**Ex 4 Dt. 25/09/2013**

**GUI Programming with Gambas**

**Objective**

To develop a Graphical User Interface in GAMBAS, for a simple application. Use MySQL for database activities.

**Installation Procedure**

**GAMBAS:** It is a language with object extensions. A program written with Gambas is a set of files. Each file describes a class in terms of object programming.

Steps to install Gambas 3 are as follows

sudo add-apt-repository ppa:nemh/gambas3

sudo apt-get update

sudo apt-get install gambas3

**Steps to install MySQL server**

Step 1: Use sudo -i command to gain root access. Enter password when prompted.

Step 2: Then run the following command

apt-get install mysql-client mysql-server mysql-common

**Aim**

To develop a GUI based application to maintain the placement database using Gambas 3. To provide options to add student details, modify and delete student details. MySql database is used.

**Application Algorithm**

**Steps To Create A New Project in Gambas :**

1. Start Gambas.

2. Click New Project and Click Next.

3. Select Create a Graphical Project and Click Next.

4. Give a name.

5. For title also give the same name.

6. Click “OK” to create the Project.

**Steps To Create A Form :**

1. On your left side, you will see a tree view, right click on the Forms folder, choose New->Form and click ok in the create form dialog box.
2. The gray form will be displaced where you place the objects in it.
3. To your right, you should see the “ToolBox” where you choose what objects you want in the form and also set the properties to the object you create.

**Designing GUI For An Application (Form Design) :**

1. The required Labels and the textfields are created using the toolbox.
2. For Creating a label, Select [A] from properties and drag it to the form. In the

properties, you can set properties like changing the text, name , font, etc for the label.

1. Create textbox similarly by selecting [abc] textbox.
2. To give action to each button, double click on the button to open the Fmain.class

**To Open A Code Editor:**

1. Buttons such as “Add”, “Update”, “Delete” and “Exit” are created for each of the four operations using button toolbox.
2. Double click over the button for which the code is to be written.

**Creating Event Procedure:**

1. Code is divided in small blocks which is known as procedures.
2. Write the code between PUBLIC SUB AND END SUB.
3. Running the Application:
   1. To run the application , select Debug->Run from the Menu Bar (or)

click over the start icon on the Tool Box.

* 1. enabling the Database Access and MYSQL DB driver,
     1. Go to 🡪Project 🡪 Properties
     2. Go to 🡪 Component Tab
     3. Select gb.db.mysql, gb.db
     4. Click ok

**Program Code:**

**Server Side:**

' Gambas class file

Obj[10] As Socket

i As Integer

buffer As String

Public $Con As New Connection

subuf As String

Public Sub \_new()

End

Public Sub Form\_Open()

i = 0

End

Public Procedure Connect()

$Con.Close() ' Close the connection

$Con.Type = "MySQL" ' Type of connection

$Con.Host = "localhost" ' Name of the server

$Con.Login = "root" ' User's name for the connection

$Con.Port = "3306" ' Port to use in the connection, usually 3306

$Con.Name = "gambas" ' Name of the database we want to use

$Con.Password = "ssn" ' User's password

$Con.Open() ' Open the connection

Print $Con.Opened

End

Public Sub Button1\_Click()

ServSock.Type = Net.Internet

ServSock.Port = "34540"

ServSock.Listen(3)

Connect()

End

Public Sub ServSock\_Connection(buffer As String)

Obj[i] = ServSock.Accept()

Obj[i].Blocking = False

i = i + 1

Print "Request Received"

End

Public Sub Socket\_Read()

' When some data arrives to

' our server, we respond with

' an echo

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim element As String[]

Dim sub As String

If Last.Status <> Net.Connected Then Return

Read #Last, subuf, Lof(Last)

Print subuf

element = Split(subuf, "/")

For Each sub In element

Print "("; sub; ") ";

Next

If element[0] = "add" Then

Print "----add"

$Con.Begin()

$Con.Exec("INSERT INTO PlacementDB VALUES(&1,&2,&3,&4,&5,&6,&7,&8)", element[1], element[2], element[3], element[4], element[5], element[6], element[7], element[8])

$Con.Commit()

Message("Your details have been successfully added to the database.")

Else If element[0] = "update" Then

Print "update"

$Con.Begin()

$Con.Exec("UPDATE PlacementDB set name='" & element[2] & "',dob='" & element[3] & "',gender='" & element[4] & "',address='" & element[5] & "',mobile='" & element[6] & "',cgpa='" & element[7] & "',arrears='" & element[8] & "' where rollno='" & element[1] & "'")

$Con.Commit()

Message("The student record has been updated")

Else If element[0] = "delete" Then

Print "delete"

$Con.Begin()

$Con.Exec("delete from PlacementDB where rollno='" & element[1] & "'")

$Con.Commit()

Message("The student details have been deleted.")

End If

End

**Client Side:**

Class: add\_student

' Gambas class file

details As String

p\_rollno As String

p\_name As String

p\_dob As String

p\_gender As String

p\_address As String

p\_mobile As String

p\_cgpa As String

p\_arrears As String

Public Sub Form\_Open()

asocket.Connect("127.0.0.1", 34540)

End

Public Sub Button2\_Click()

Close asocket

add\_student.Close

End

Public Sub Button1\_Click()

p\_rollno = rollno.Text

p\_name = name.Text

p\_dob = dob.Text

p\_gender = gender.Text

p\_address = address.Text

p\_mobile = mobile.Text

p\_cgpa = cgpa.Text

p\_arrears = arrears.Text

details = "add" &/ p\_rollno &/ p\_name &/ p\_dob &/ p\_gender &/ p\_address &/ p\_mobile &/ p\_cgpa &/ p\_arrears

' Print album\_id

' Print album\_name

' Print album\_price

'Print details

' Write #asocket, "add", 3

Write #asocket, details, Len(details)

End

Class: modify\_student

' Gambas class file

Public cb As String

Public Sub Form\_Open()

usocket.Connect("127.0.0.1", 34540)

End

Public Sub Button1\_Click()

update\_student.Close

End

Public Sub Button2\_Click()

Dim p\_rollno As String

Dim p\_name As String

Dim p\_dob As String

Dim p\_gender As String

Dim p\_address As String

Dim p\_mobile As String

Dim p\_cgpa As String

Dim p\_arrears As String

Dim details As String

p\_rollno = rollno\_combo.Text

p\_name = name.Text

p\_dob = dob.Text

p\_gender = gender.Text

p\_address = address.Text

p\_mobile = mobile.Text

p\_cgpa = cgpa.Text

p\_arrears = arrears.Text

details = "update" &/ p\_rollno &/ p\_name &/ p\_dob &/ p\_gender &/ p\_address &/ p\_mobile &/ p\_cgpa &/ p\_arrears

'Print details

' Write #asocket, "add", 3

Write #usocket, details, Len(details)

End

Public Sub usocket\_Ready()

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' When connection proccess has finished

' successfully, "Connected" event will raise

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Print "connected"

cb = "values"

Write #usocket, cb, Len(cb)

End

Public Sub usocket\_Read()

Dim rolls As String

Dim element As String[]

Dim sub As String

Dim allval As String

If usocket.Status <> Net.Connected Then Return

Read #usocket, rolls, Lof(usocket)

Print rolls

element = Split(rolls, "/")

If element[0] = "values" Then

Print "values"

rollno\_combo.Text = element[1]

name.Text = element[2]

dob.Text = element[3]

gender.Text = element[4]

address.Text = element[5]

mobile.Text = element[6]

cgpa.Text = element[7]

arrears.Text = element[8]

Else

For Each sub In element

rollno\_combo.Add(sub)

Next

Endif

End

Public Sub id\_combo\_Change()

Dim cbv As String

cbv = "val" &/ rollno\_combo.Text

Print cbv

'cbv = regcb.Text

Write #usocket, cbv, Len(cbv)

Print "combo"

End

Class: delete\_student

' Gambas class file

Public Sub Label1\_MouseDown()

End

Public Sub Form\_Open()

dsocket.Connect("127.0.0.1", 34540)

End

Public Sub Button1\_Click()

delete\_student.Close

End

Public Sub dsocket\_Ready()

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' When connection proccess has finished

' successfully, "Connected" event will raise

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim cb As String

Print "connected"

cb = "values"

Write #dsocket, cb, Len(cb)

End

Public Sub dsocket\_Read()

Dim rollno As String

Dim element As String[]

Dim sub As String

Dim allval As String

If dsocket.Status <> Net.Connected Then Return

Read #dsocket, rollno, Lof(dsocket)

Print rollno

element = Split(rollno, "/")

For Each sub In element

rollno\_combo.Add(sub)

Next

End

Public Sub Button2\_Click()

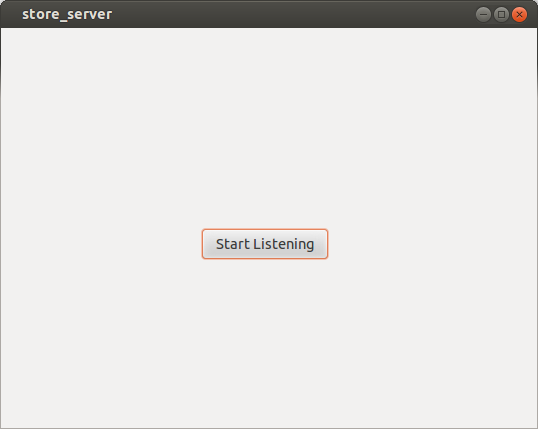
Dim id As String

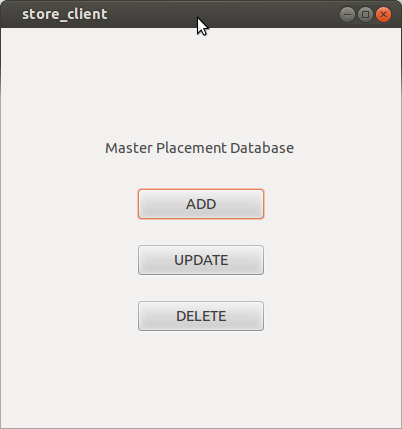
id = "delete" &/ rollno\_combo.Text

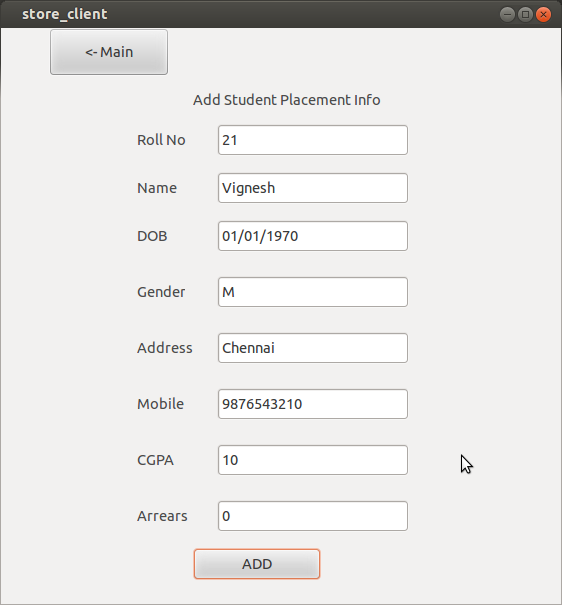
Write #dsocket, id, Len(id)

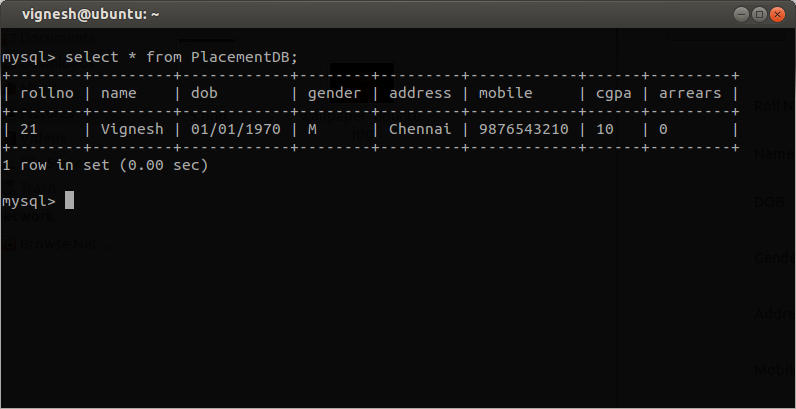
End

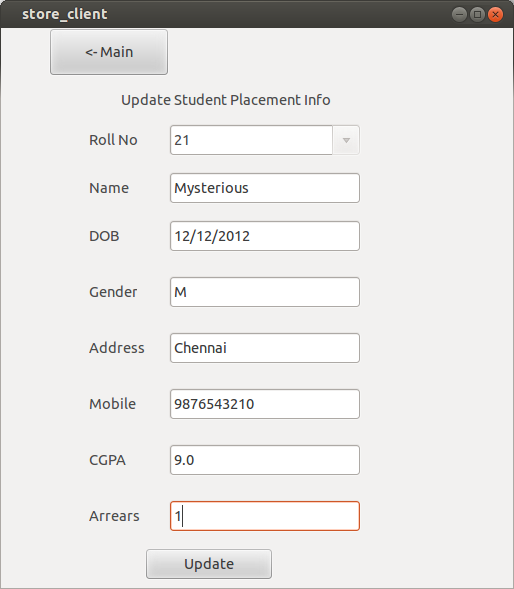
**Output Screenshots:**

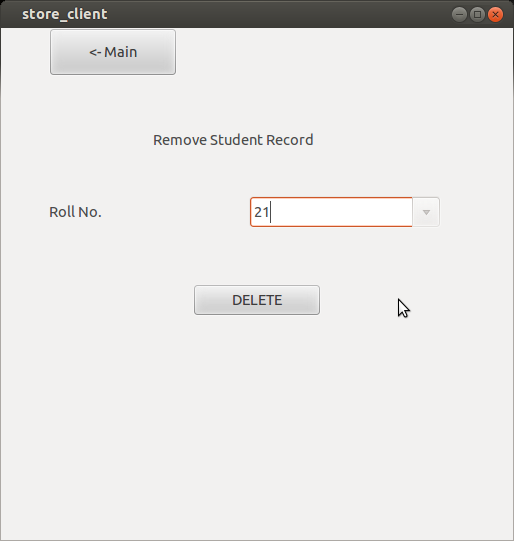
****

****

****

****

****

****

**Result**

The Placement Database Application is created using Gambas and the form is designed using labels, textbox, buttons etc. The properties are set like changing the text, font, color, height etc for each object created in the form. MYSQL connection is enabled and connection is made. For every click action on the button, changes are applied in the form and simultaneously in the database also using Gambas code.