ETES-204

Practical File

Submitted By:

Pandey Mohiya-Asu2013010100136

Chetan Gupta-Asu2013010100018

Prateek Verma-Asu2013010100052

Submitted to:

Mrs.Harsimran Kaur

CONTENT:

INTRODUCTION.

- ***Jdbc :***

- The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database.
- JDBC helps you to write Java applications that manage these three programming activities:

1. Connect to a data source, like a database
2. Send queries and update statements to the database
3. Retrieve and process the results received from the database in answer to your query

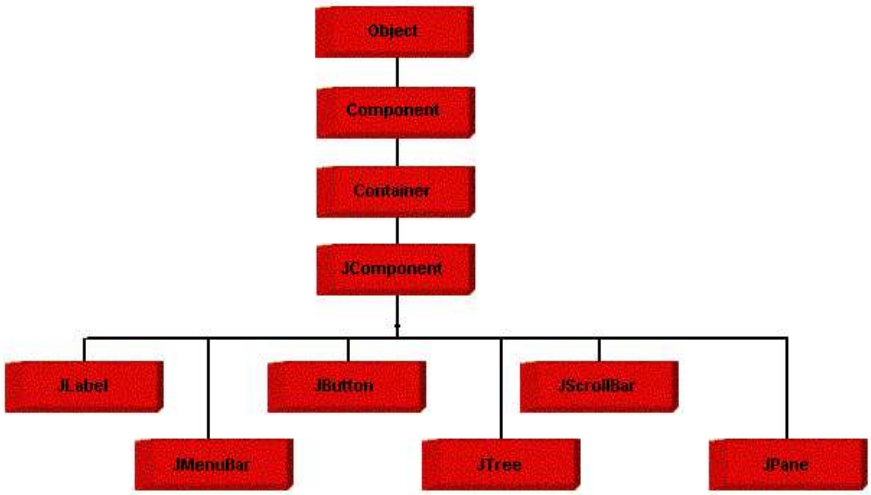
- The following simple code fragment gives a simple example of these three steps:

```
public void connectToAndQueryDatabase(String username, String password)
 {
Connection con = DriverManager.getConnection(
"jdbc:myDriver:myDatabase",
username,
password);

Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");

while (rs.next()) {
int x = rs.getInt("a");
String s = rs.getString("b");
float f = rs.getFloat("c");
}
}
```

- This short code fragment instantiates a DriverManager object to connect to a database driver and log into the database, instantiates a Statement object that carries your SQL language query to the database; instantiates a ResultSet object that retrieves the results of your query, and executes a simple while loop, which retrieves and displays those results. It's that simple.

- ***Swings :***

- Swing consists of elements or constituents that a developer will be making use of to come up with a Graphical User Interface. Let us take a look at some of the most commonly used elements or components of the Swing. The programming know-how of AWT is not at all required to equip with these swing programs.



- ***Awt:***

- The AWT provides four container classes. They are class Window and its two subtypes --class Frame and class Dialog -- as well as the Panel class. In addition to the containers provided by the AWT, the Applet class is a container -- it is a subtype of the Panel class and can therefore hold components. Brief descriptions of each container class provided by the AWT are provided below.

A top-level display surface (a window). An instance of the Window class is not attached to nor embedded within another container. An instance of the Window class has no border and no title.
A top-level display surface (a window) with a border and title. An instance of the Frame class may have a menu bar. It is otherwise very much like an instance of the Window class.

A top-level display surface (a window) with a border and title. An instance of the Dialog class cannot exist without an associated instance of the Frame class.

A generic container for holding components. An instance of the Panel class provides a container to which to add components.

- ***Events:***

Web pages are all about interaction. Users perform a countless number of actions such as moving their mice over the page, clicking on elements, and typing in textboxes — all of these are examples of events. In addition to these user events, there are a slew of others that occur, like when the page is loaded, when video begins playing or is paused, etc. Whenever something interesting occurs on the page, an event is fired, meaning that the browser basically announces that something has happened. It's this announcement that allows developers to "listen" for events and react to them appropriately.

Socket program.

Socket

A socket is one of the most fundamental technologies of computer networking. Sockets allow applications to communicate using standard mechanisms built into network hardware and operating systems. Although network software may seem to be a relatively new "Web" phenomenon, socket technology actually has been employed for roughly two decades.

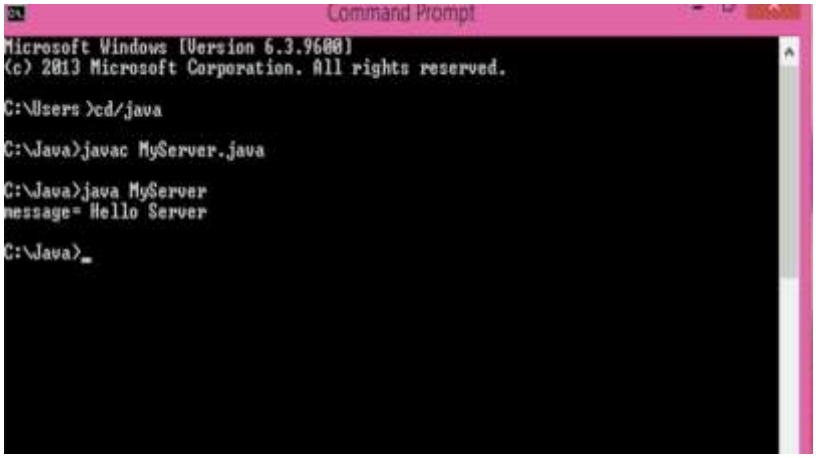
- ***My server***

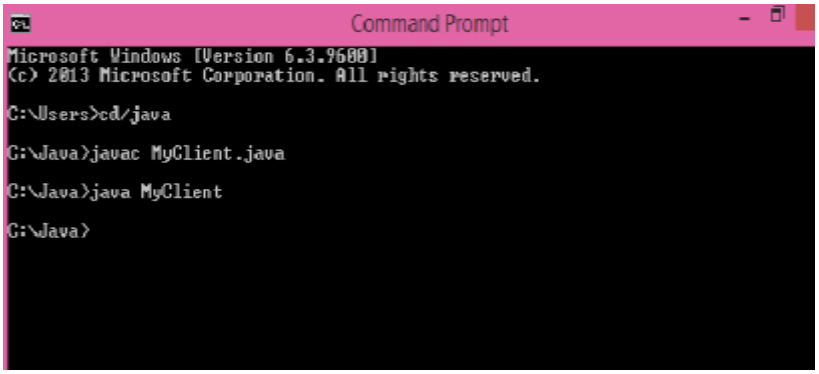
```
1. import java.io.*;
2. import java.net.*;
3. public class MyServer {
4. public static void main(String[] args){
5. try{
6. ServerSocket ss=new ServerSocket(1008);
7. Socket s=ss.accept();//establishes connection
8. DataInputStreamdis=new
   DataInputStream(s.getInputStrea
```

```
9.  m());
10. String  str=(String)dis.readUTF();
11. System.out.println("message= "+str);
12. ss.close();
13. }catch(Exception e){System.out.println(e);}
14. }
15. }
```

- **My client**

```
1.  import java.io.*;
2.  import java.net.*;
3.  public class MyClient {
4.  public static void main(String[] args) {
5.  try{
6.  Socket s=new Socket("USER-PC",1008);
7.  DataOutputStream dout=new DataOutputStream(s.getOutputStream());
8.  dout.writeUTF("Hello Server");
9.  dout.flush();
10. dout.close();
11. s.close();
12. }catch(Exception e){System.out.println(e);}
13. }
14. }
```





- **RMI program:**

  RMI (Remote Method Invocation) is a Java API for manipulating remote objects (e.g an object instantiated on another virtual machine, possibly on another machine on the network) in a transparent way, that is to say in the same way as if the object was located in the virtual machine (JVM) of the local machine.

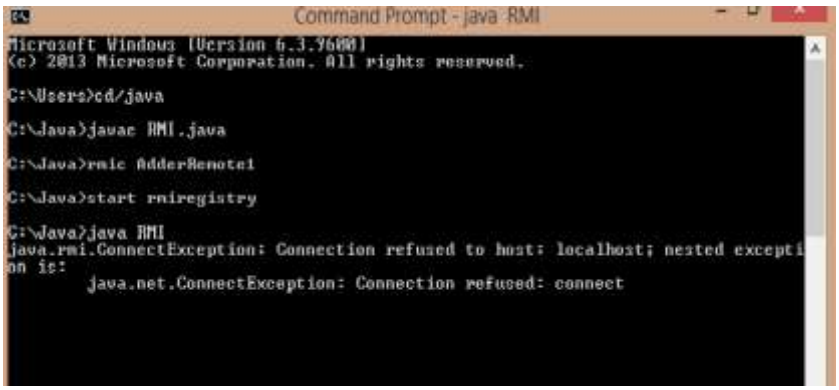
- **For Server:**

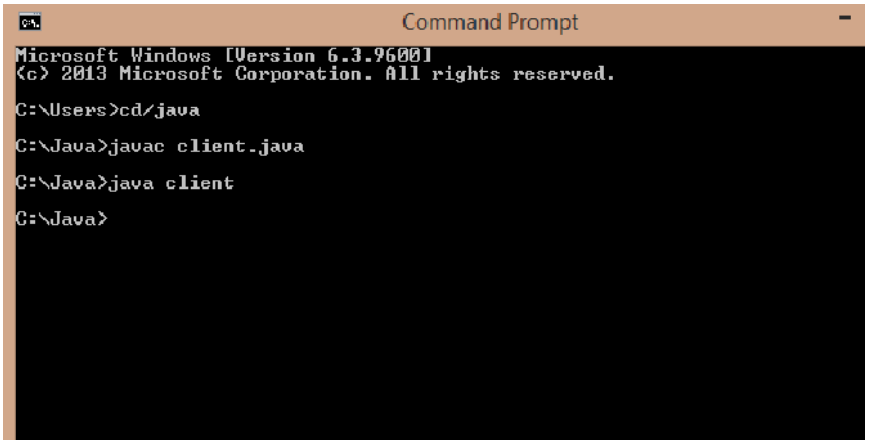
```
import java.rmi.*;
import java.rmi.registry.*;
public class RMI
{
public static void main (String args[])
{
try{
AdderRemote1 stub=new AdderRemote1();
Naming.rebind("rmi://localhost:1008/USER-PC",stub);
}catch(Exception e){System.out.println(e);}
}}
```

For client:

```
import java .rmi.*;
import java.rmi.registry.*;
public class client
{
public static void main(String args[])
{
try
{
```

```
21. AdderRemote stub=(AdderRemote)Naming.lookup("rmi://localhost:1008/USER-PC");
22. System.out.println(stub.add(34,4));
23. }catch(Exception e){}
24.}}
```





PROJECT-[DESIGNING A CAFETERIA SYSTEM]

AIM:

TO DEVELOP AN INTERACTIVE APPLICATION OF CAFETERIA MANAGEMENT SYSYTEM.

Components:

- **FRAME1[LOGIN]**

  - **BUTTON1[LOGIN]-**
  - **BUTTON2[SIGNUP]**
  - **BUTTON3[UPDATE]**
  - **BUTTON4[VIEW]**
  - **BUTTON5[LOGOUT]**
- **FRAME2[SIGNUP]**


SIGNUP
NAME:
marc
ENROLLMENTNO.:
1
PASSWORD:
CREDITS:
1000
STORE
DELETE
OK
view

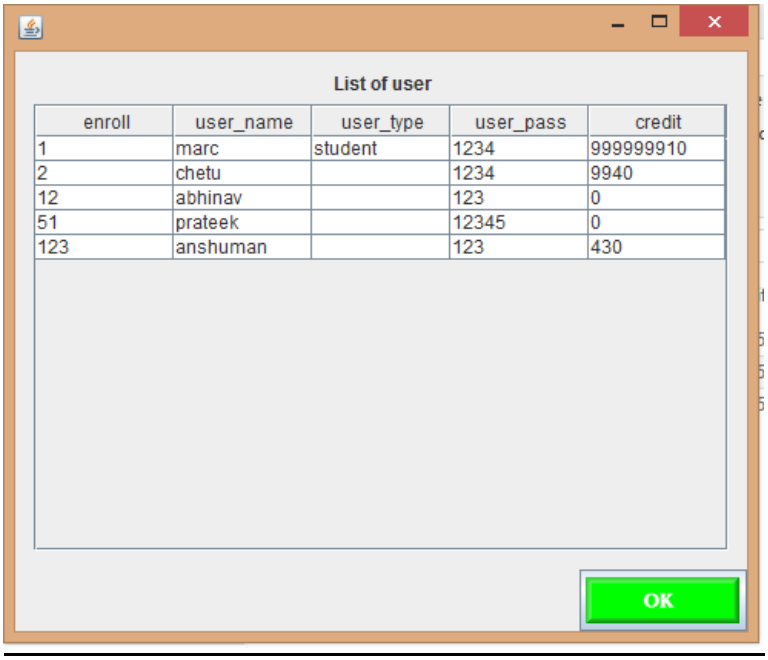

SIGNUP
NAME:
prateek
ENROLLMENTNO.:
51
PASSWORD:
CREDITS:
Message
You Missed a Field
OK
STORE
view

- **BUTTON1[STORE]**
- **BUTTON2[DELETE]**
- **BUTTON3[OK]**
- **BUTTON[VIEW]**

- **FRAME3[UPDATE]**

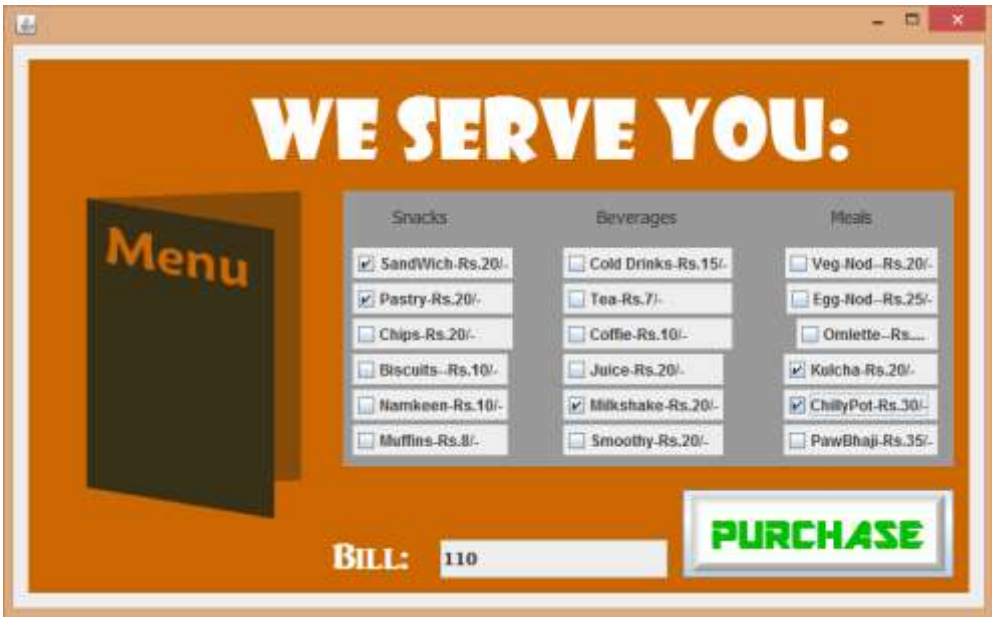
- **BUTTON1[UPDATE]**
- **BUTTON2[OK]**
- **BUTTON[VIEW]**

- **FRAME4[VIEW]**



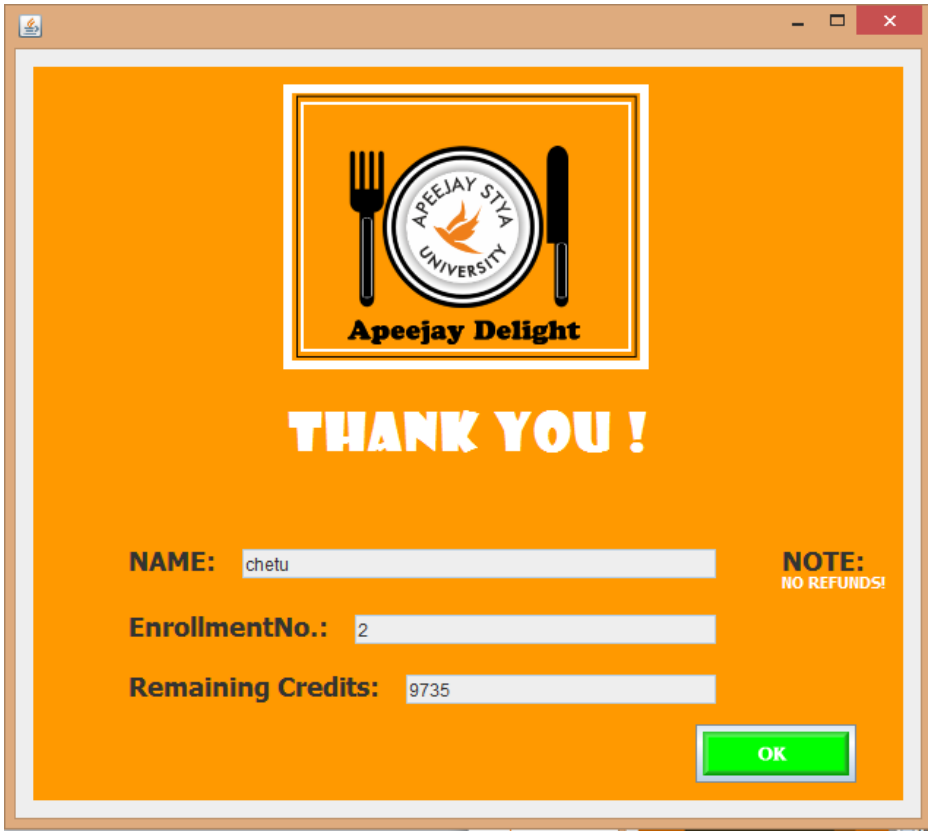
- **BUTTON1[OK]**

- **FRAME5[CAFE]**



- **TEXTBOX[BILL]**
- **BUTTON1[PURCHASE]**

- **FRAME6[THANKU]**



- **TEXTBOX1[NAME]**
- **TEXTBOX2[ENROLSLMENT NO]**
- **TEXTBOX3[REMAINING CREDITS]**
- **BUTTON[OK]**

Codings:

Components:

- **FRAME1[LOGIN]**
  - **BUTTON1[LOGIN]**

```java
225
226     private void b1ActionPerformed(java.awt.event.ActionEvent evt) {
227         // TODO add your handling code here:
228
229         String userName = getUsername();
230         String password = getPassword();
231         getUsername();
232         getPassword();
233
234
235          if(r1.isSelected())
236          {
237              String query = "SELECT *FROM  java_prog1 where user_name = '"+userName+"'";
238              String que="insert into whslog (user_name,user_pass) values ('"+userName+"','"+password+"')";
239              Connect con = new Connect();
240              PreparedStatement ps,ps2 ;
241              ResultSet rs;
242              try
243              {
244                  ps = con.getConnect().prepareStatement(query);
245                 ps2=con.getConnect().prepareStatement(que);
246
247                  rs = ps.executeQuery(query);
248                    ps2.executeUpdate(que);
249                  while(rs.next()){
250                  if(userName.equals(rs.getString("user_name")) && password.equals(rs.getString("user_pass")))
251              {
252                  cafe_frame m = new cafe_frame() ;
253                  m.setVisible(true);
254                  dispose();
255              }}
256              }
```

```java
289          }
290      }
291      public String getUsername()
292      {
293          String userName = t1.getText();
294          return userName;
295
296      }
297      public String getPassword()
298      {
299
300          String password = new String(pw1.getPassword());
301          return password;
302      }
303      private void b2ActionPerformed(java.awt.event.ActionEvent evt) {
304          // TODO add your handling code here:
305          signup_frame m2=new signup_frame();
306          m2.setVisible(true);
307          dispose();
308      }
309
310      private void b3ActionPerformed(java.awt.event.ActionEvent evt) {
311          // TODO add your handling code here:
312          update_frame m3=new update_frame();
313          m3.setVisible(true);
314          dispose();
315      }
316
317      private void b4ActionPerformed(java.awt.event.ActionEvent evt) {
318          // TODO add your handling code here:
319          View m4 = new View();
320          m4.setVisible(true);
```

```java
257              catch(SQLException e)
258              {
259                  System.out.println(e);
260              }
261          }
262          if(r2.isSelected())
263          {
264              String query = "SELECT *FROM  java_prog1 where user_name = '"+ userName+"'";
265              String que="insert into whslog (user_name,user_pass) values ('"+userName+"','"+password+"')";
266              Connect con = new Connect();
267              PreparedStatement ps,ps2 ;
268              ResultSet rs;
269              try
270              {
271                  ps = con.getConnect().prepareStatement(query);
272                 ps2=con.getConnect().prepareStatement(que);
273                  ps2.executeUpdate(que);
274                  rs = ps.executeQuery(query);
275
276
277                  while(rs.next()){
278                  if(userName.equals(rs.getString("user_name")) && password.equals(rs.getString("user_pass")))
279              {
280                  cafe_frame m1 = new cafe_frame() ;
281                  m1.setVisible(true);
282                  dispose();
283              }}
284              }
285              catch(SQLException e)
286              {
287                  System.out.println(e);
288              }
```

```java
        }
    }
    public String getUsername()
    {
        String userName = t1.getText();
        return userName;

    }
    public String getPassword()
    {

        String password = new String(pw1.getPassword());
        return password;
    }
```

- **BUTTON2[SIGNUP]**

```java
    private void b2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        signup_frame m2=new signup_frame();
        m2.setVisible(true);
        dispose();
    }
```

- **BUTTON3[UPDATE]**

```java
    private void b3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        update_frame m3=new update_frame();
        m3.setVisible(true);
        dispose();
    }
```

- **BUTTON4[VIEW]**

```java
    private void b4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        View m4 = new View();
        m4.setVisible(true);
        dispose();
    }
```

- **BUTTON5[LOGOUT]**

```java
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String query = "delete FROM  wholog; ";
            Connect con = new Connect();
            PreparedStatement ps ;

            try
            {
                ps = con.getConnect().prepareStatement(query);

                ps.executeUpdate(query);
            }


            catch(SQLException e)
            {
                System.out.println(e);
            }

    }
```

- **FRAME2[SIGNUP]**
  - **BUTTON1[STORE]**

```java
    private void b1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String name = t1.getText();
        String roll = t2.getText();
        String pass = new String(pw1.getPassword());
        String credit = t4.getText();


         String query = "insert into java_prog1 (enroll,user_name, user_pass, credit)"
                 + "values('"+roll+"', '"+name+"' , '"+pass+"','"+credit+"')";
         if(roll.equals("")||name.equals("")||pass.equals("")||credit.equals(""))
         {
             JOptionPane.showMessageDialog(this,"You Missed a Field");
         }
         else
         {
            Connect con = new Connect();
            PreparedStatement ps ;
            ResultSet rs;
            try
            {
                ps = con.getConnect().prepareStatement(query);

                ps.executeUpdate(query);

            }
            catch(SQLException e)
            {
                System.out.println(e);
            }

```

```java
            JOptionPane.showMessageDialog(this,"Added Successfully");
         }
    }
```

  - **BUTTON2[DELETE]**

```java
251     private void b2ActionPerformed(java.awt.event.ActionEvent evt) {
252         // TODO add your handling code here:
253         String roll = t2.getText();
254         String password = new String(pw1.getPassword());
255
256
257             String query = " select *FROM java_prog1 where enroll = '"+roll+"'";
258             Connect con = new Connect();
259             PreparedStatement ps ,ps2;
260             ResultSet rs;
261             try
262             {
263                 ps = con.getConnect().prepareStatement(query);
264
265                 rs =ps.executeQuery(query);
266                 while(rs.next())
267                 {
268                 if(roll.equals(rs.getString("enroll")) && password.equals(rs.getString("user_pass")))
269             {
270
271                 String en = t2.getText();
272                 String pass = new String(pw1.getPassword());
273                 String quer = " delete from java_prog1 where enroll = '"+en+"' && user_pass = '"+pass+"'";
274
275                 ps2 = con.getConnect().prepareStatement(quer);
276
277                 ps2.executeUpdate(quer);
278
279                 JOptionPane.showMessageDialog(this,"delete Successfully");
280
281             }
```

```java
282                 }
283             }
284             catch(SQLException e)
285             {
286                 System.out.println(e);
287             }
```

- **BUTTON3[OK]**

```java
244     private void b3ActionPerformed(java.awt.event.ActionEvent evt) {
245         // TODO add your handling code here:
246         login_frame m = new login_frame();
247         m.setVisible(true);
248         dispose();
249     }
```

- **BUTTON4[VIEW]::SAME AS BUTTON4 IN FRAME[LOGIN].**

- **FRAME3[UPDATE]**
  - **BUTTON1[UPDATE]**

```java
170
171     private void b1ActionPerformed(java.awt.event.ActionEvent evt) {
172         // TODO add your handling code here:
173         String roll = t1.getText();
174         String password = new String(t2.getPassword());
175
176
177             String query = "SELECT *FROM  java_prog1 where enroll = '"+roll+"'";
178             Connect con = new Connect();
179             PreparedStatement ps,ps2;
180             ResultSet rs;
181             try
182             {
183                 ps = con.getConnect().prepareStatement(query);
184
185                 rs = ps.executeQuery(query);
186                 while(rs.next()){
187                 if(roll.equals(rs.getString("enroll")) && password.equals(rs.getString("user_pass")))
188             {
189                 String cre = t3.getText();
190                 int credit=Integer.parseInt(cre);
191                 String en = t1.getText();
192                 String pass = new String(t2.getPassword());
193     String quer = " update java_prog1 set credit = '"+credit+"' where enroll = '"+en+"'&& user_pass ='"+pass+"'";
194                 ps2 = con.getConnect().prepareStatement(quer);
195
196                 ps2.executeUpdate(quer);
197                 JOptionPane.showMessageDialog(this,"Update Successful");
198             }}
199             }
```

```java
198                 }}
199                 }
200                 catch(SQLException e)
201                 {
202                     System.out.println(e);
203                 }
204
205     }
```

- **BUTTON2[OK]:: SAME AS BUTTON3 IN FRAME[SIGNUP].**
- **BUTTON[VIEW]:: SAME AS BUTTON4 IN FRAME[LOGIN].**

- **FRAME4[VIEW]**
  - **BUTTON1[OK]:: SAME AS BUTTON3 IN FRAME[SIGNUP].**
- **FRAME5[CAFE]**
  - **TEXTBOX[BILL]**

```java
465     private void jCheckBox8ActionPerformed(java.awt.event.ActionEvent evt) {
466         // TODO add your handling code here:
467         bill = bill + 7 ;
468         t1.setText(""+bill+"");
469     }
```

EVERY CHECKBOX HAVE CODE TO UPDATE BILL AND SET IT TO TEXTBOX

- **BUTTON1[PURCHASE]**

```java
    private void b1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String que = "SELECT *FROM  whclog";
            Connect con = new Connect();
            PreparedStatement ps1 ;
            ResultSet rs1;
            try
            {
                ps1=con.getConnect().prepareStatement(que);
                rs1 = ps1.executeQuery(que);
                while(rs1.next())
                {


        String userName = rs1.getString("user_name");
        String passWord = rs1.getString("user_pass");

        String query = "SELECT *FROM  java_prog1 where user_name = '"+userName+"' && user_pass = '"+passWord+"'";

            PreparedStatement ps ;
            ResultSet rs;
            try
            {
                ps = con.getConnect().prepareStatement(query);

                rs = ps.executeQuery(query);
                while(rs.next()){
                    String line = rs.getString("credit");
                    int store = Integer.parseInt(line);
                if(userName.equals(rs.getString("user_name")) && passWord.equals(rs.getString("user_pass")))
                {
```

```java
                {
                if(bill < store)
                {
                    store = store - bill;
                    String quer = "update java_prog1 set credit = '"+store+"' where user_name = '"+userName+"' &&

                      try
                      {
                        ps = con.getConnect().prepareStatement(quer);

                         ps.executeUpdate(quer);

                      }
                      catch(SQLException e)
                      {
                          System.out.println(e);
                      }
                      thanku_frame k = new thanku_frame();
                      k.setVisible(true);
                      dispose();
                }
                else
                {
                    JOptionPane.showMessageDialog(this, "you need to update your credits");
                    login_frame n = new login_frame();
                    n.setVisible(true);
                    dispose();
                }
            }})
            }
            catch(SQLException e)
```

```java
            catch(SQLException e)
            {
                System.out.println(e);
            }
                }}
                catch(SQLException e)
                        {
                        System.out.println(e);
                        }
    }
```

- **FRAME6[THANKU]**

- **BUTTON[OK]::**

```java
    private void b1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        login_frame m=new login_frame();
        m.setVisible(true);
        dispose();

        String query = "delete FROM  wholog; ";
            Connect con = new Connect();
            PreparedStatement ps ;

            try
            {
                ps = con.getConnect().prepareStatement(query);

                ps.executeUpdate(query);
            }


            catch(SQLException e)
            {
                System.out.println(e);
            }


    }
```