Lab Instructions and Solution for Module- 3: Use of MySQL Database, connect C# Project to MySQL Database,

Objectives

After completing this module you will be able to:

* MySQL Database Setup and use of PhpMyAdmin.
* how to use and connect C# with MySQL using **MySQL Connect/NET**
* Connect Database with our Project Grading System.
* Doing and forwarding our project work.

Task 1: Installation of XAMPP/WAMPP

XAMPP/ WAMPP is a software which is for doing several software module installation like

- Appache
- MySQL database
- PHP
- Filezilla etc.

XAMPP combines many different software packages into one package. Here's an overview of all packages.

Apache HTTPD, MySQL, PHP, Perl, FileZilla FTP Server, phpMyAdmin, OpenSSL, Freetype, Webalizer, mod_perl, eAccelerator, mcrypt, SQLite, Mercury Mail Transport System, fake sendmail for windows, FPDF Class

Task 2: Using MySQL Database on XAMPP/WAMPP

Steps:

1. Open your browser and type on web address bar : **localhost/**
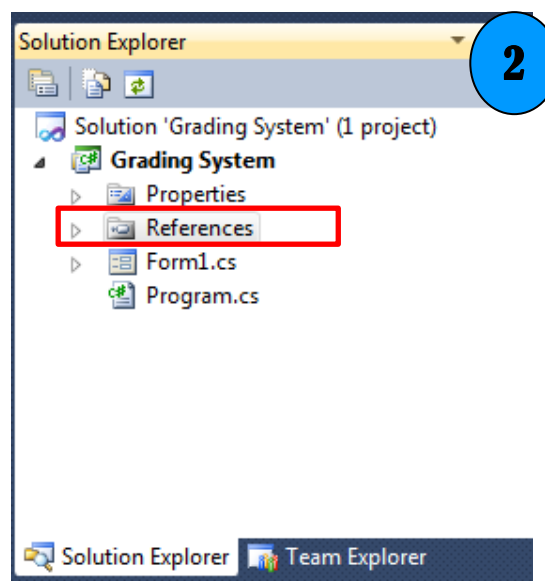2. Here You see all the software packages on xampp.

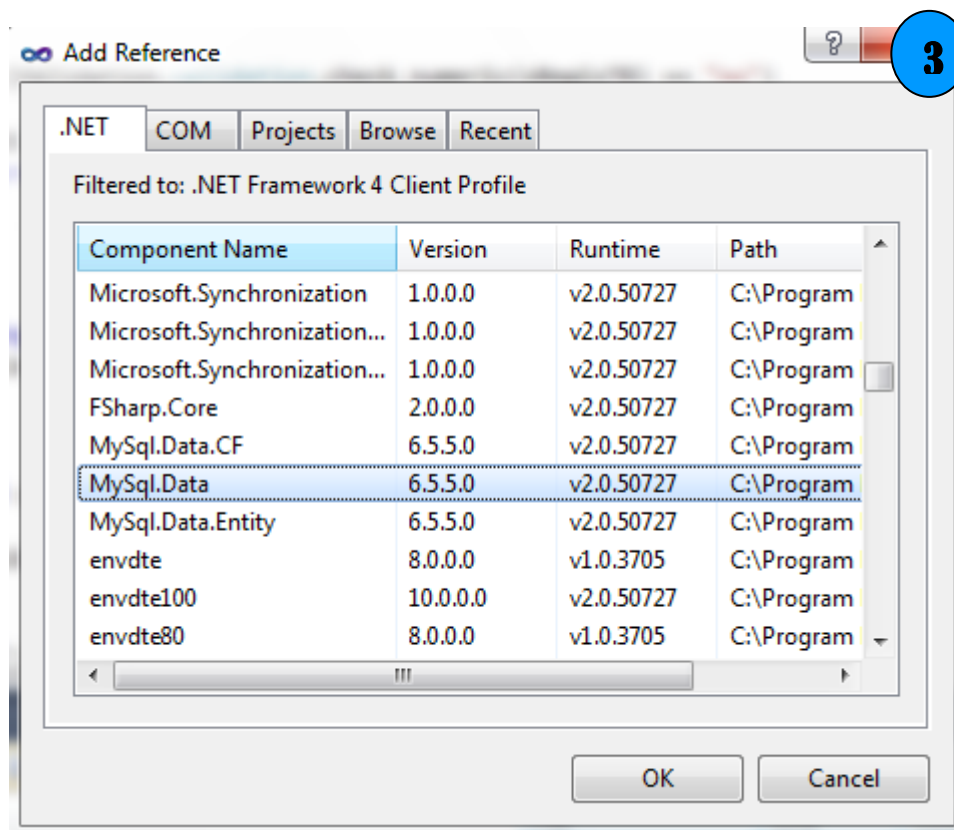3. Click on phpmyadmin to see the Mysql Database part(Fig-1).
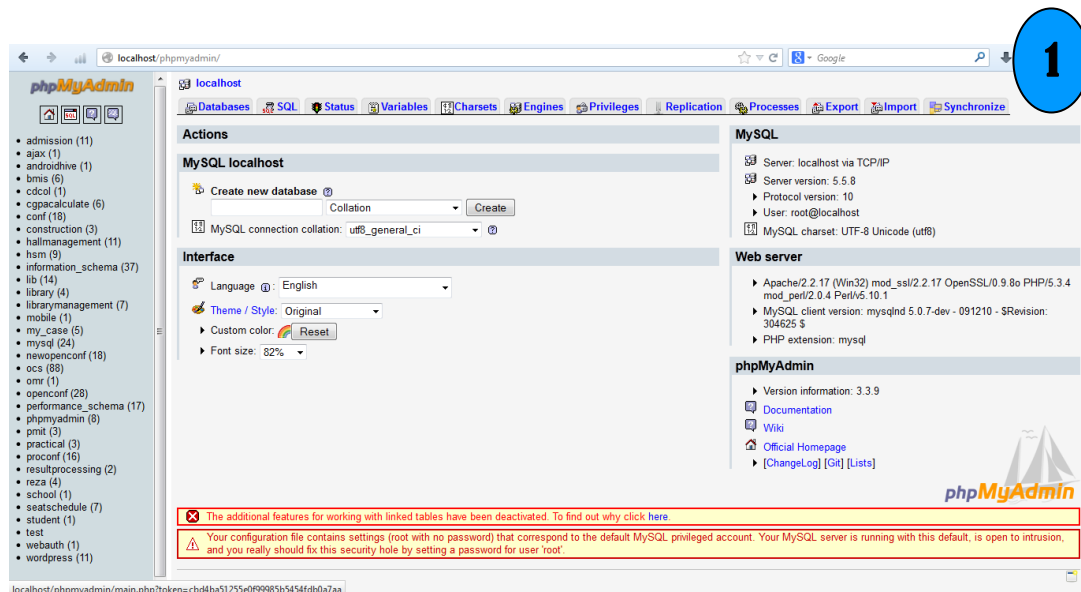

Task 3: Installation of MySQL Connector

Steps:

1. Downloading Connector/Net from
   http://dev.mysql.com/downloads/connector/net/6.1.html
2. Adding Reference and Creating the MySQL Connector DLL from the Project(Fig-2)

3. Right click on References and select > **Add Reference…** and see the dialog below(Fig-3).
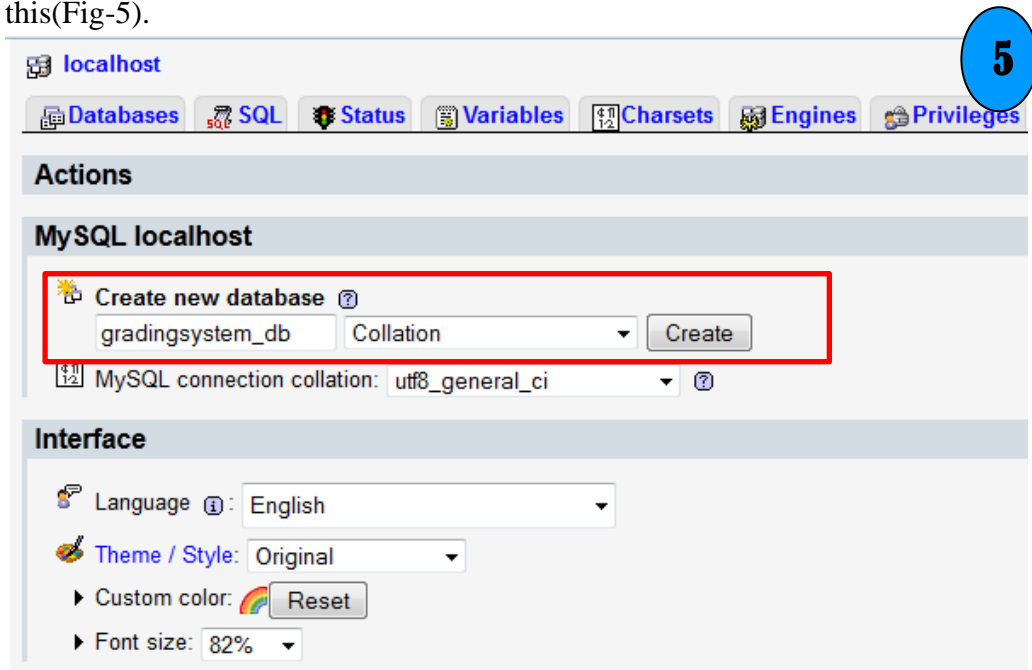
4. From **.NET** Tab, select the **Mysql.Data** component and click ok.
5. Now, see the references to see Mysql.Data (Fig-4).
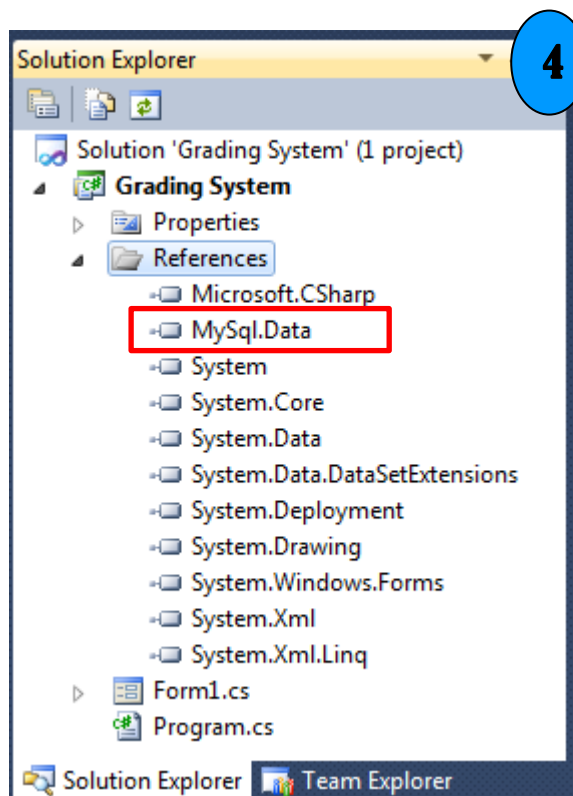
Task 4: Creating Database and It's table based on our project

Steps:
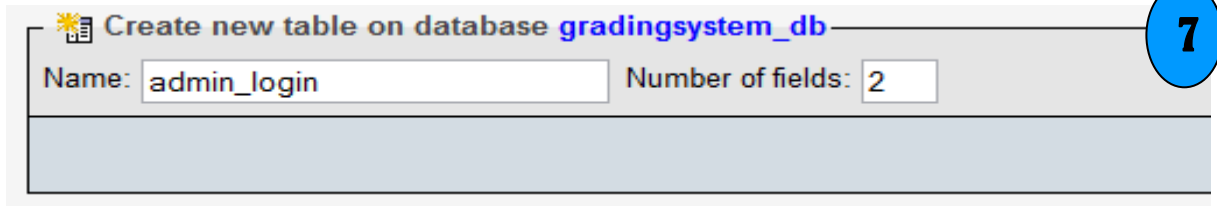1. Creating a project Database on MySQL.
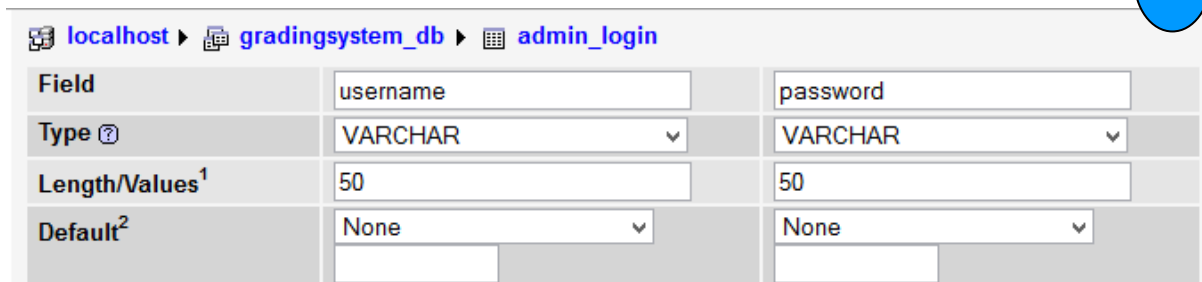2. Go to the Phpmyadmin and create database as name "**gradingsystem_db**" like this(Fig-5).

3. Do the following design which form name is **Form1.cs** (Fig-6).Then We have to create table based on our project style
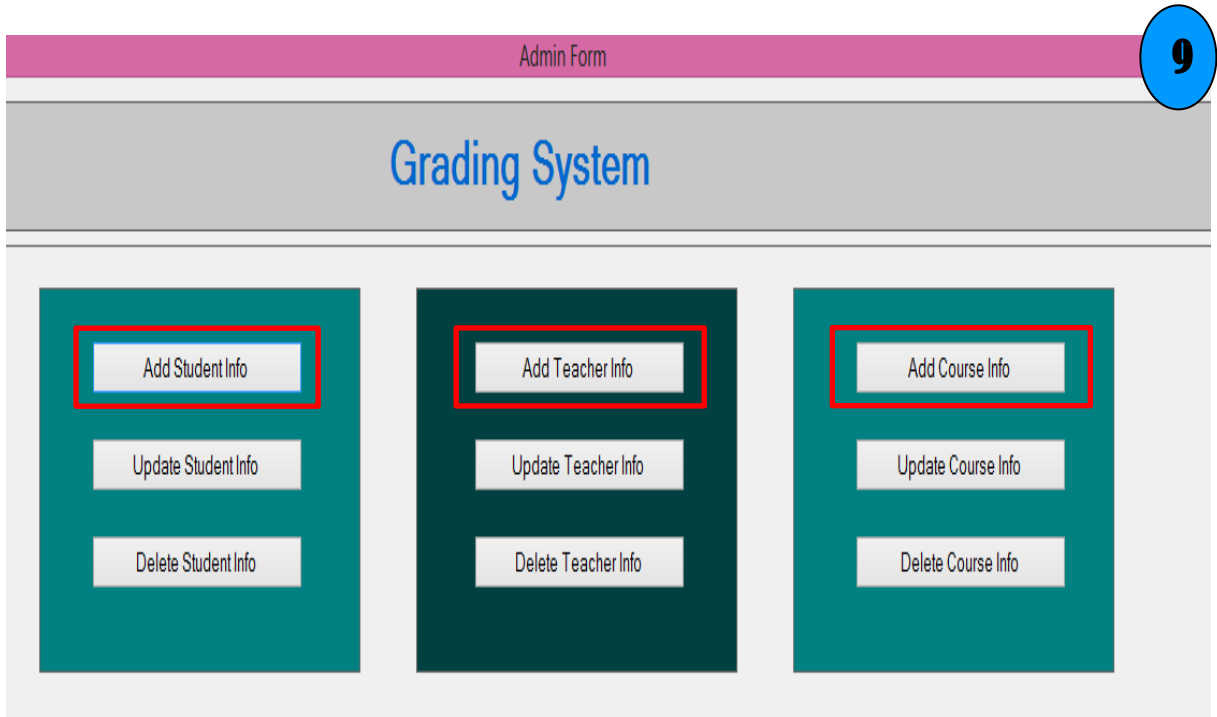   Admin/ Teacher Login: **Form1.cs**(Fig-6).

4. Go to the Phpmyadmin and create table as name "**admin_login**" like this(Fig-7).
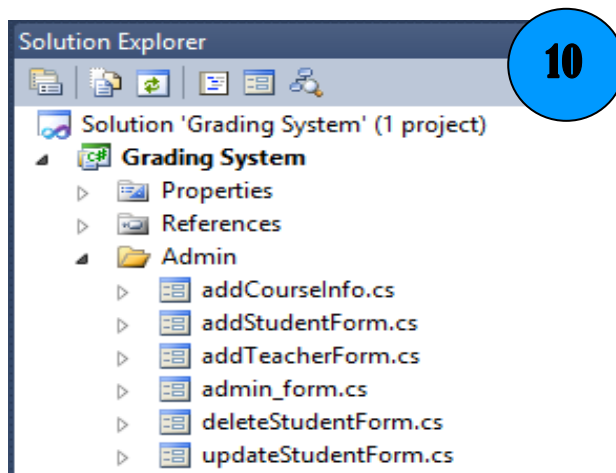

Now we add field name(column name) into our table which is like this(Fig-8)

5. After Admin login he/she see the following option(Fig-9).you design the following UI(Fig-9).This form name is **admin_form.cs** which create in Admin folder(Fig-10).

6. Now we click Add Student Info Button(Fig-9) and see the following design which you will make(Fig-11).This form name is **addStudentForm.cs** which create in Admin folder (Fig-10).

7. Go to the Phpmyadmin and create table as name "**admin_login**" like this(Fig-12).

Now we add field name(column name) into our table which is like this(Fig-13)

8. Now we click Add Teacher Info Button(Fig-9) and see the following design which you will make(Fig-14).This form name is **addTeacherForm.cs** which create in Admin folder (Fig-10).

9. Go to the Phpmyadmin and create table as name "**teacher_info**" like this(Fig-12).

Task 1: Draw the database design for *teacher_info* table.

Note: If you try doesn't work, see *Appendix-A* which is developed at the end of the document.

10. Now we click Add Course Info Button(Fig-9) and see the following design which you will make(Fig-16).This form name is **addCourseInfo.cs** which create in Admin folder (Fig-10).

11. Go to the Phpmyadmin and create table as name "**teacher_info**" like this(Fig-17).

Task 2: Draw the database design for *course_info* table.

Note: If you try doesn't work, see ***Appendix-A*** which is developed at the end of the document.

Task 5: Connection with Database with our project

For connect to database we create a class file name is **dbConnect.cs** which create in Database folder(Fig-18)
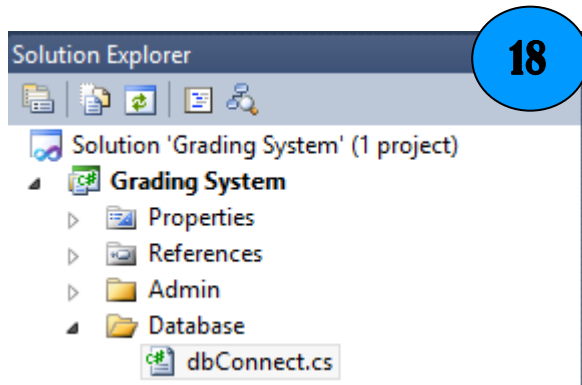

```csharp
class dbConnect
  {

        MySqlConnection connection = Initialize();
        //For initialize
        public static MySqlConnection Initialize()
          {
        MySqlConnection connection;
        string connectionString;

        string server = "localhost";
        string database = "gradingsystem_db";
        string uid = "root";
        string password = "";

    connectionString = "SERVER=" + server + ";" + "DATABASE=" + database + ";" +
"UID=" + uid + ";" + "PASSWORD=" + password + ";";
        connection = new MySqlConnection(connectionString);
        return connection;


          }
  }
```
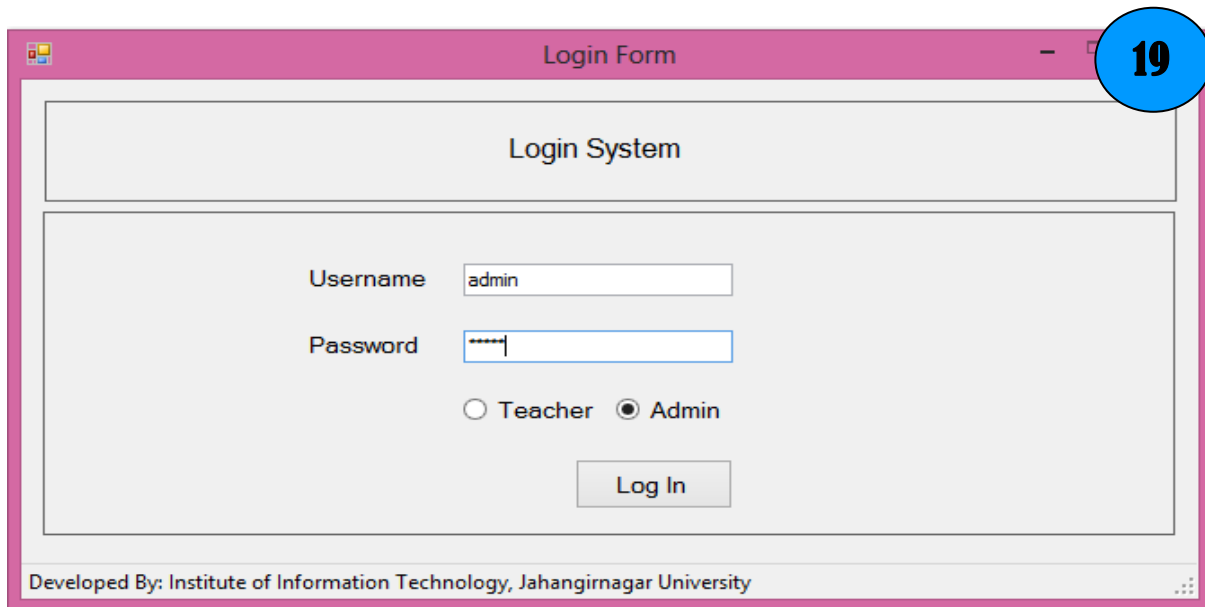
Task 6: Working with database with Admin/Teacher login form in our project

Steps:
1. Now we login as Admin(Fig-19) into the Admin Home Page(Fig-9)

2. Double click the Log In Button and write the following code:

```
MySqlConnection con = Database.dbConnect.Initialize();

if (usernameTB.Text == "" || passwordTB.Text == "")
{
    MessageBox.Show("Username or password is empty. try again.", "Error");
    return;
}

con.Open();
string query = "";
if (adminRB.Checked)
query = "SELECT `username`, `password` FROM `admin_login` WHERE
`username`='" + usernameTB.Text + "' and `password` = '" + passwordTB.Text + "'";

    if (teacherRB.Checked)
    query = "SELECT `t_user_name`, `t_pass_word` FROM `teacher_info` WHERE
`t_user_name`='" + usernameTB.Text + "' and `t_pass_word` ='" + passwordTB.Text + "'";

MySqlCommand cmd = new MySqlCommand(query, con);
MySqlDataReader reader = cmd.ExecuteReader();
```

```csharp
if (reader.Read())
 {
    if (adminRB.Checked)
    {
       Admin.admin_form aa = new Admin.admin_form();
       aa.Visible = true;
    }
    if (teacherRB.Checked)
    {

       Admin.teacherHomePage tf = new Admin.teacherHomePage();
       tf.Visible = true;
    }

 }

 else
 {
    MessageBox.Show("Username or password is incorrect. try again.", "Error");
 }

 reader.Close();
 con.Close();
```
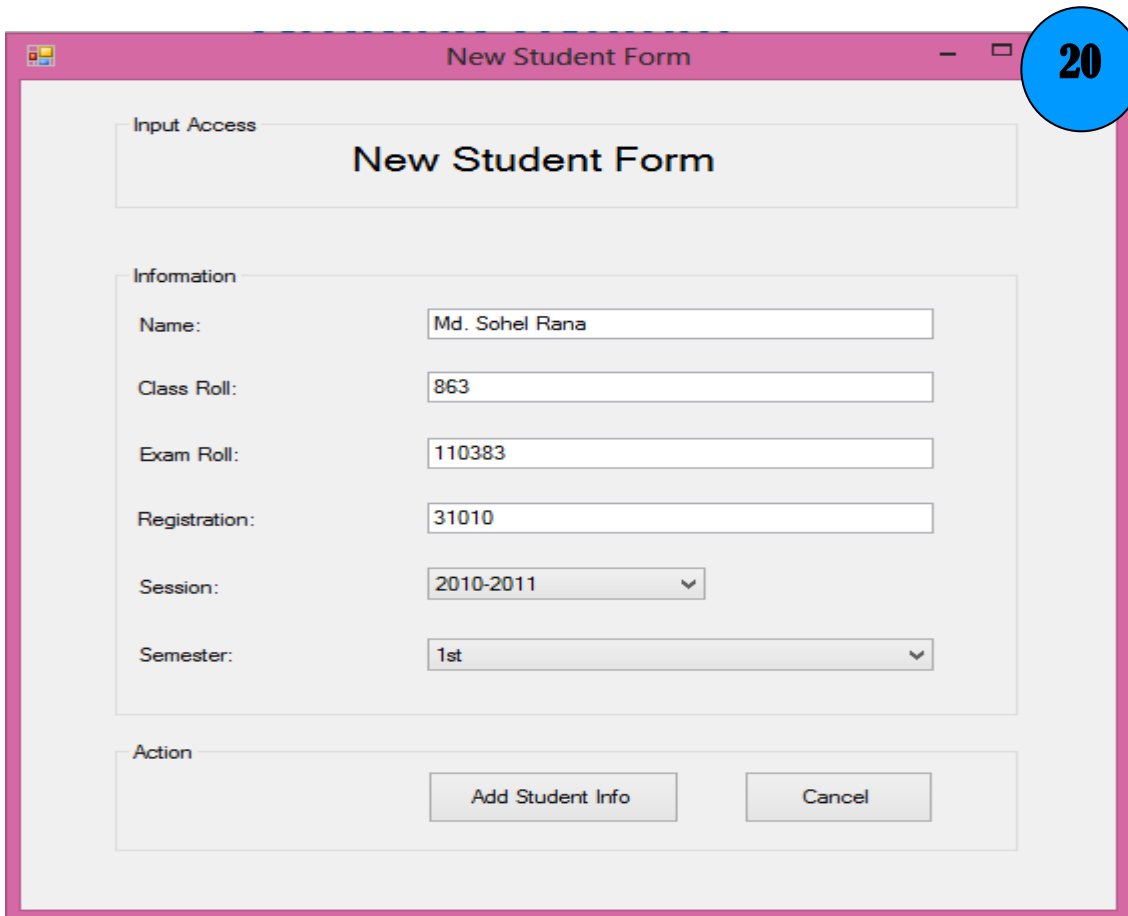
Task 7: Add,Delete,Update Student Information to database in our project

Practice-1: Add Student Information into **student_info** table

Steps:

1. After login as Admin we will see the Admin Home Page and see a option Add Student Info Button(Fig-9).If we click Add Student Info Button we will see the following picture(Fig-20)

2. Now we double click Add Student Info Button and write the following code

```
MySqlConnection con = Database.dbConnect.Initialize();

if (sNameTB.Text == "" || sClassRollTB.Text == "" || sExamRollTB.Text == "" ||
sRegisTB.Text == "")
    {

    MessageBox.Show("empty field is not allow. try again.", "Empty");
```

```
            return;
        }

        con.Open();

        string sql = "insert into
`student_info`(`s_name`,`s_c_roll`,`s_e_roll`,`s_reg_number`,`session`,`semester`) values('"
+ sNameTB.Text + "','" + sClassRollTB.Text + "','" + sExamRollTB.Text + "','" +
sRegisTB.Text + "','" + sSessionCB.SelectedItem+ "','" + sSemesterCB.SelectedItem+"');";

    try
        {
            DialogResult d = MessageBox.Show("Are you sure to do this action?",
"Confirm", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (d == DialogResult.Yes)
            {
                MySqlCommand cmd = new MySqlCommand(sql, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Successfully data inserted.", "Success");
            }
            con.Close();

        }
        catch (Exception ex)
        {
            MessageBox.Show("Error Message:" + ex.Message);
            con.Close();

        }
```

Practice-2: Delete Student Information from **student_info** table(Do it). Make design the following picture(Fig-21)

Input Annex

Enter Exam Roll _____ [Search]

Information Annex

Name _____

Class Roll _____

Exam Roll _____

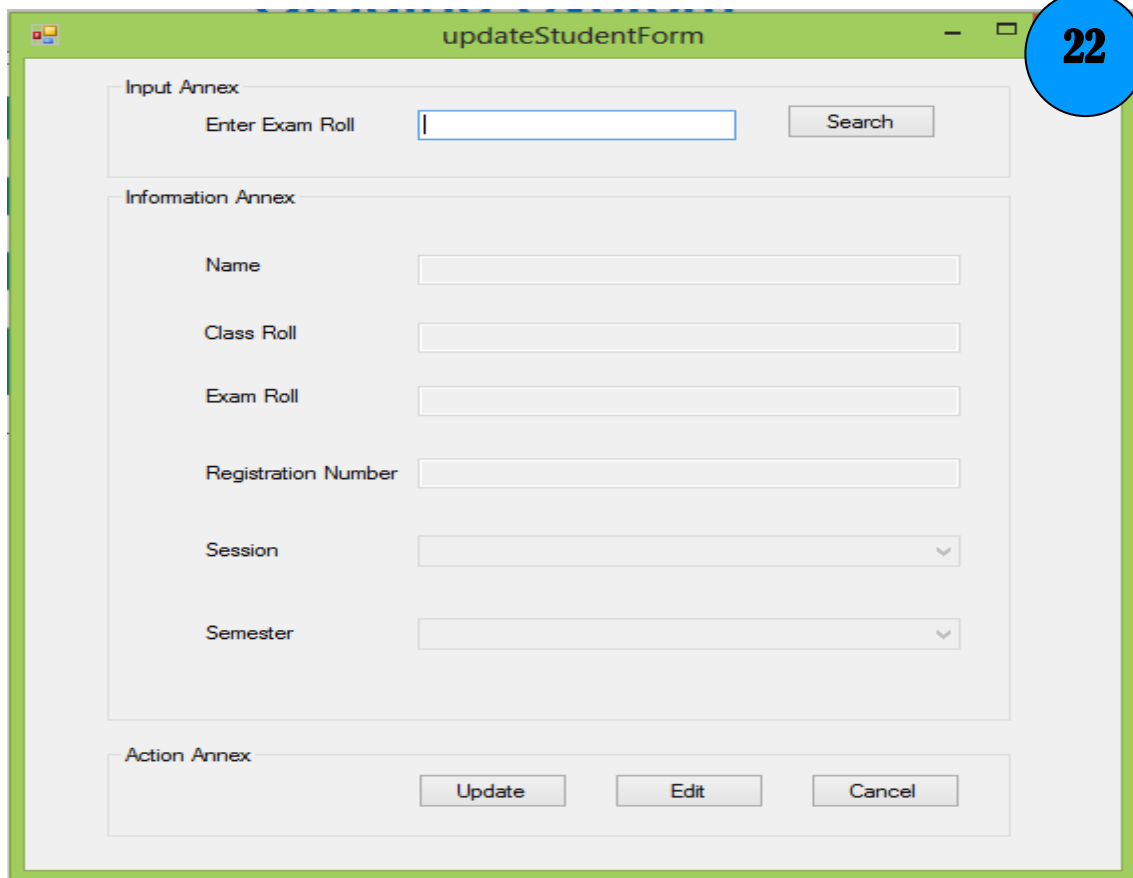Registration Number _____

Session _____ ▾

Semester _____ ▾

Action Annex

[Delete]   [Cancel]

21

Practice-3: Edit Student Information from **student_info** table(Do it). Make design the following picture(Fig-22)

Task 8: Add,Delete,Update Teacher Information to database in our project

Practice-1: Add Teacher Information to **teacher_info** table(Do it).Make design the following picture(Fig-23)

Practice-2: Delete Teacher Information from **teacher_info** table (Do it)

Practice-3: Edit Teacher Information from **teacher_info** table (Do it)

Task 9: Add,Delete,Update Course Information to database in our project

Practice-1: Add Course Information to **course_info** table(Do it).Make design the following picture(Fig-24)

Practice-2: Delete Course Information from **course_info** table (Do it)

Practice-3: Edit Course Information from **course_info** table (Do it)

Appendix-A

Help 1: Database Design for *teacher_info.*

Help 2: Database Design for *course_info.*