

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

- Apache
- MySQL database
- PHP
- Filezilla etc.

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

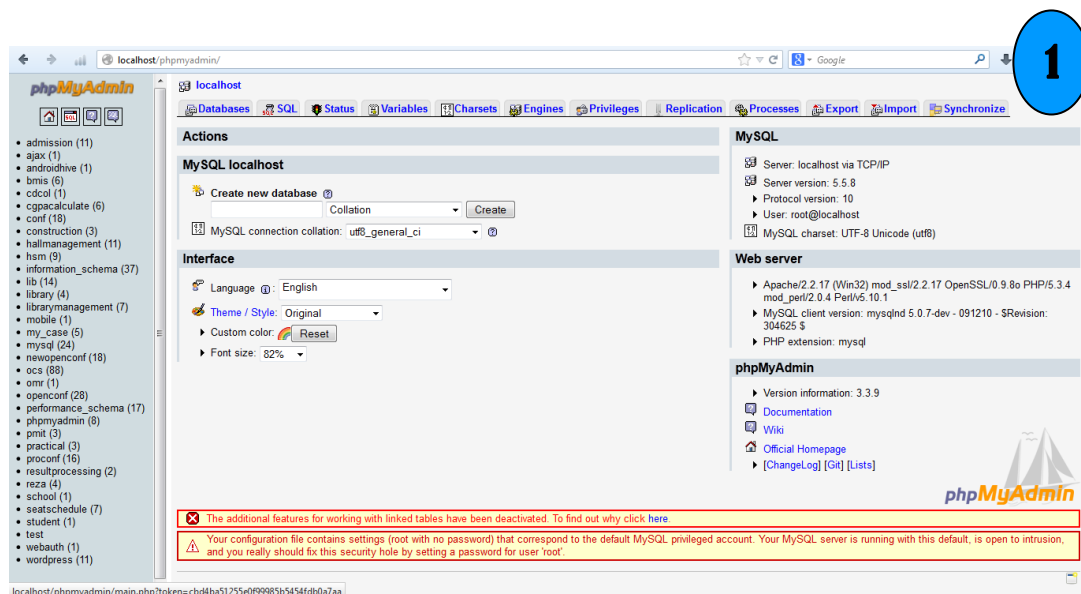
[Apache HTTPD](#), [MySQL](#), [PHP](#), [Perl](#), [FileZilla FTP Server](#), [phpMyAdmin](#), [OpenSSL](#), [Freetype](#), [Webalizer](#), [mod_perl](#), [eAccelerator](#), [mcrypt](#), [SQLite](#), [Mercury Mail Transport System](#), [fake sendmail for windows](#), [FPDF Class](#)

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.

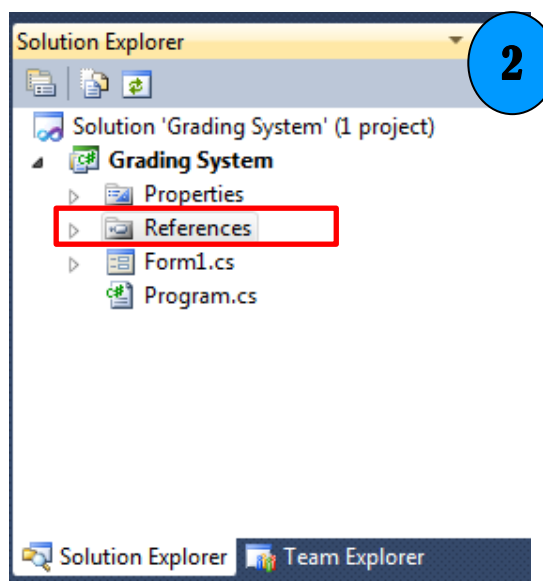
3. Click on phpmyadmin to see the Mysql Database part.



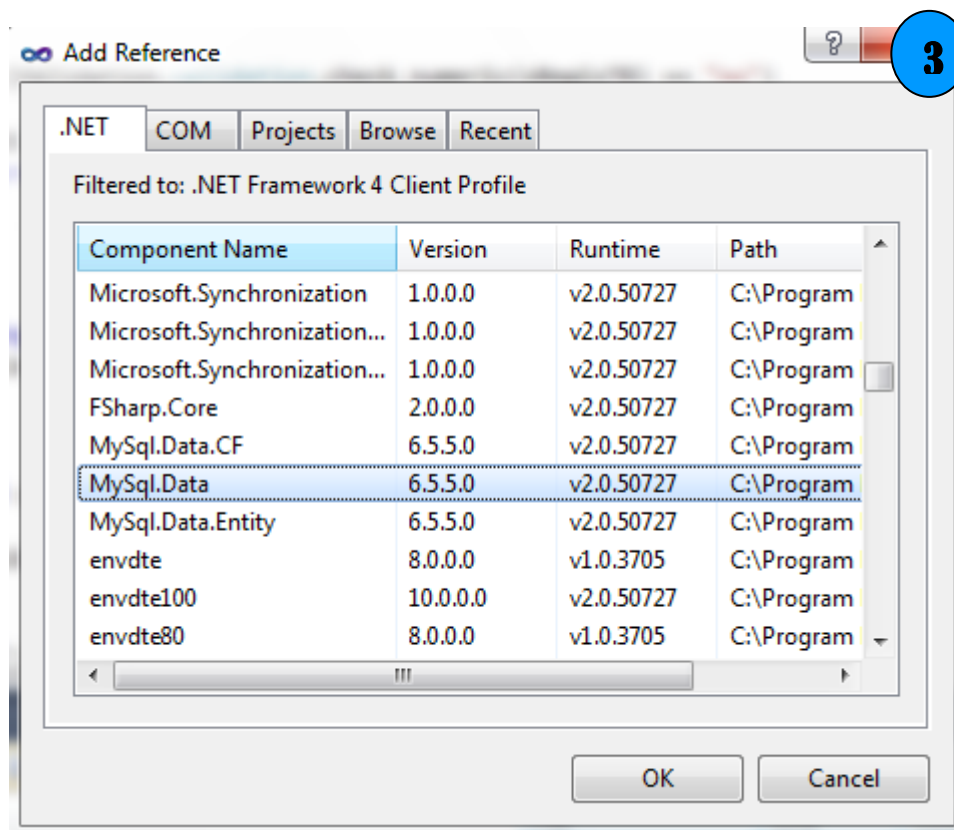
Task 3: Installation of MySQL Connector

Steps:

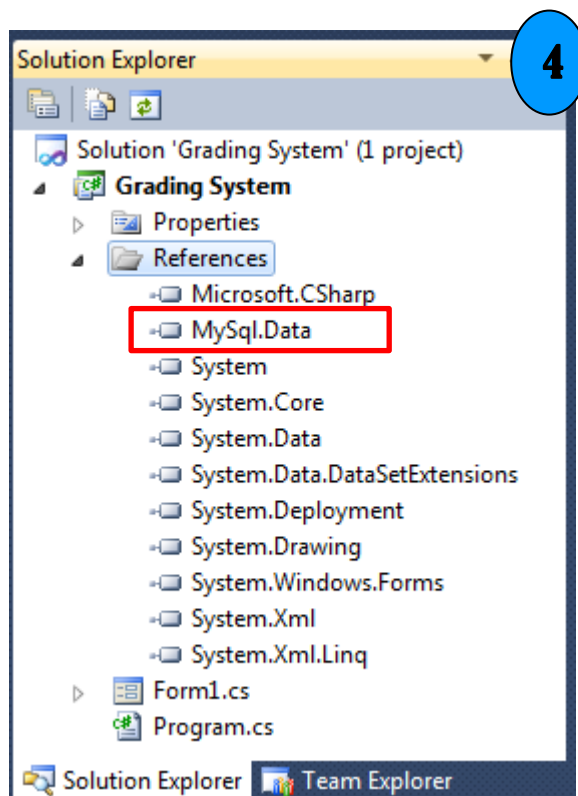
1. Downloading Connector/Net from <http://dev.mysql.com/downloads/connector/net/6.1.html>
2. Adding Reference and Creating the MySQL Connector DLL from the Project



3. Right click on References and select > **Add Reference...** and see the dialog below.



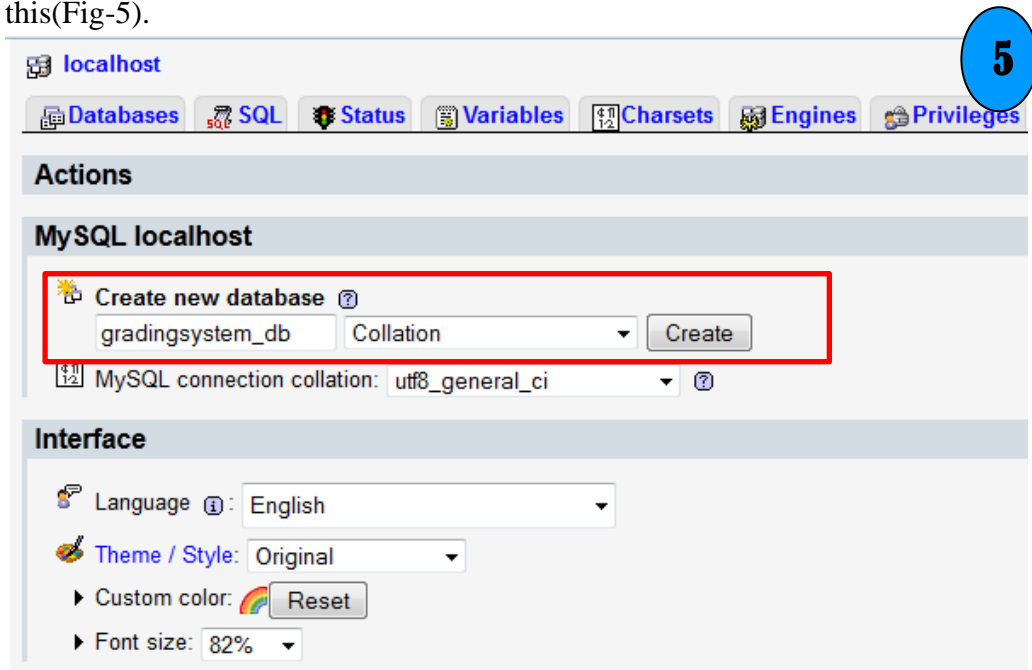
4. From **.NET** Tab, select the **MySQL.Data** component and click ok.
5. Now, see the references to see MySQL.Data References add or not.



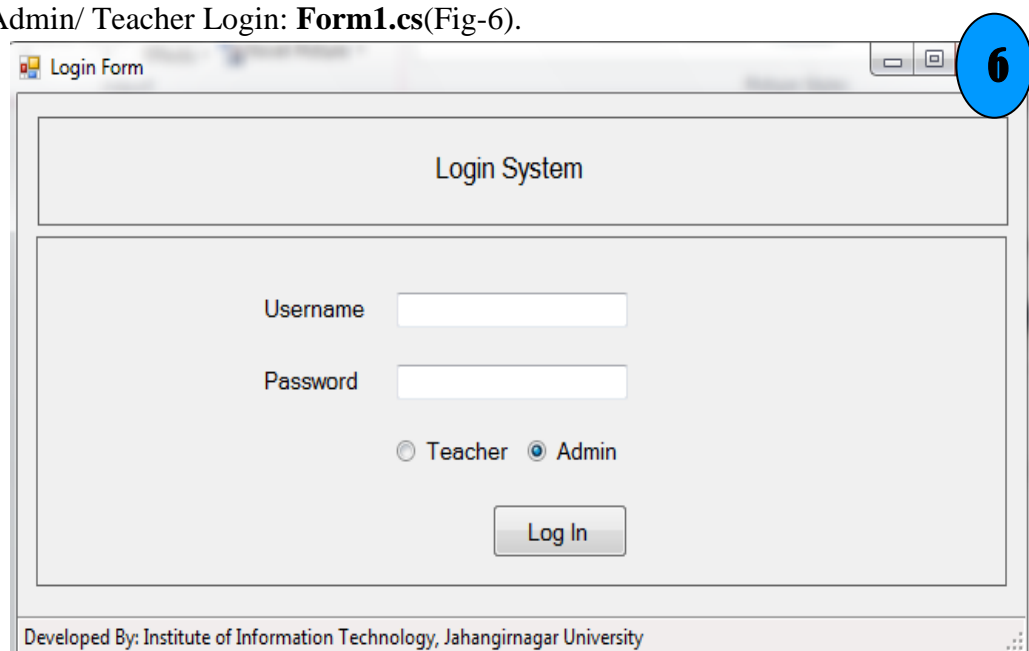
Task 4: Creating Database and It's table based on our project

Steps:

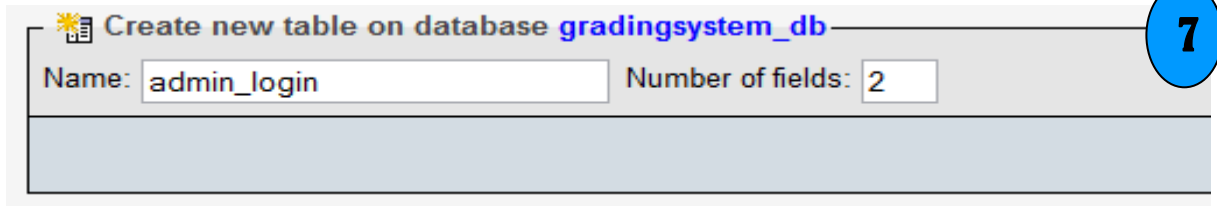
1. Creating a project Database on MySQL.
2. Go to the Phpmyadmin and create database as name “**gradingsystem_db**” like this(Fig-5).



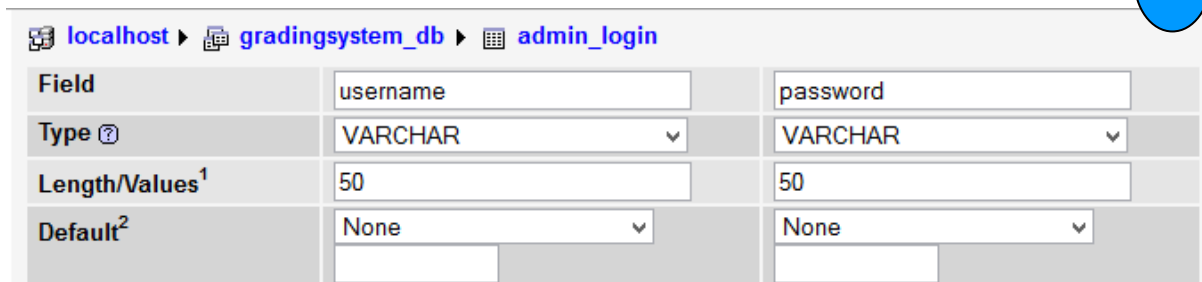
3. Do the following design which form name is **Form1.cs** (Fig-6).Then We have to create table based on our project style
Admin/ Teacher Login: **Form1.cs**(Fig-6).



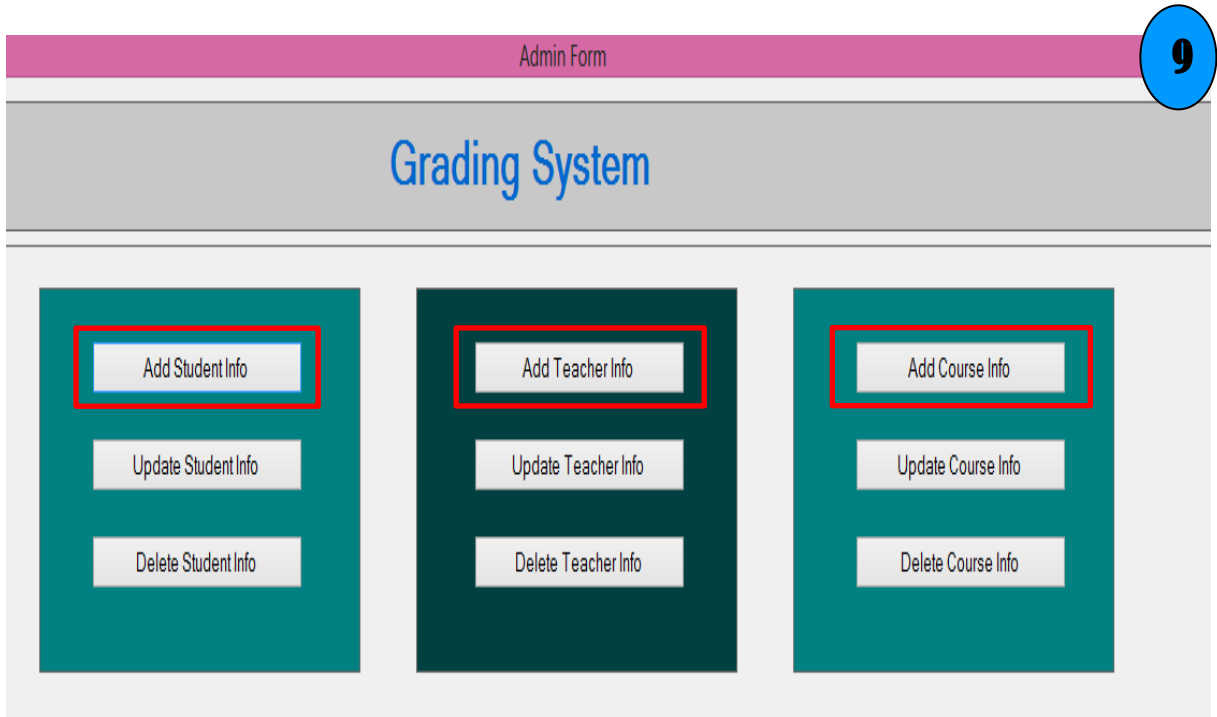
4. Go to the Phpmyadmin and create table as name “**admin_login**” like this(Fig-7).

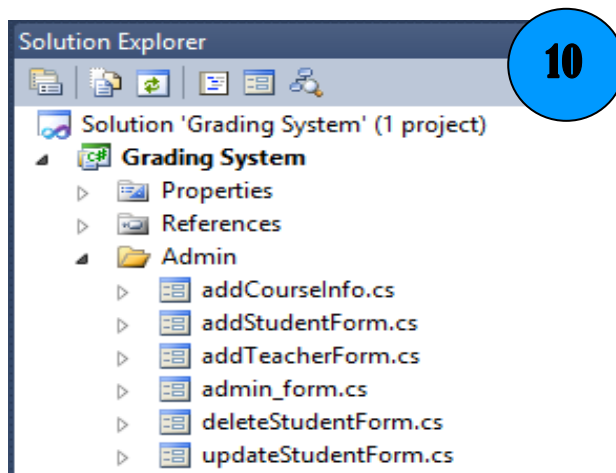


Now we add field name(column name) into our table which is like this(Fig-8)



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).

New Student Form

Input Access

New Student Form

Information

Name:

Class Roll:

Exam Roll:

Registration:

Session:

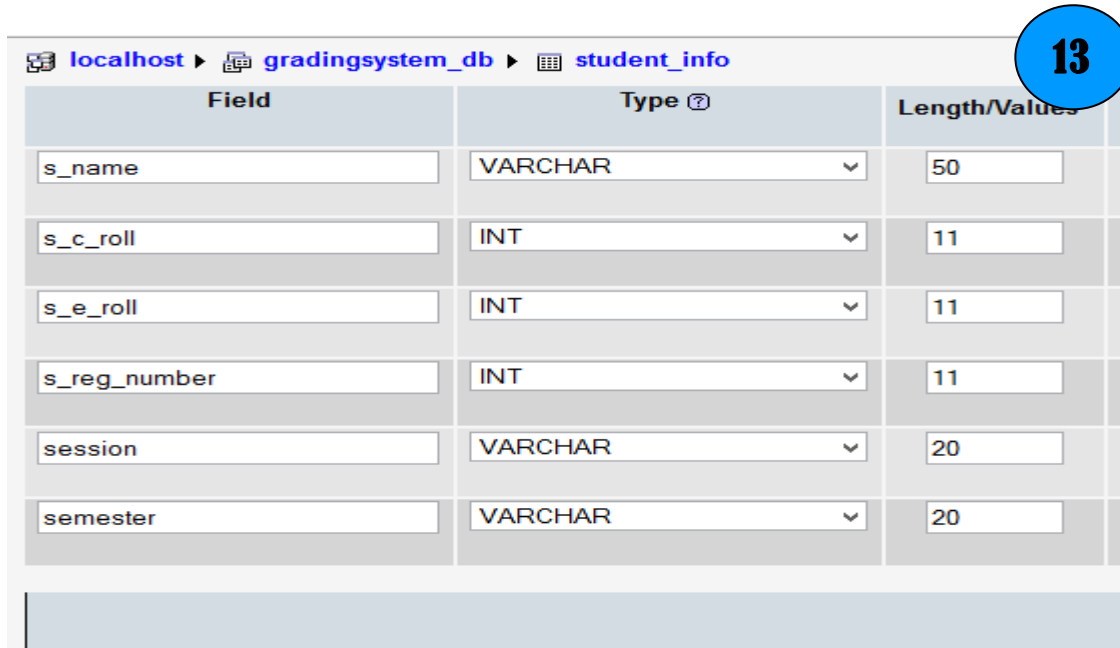
Semester:

Action

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)



Field	Type ?	Length/Values
s_name	VARCHAR	50
s_c_roll	INT	11
s_e_roll	INT	11
s_reg_number	INT	11
session	VARCHAR	20
semester	VARCHAR	20

- 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).

New Teacher Form

Input Access

New Teacher Form

Information

Name

Short Code

Designation

Professor

Department

User Name

Password

Action

Add Teacher Info

Cancel

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

Create new table on database gradingsystem_db

Name: teacher_info

Number of fields: 6

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).

addCourseInfo

16

Input Access

Add Course Info

Information

Course Title

Course Code

Credit

3

Session

2009-2010

Semester

1st

Action

Add Course Info

Cancel

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

Create new table on database gradingsystem_db

17

Name: course_info

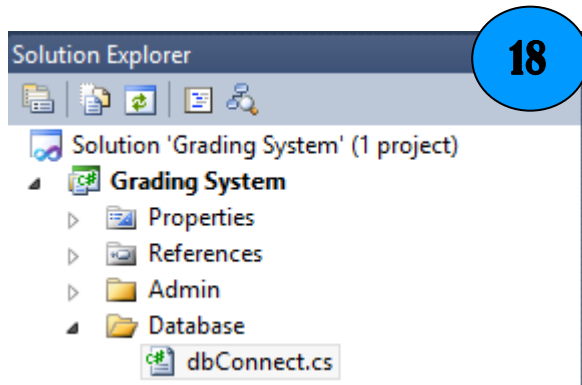
Number of fields: 5

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)



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

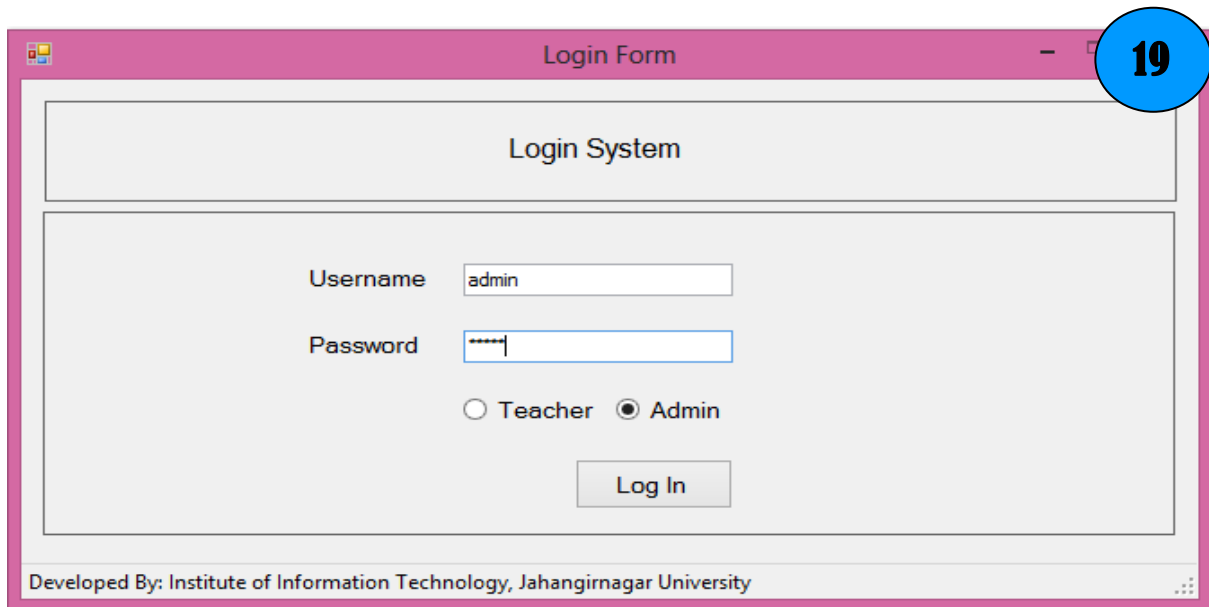
        string server = "localhost";
        string database = "gradingsystem_db";
        string uid = "root";
        string password = "";

        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();
```

```
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");
}

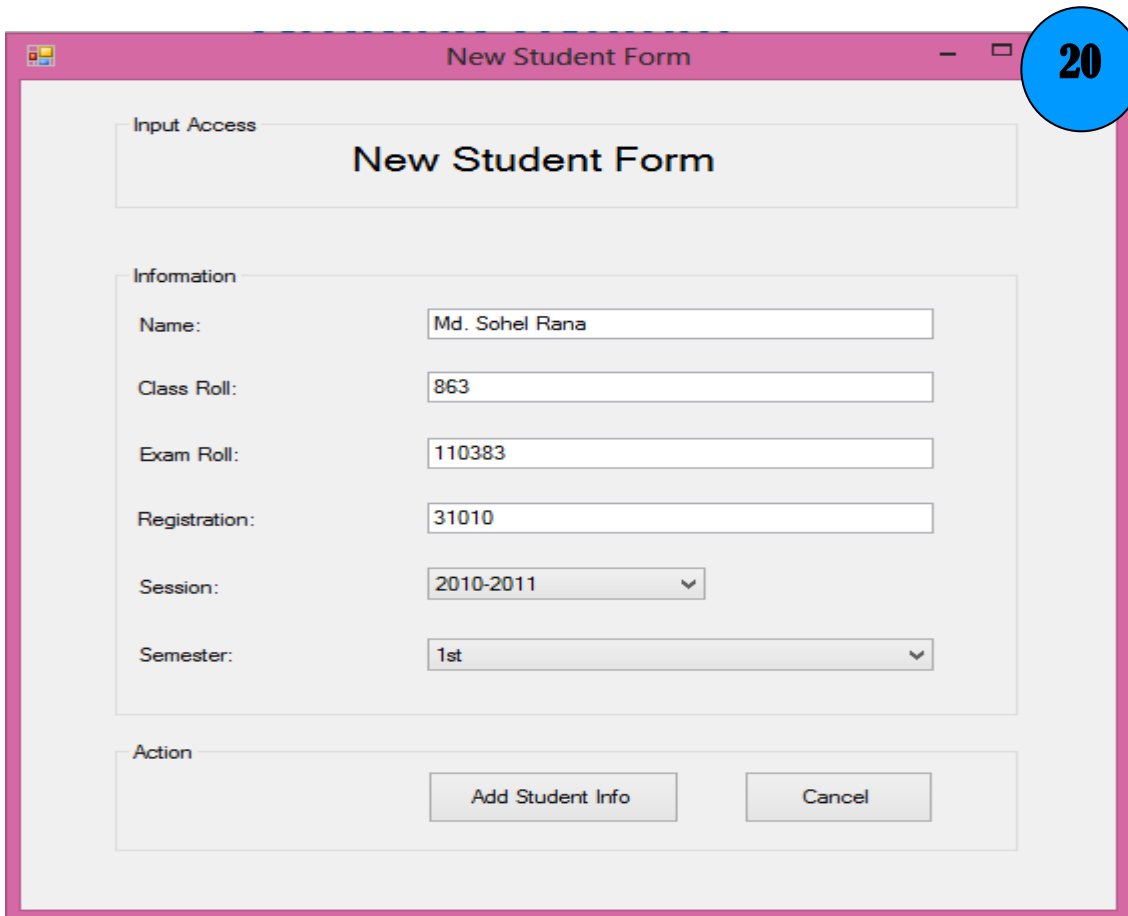
reader.Close();
con.Close();
```

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)

21

Input Annex

Enter Exam Roll

Information Annex

Name

Class Roll

Exam Roll

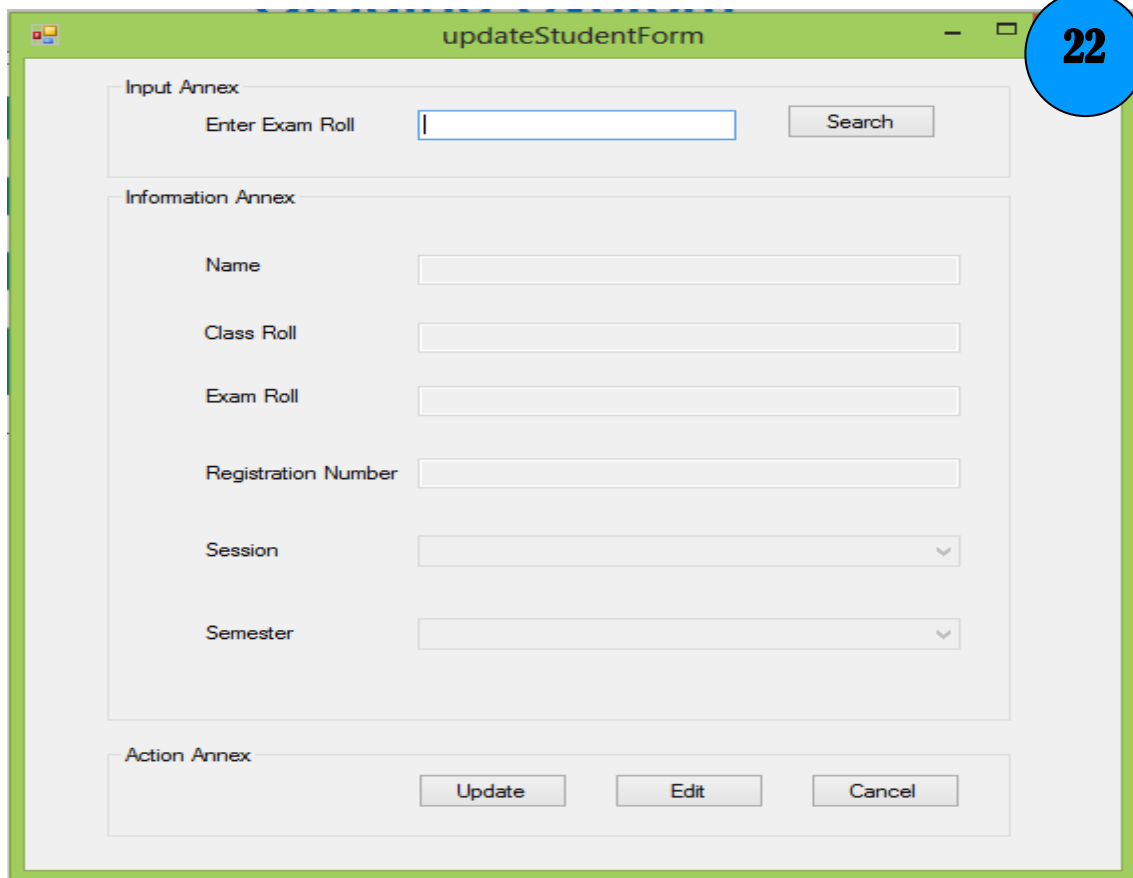
Registration Number

Session

Semester

Action Annex

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



The screenshot shows a Java Swing window titled "updateStudentForm". It is divided into three main sections:

- Input Annex:** Contains a label "Enter Exam Roll" followed by a text input field and a "Search" button.
- Information Annex:** Contains a vertical list of labels and input fields: "Name", "Class Roll", "Exam Roll", "Registration Number", "Session" (with a dropdown arrow), and "Semester" (with a dropdown arrow).
- Action Annex:** Contains three buttons: "Update", "Edit", and "Cancel".

A blue circle with the number "22" is overlaid on the top right corner of the window.

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)

New Teacher Form

Input Access

New Teacher Form

Information

Name: Md. Shamim Al Mamun

Short Code: SAM

Designation: Assistant Professor

Department: IIT

User Name: Shamim

Password: [masked]

Action

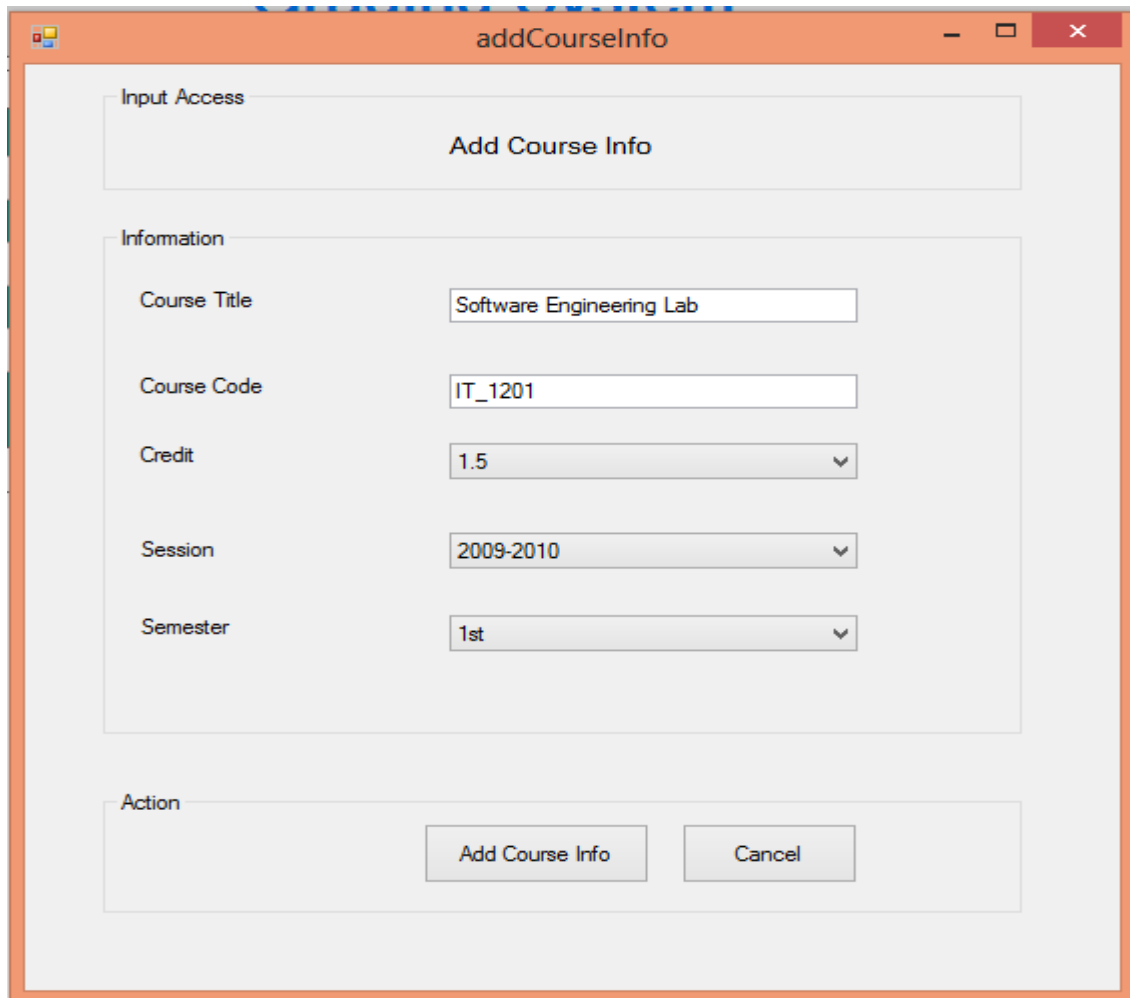
Add Teacher Info Cancel

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)



The screenshot shows a Java Swing window titled "addCourseInfo". The window has a light gray background and an orange border. It is divided into three main sections: "Input Access", "Information", and "Action".

- Input Access:** A single text field containing the text "Add Course Info".
- Information:** A section containing five labeled text fields and three dropdown menus:
 - Course Title:** "Software Engineering Lab"
 - Course Code:** "IT_1201"
 - Credit:** "1.5" (with a dropdown arrow)
 - Session:** "2009-2010" (with a dropdown arrow)
 - Semester:** "1st" (with a dropdown arrow)
- Action:** A section containing two buttons: "Add Course Info" and "Cancel".

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*.

localhost ► gradingsystem_db ► teacher_info

Field	Type ?	Length/Values ¹
t_name	VARCHAR ▼	50
t_s_code	VARCHAR ▼	20
t_desig	VARCHAR ▼	50
t_dept	VARCHAR ▼	50
t_user_name	VARCHAR ▼	50
t_pass_word	VARCHAR ▼	50

Help 2: Database Design for *course_info*.

localhost ► gradingsystem_db ► course_info

Field	Type ?	Length/Values ¹
c_title	VARCHAR ▼	50
c_code	VARCHAR ▼	50
credit	VARCHAR ▼	50
session	VARCHAR ▼	50
semester	VARCHAR ▼	50