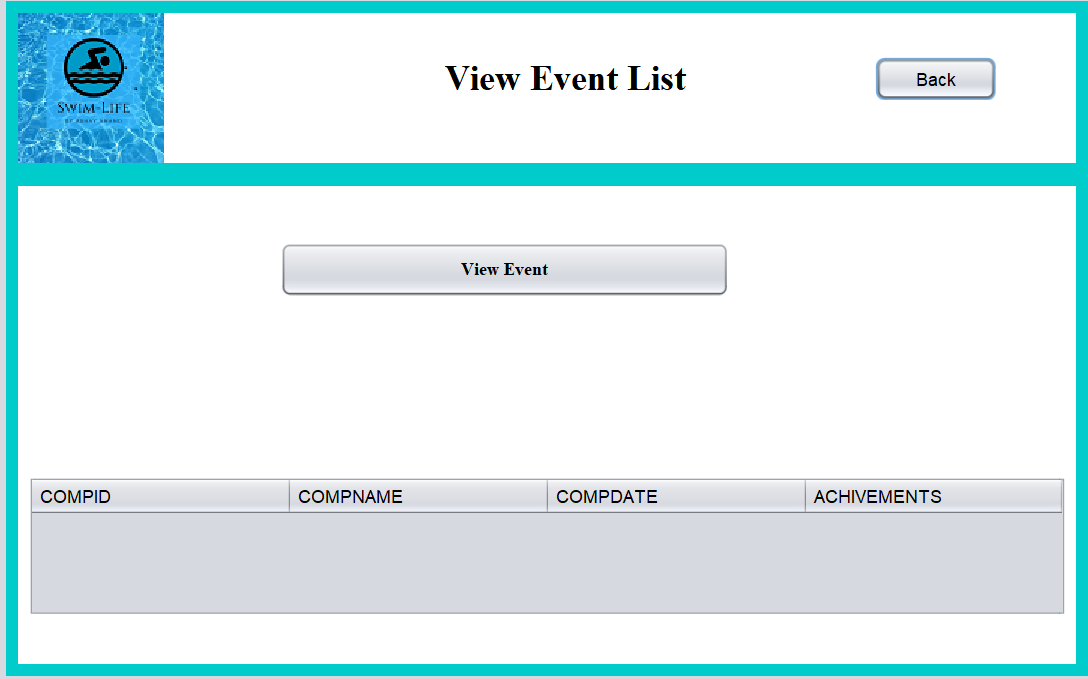


**Procedure:**

By clicking the button recent timings you can get the most recent entry in the timings you have performed during any recorded training sessions in the table along with the name of the event and other details

* **SQL** **select query** is used to get all Recent Timings data from the competiondetails table
* **DefaultTableModel()** class is used to create a table to display the data
* **Vector()** class is used to add the Recent Timing details to the table

**View Competition Events GUI:**

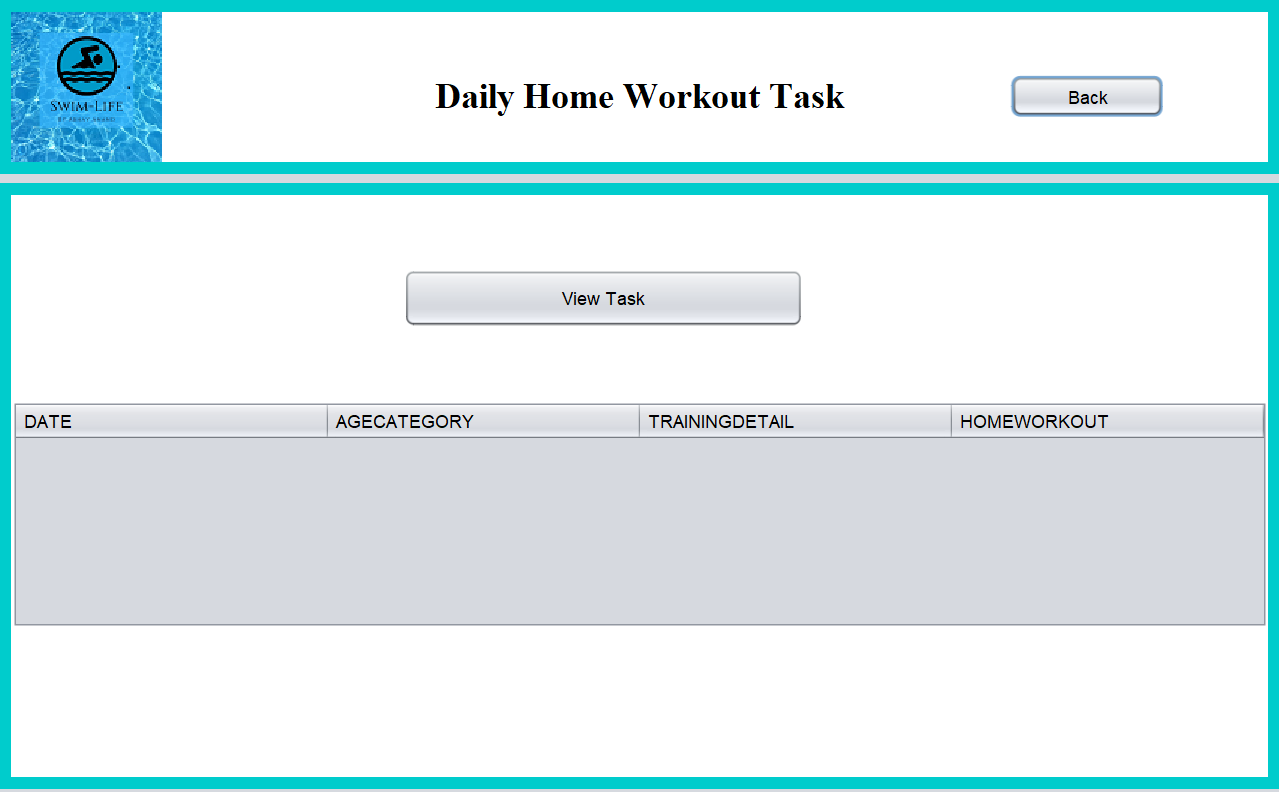


Procedure:

The view event button allows for the students to view the achievements the school one most recent Competition (recent entry) along with a specific comp ID, for example 0001 , and the date of the competition in date/time format.

* **SQL** **select query** is used to get all Recent Timings data from the competiondetails table
* **DefaultTableModel()** class is used to create a table to display the data
* **Vector()** class is used to add the Recent Timing details to the table

**View Home Workout GUI:**

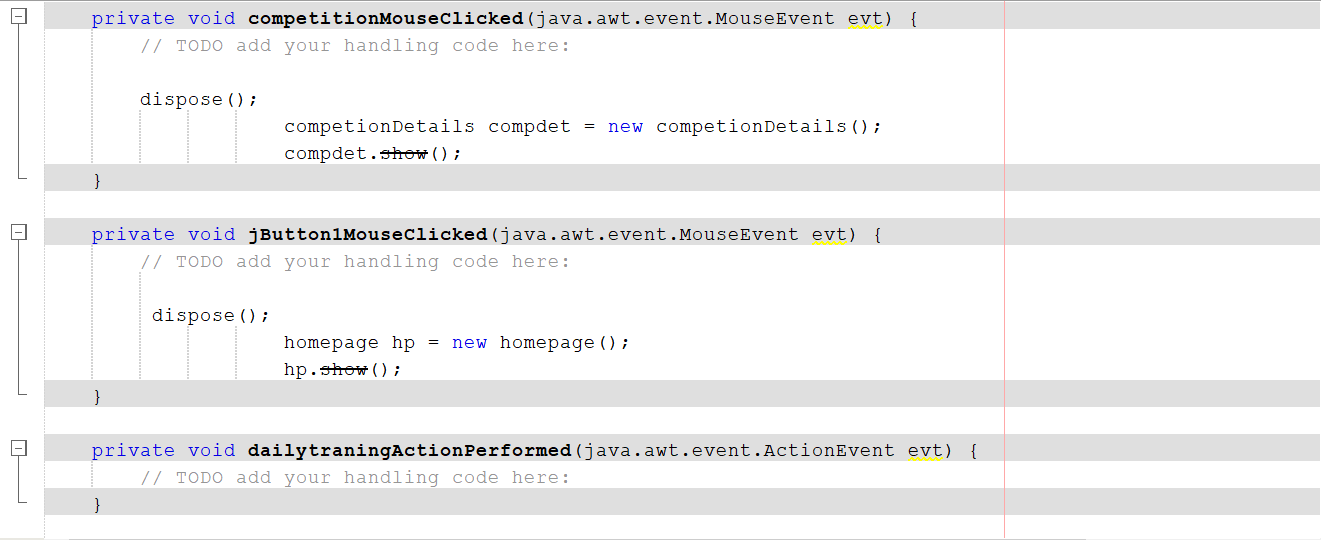
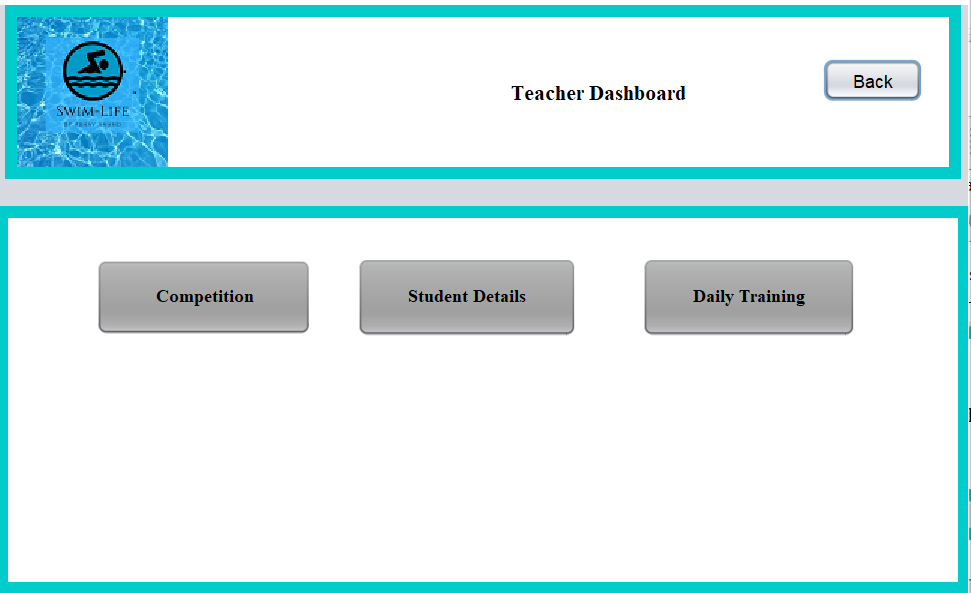


**Procedure:**

The view task button allows for the students to view the home workout the teachers posted on all specified dates. It also has columns for which specific age category should practice it and add the training performed during the morning practise if any.

* **SQL** **select query** is used to get all Recent Timings data from the trainingdetails table where the given date is equal to the current date
* **DefaultTableModel()** class is used to create a table to display the data
* **Vector()** class is used to add the Recent Timing details to the table
* **setRowCount()** method is used to set the starting index value of the table

**Teacher Dashboard GUI:**

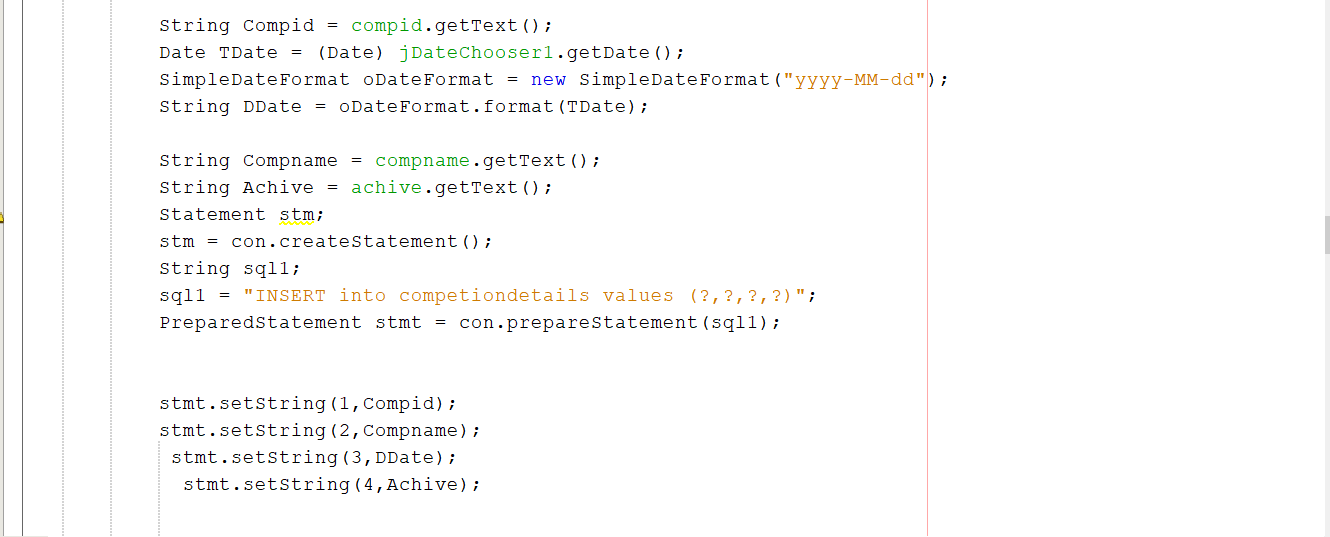
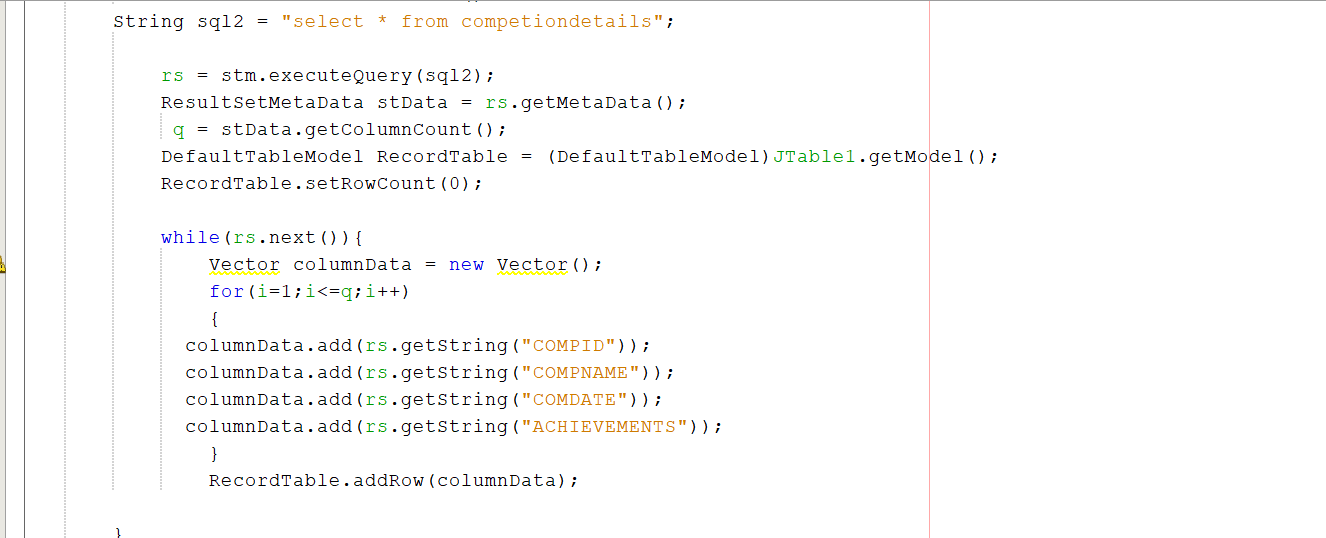


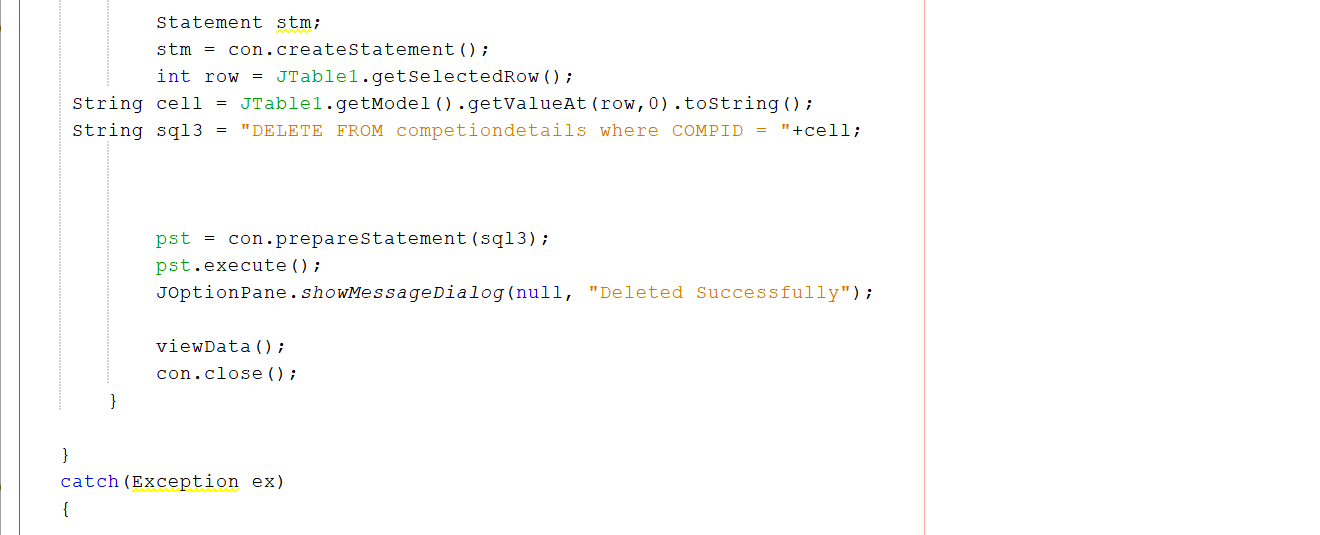
**Procedure**:

This Dashboard contains 3 buttons which each lead to a different screen however unlike the student dashboard, the coaches are able to edit and add elements instead of just view them

* **competitionMouseClicked(), jButtonMouseClicked(), dailytraningMouseClicked()** methods are used to display the required pages such as Competition, Student Details and Daily Training

**Edit/Add Competition Details GUI:**





**Procedure**:   
  
**Add Details**: This Buttons is used to add the specific fields – competition id, data,name and schools achievements to the database “Competition Details”

* **JDateChooser** library used to get the date in the same Date data type which will provide the GUI
* SQL **insert query** is used to insert the provided data to the table

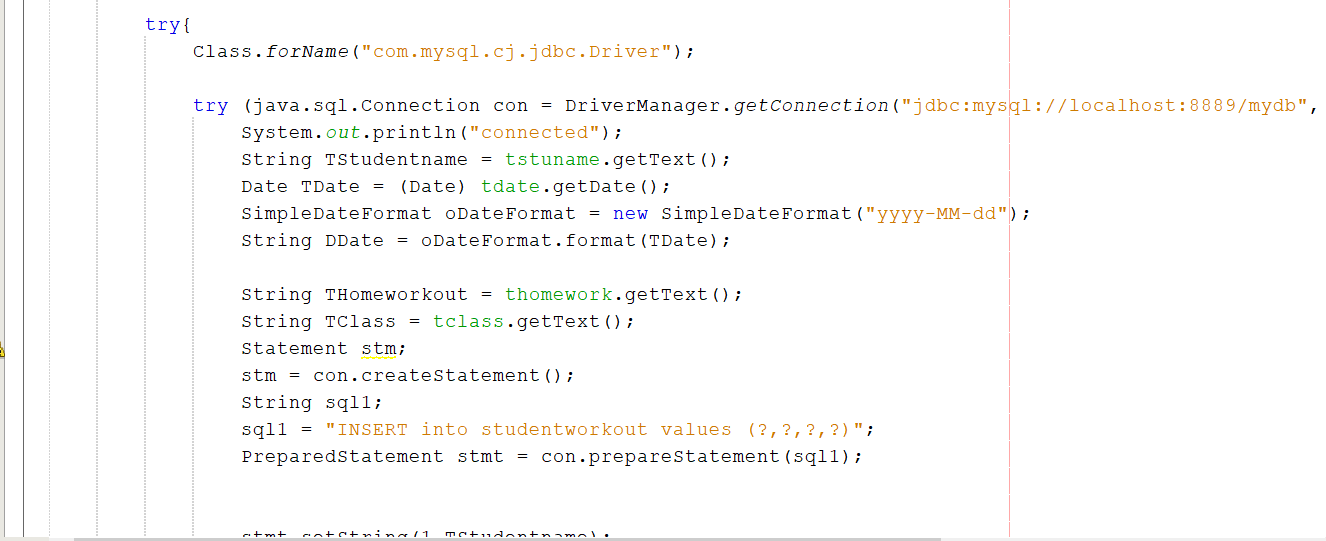
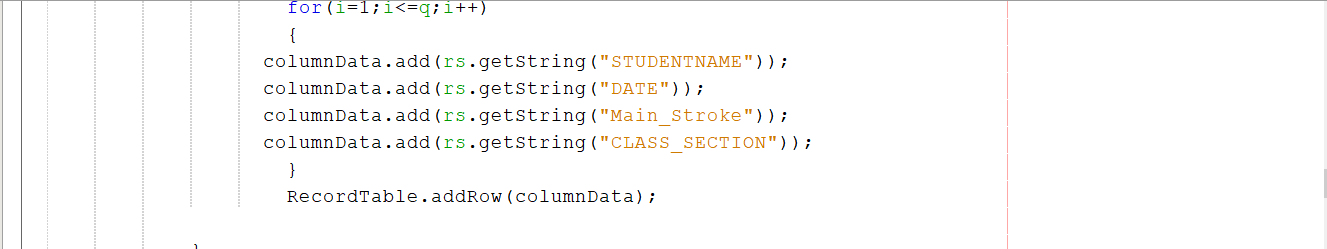
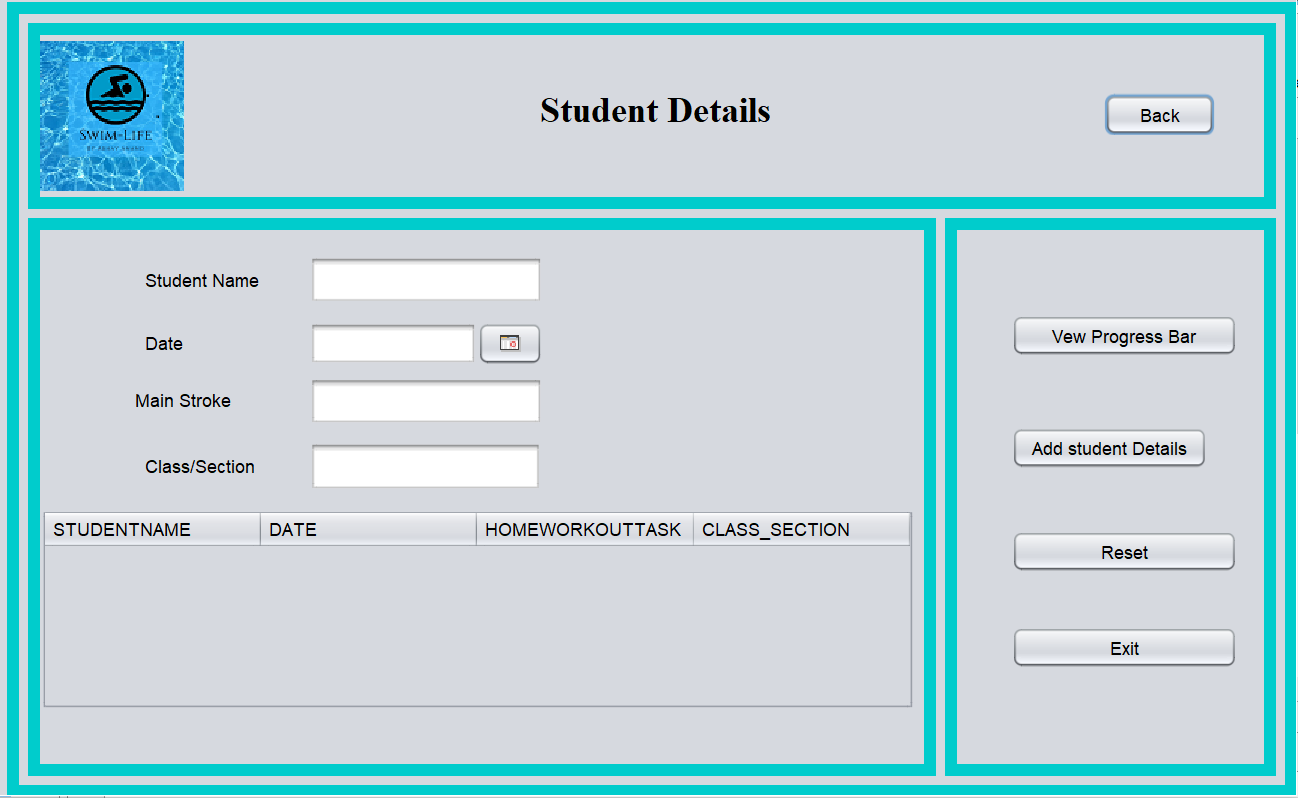
**Delete**: This Buttons is used to remove a row from the database “Competition Details”, which has the same value as any of the input put in the specific fields– competition id, data, name and schools achievements.

* SQL **delete query** used to remove the details from the database

**View Performance**: This Buttons is used to Output all data in the database “Competition Details”– competition id, data, name and schools achievements in the table made

* SQL **select query** is used to get all the details
* **DefaultTableModel**() class has been used for create a table
* **RecordTable.addRow()** method used for add the data to the table
* **viewData()** method is used to display all the details when the View Performance button have triggered

**Student details GUI:**



**Procedure:**

**Add Student Details**: This Buttons is used to add the specific fields – Student name, Birth date, main stroke and class/section to the database “Student Workout”

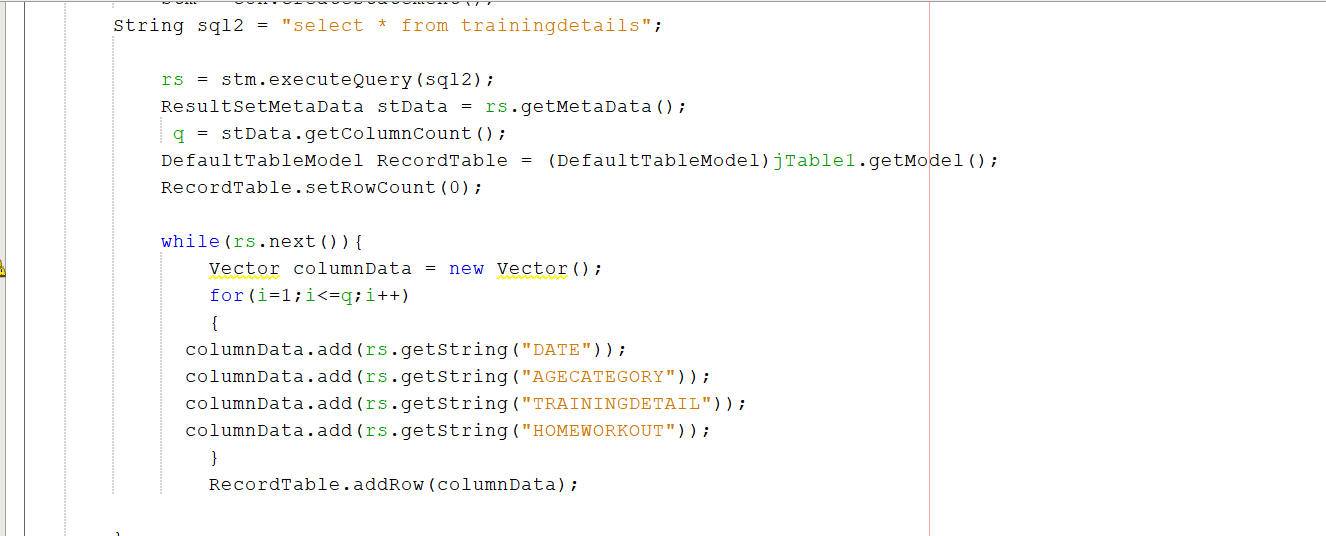
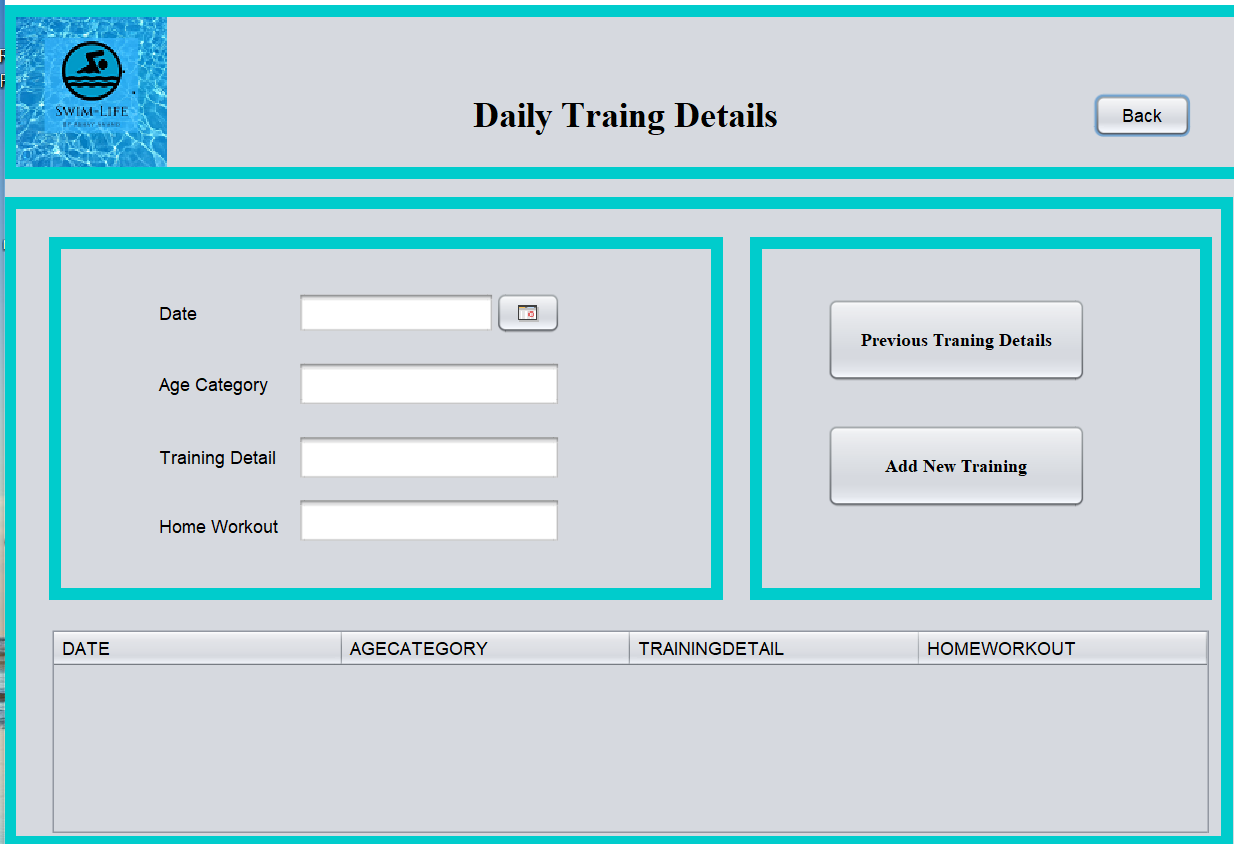
* **JDateChooser** library used to get the date in the same Date data type which will provide the GUI
* SQL **insert query** is used to insert the provided data to the table

**Reset**: This Buttons is used to reset the entered information in the text fields

**Exit**: This Buttons is used to exit the application and return to the home screen

**View Progress Bar:** This Buttons is used to Output data in the databases “Home workout”, “Student workout”– student name, birth date, home workout task and Class/section in the table mad

**Daily Training Details GUI:**



**Procedure:**

**Add New Training Details:** This Buttons is used to add the specific fields – age category, date, training detail and home workout to the database “TrainingDetails”

**Previous Training Details:** This Buttons is used to Output data in the databases “TrainingDetails” - age category, date, training detail and home workout in the table made.

**Main Database Connectivity Steps:**

In making a connection with database the following main steps were used:

1) Import java.sql package with all classes

2) Load JDBC driver using Class.forName("com.mysql.jdbc.Driver");

3) Open database connection using getConnection() method

Connection con =

DriverManager.getConnection("jdbc:mysql://localhost/cp2classes","root","Mysql@12 3");

4) Execute MySQL query add username and password (root)

5) Close database connection using close() method

**External Libraries used:**

• **Jcalender** package contains interface which allows user to enter date through calender and uses a calender.  
• **java**. **sql**.\* Provides the features for accessing and processing data stored in a relational  database using the **Java** programming language.

• **javax**.**swing**. Provides a set of "lightweight" components

**References:**

https://www.javatpoint.com/example-to-connect-to-the-mysql-database

https://www.mysqltutorial.org/connecting-to-mysql-using-jdbc-driver/

https://docs.oracle.com/javase/tutorial/uiswing/components/index.html

https://www.tutorialspoint.com/swing/swing\_controls.htm

https://www.javatpoint.com/java-swing

https://www.tutorialspoint.com/mysql/index.htm

https://dev.mysql.com/doc/refman/8.0/en/tutorial.html

https://docs.oracle.com/javase/7/docs/api/java/sql/package-use.html