**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](http://httpd.apache.org), [MySQL](http://www.mysql.org), [PHP](http://www.php.net), [Perl](http://www.cpan.org/), [FileZilla FTP Server](http://filezilla.sourceforge.net/), [phpMyAdmin](http://www.phpmyadmin.net/), [OpenSSL](http://www.openssl.org), [Freetype](http://www.freetype.org), [Webalizer](http://www.mrunix.net/webalizer/), [mod\_perl](http://perl.apache.org), [eAccelerator](http://eaccelerator.net/), [mcrypt](http://mcrypt.sourceforge.net/), [SQLite](http://www.sqlite.org/), [Mercury Mail Transport System](http://www.pmail.com/overviews/ovw_mercury.htm), [fake sendmail for windows](http://glob.com.au/sendmail/), [FPDF Class](http://www.fpdf.org/)

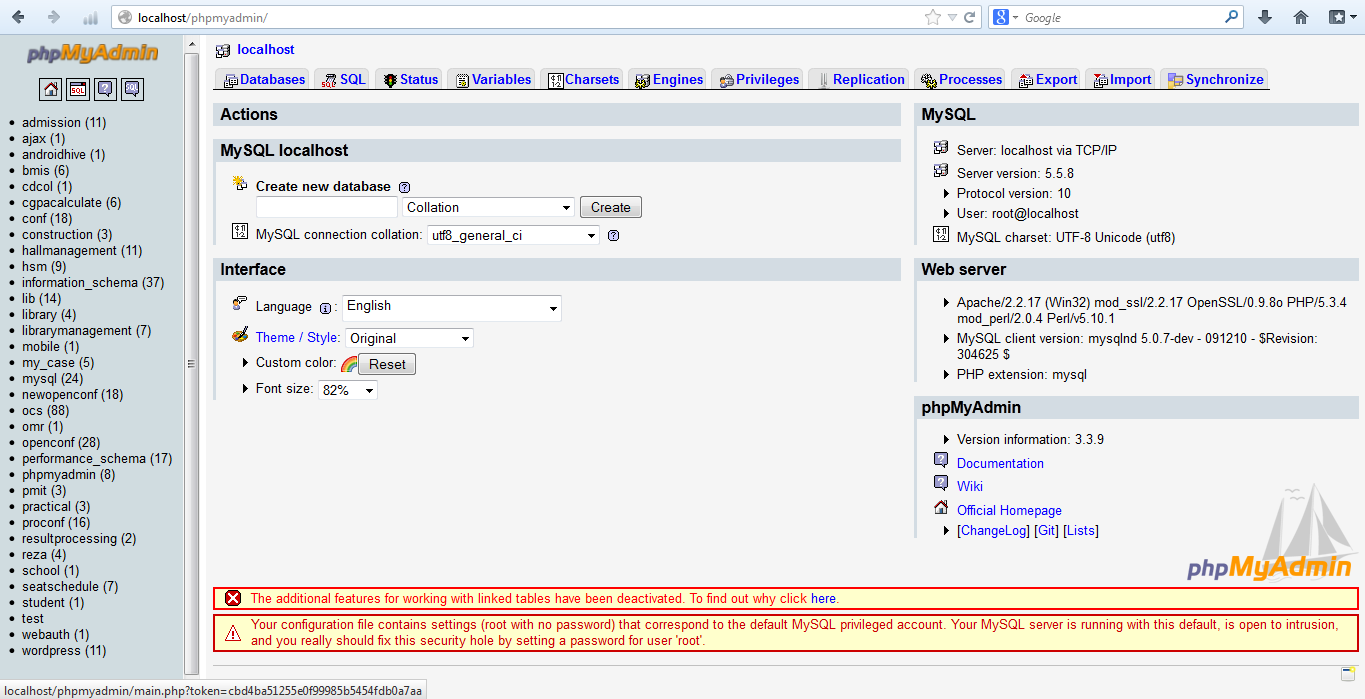
|  |  |
| --- | --- |
|  |  |

**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.

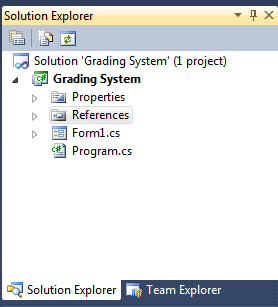
**1**



**Task 3: Installation of MySQL Connector**

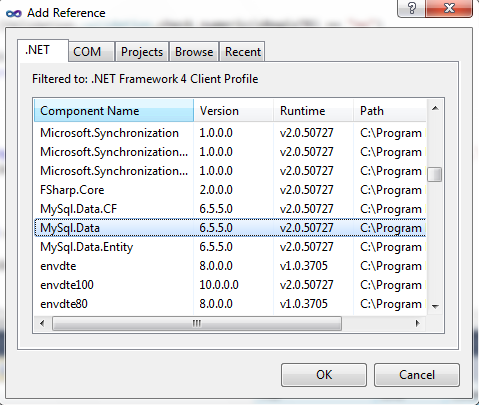
**Steps:**

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



**2**

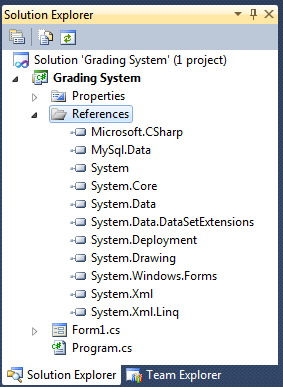
1. Right click on References and select > **Add Reference…** and see the dialog below.



**3**

1. From **.NET** Tab, select the **Mysql.Data** component and click ok.
2. Now, see the references to see Mysql.Data Refences add or not.

**4**

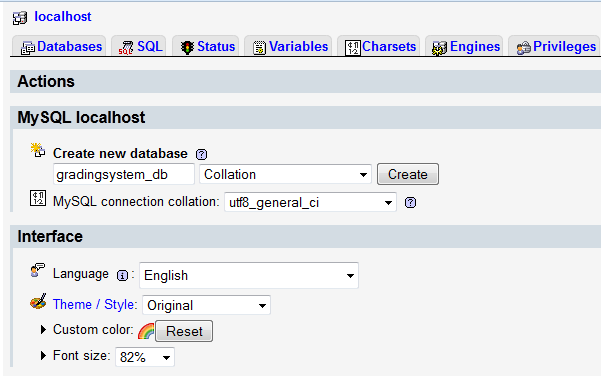


**Task 4: Creating Database and It’s table based on our project**

**Steps:**

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

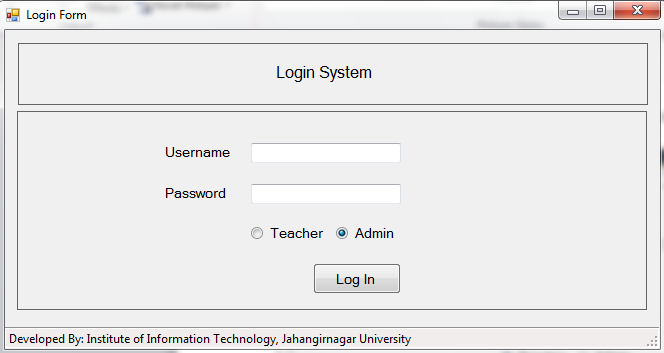
**5**



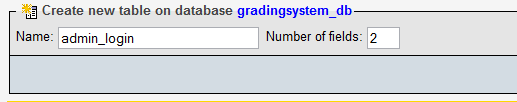
1. Do the following design which form name is **Form1.cs** (Fig-6).Then We have to create table based on our project style

Admin/ Teacher Login: **Form1.cs**(Fig-6).

**6**



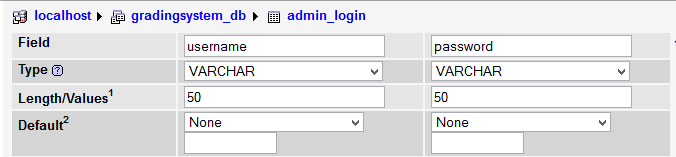
4. Go to the Phpmyadmin and create table as name “**admin\_login**” like this(Fig-7).



**7**

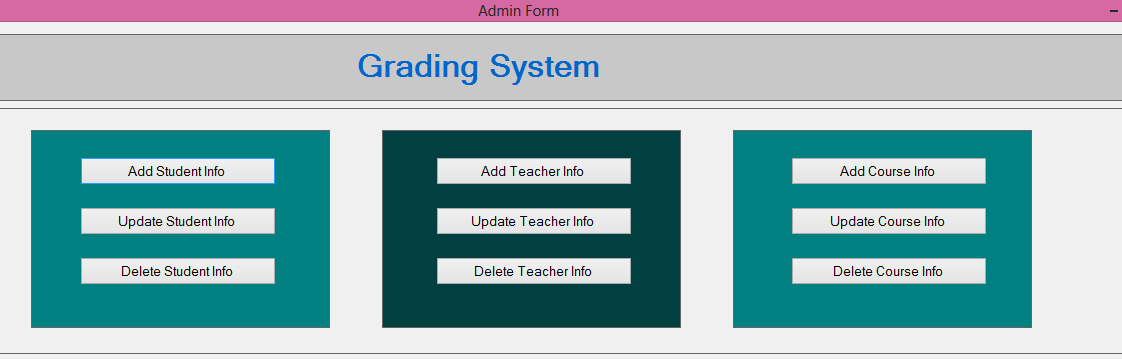
Now we add field name(column name) into our table which is like this(Fig-8)

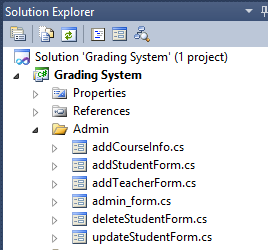
**8**



1. 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).

**9**

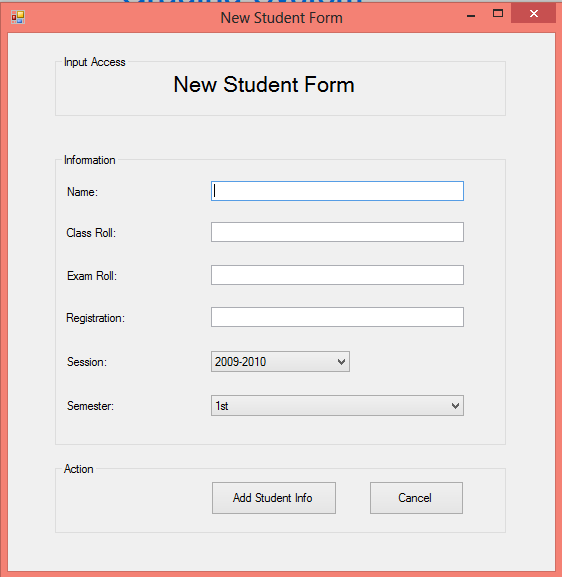




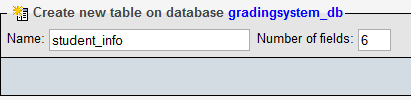
**10**

1. 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).

**11**



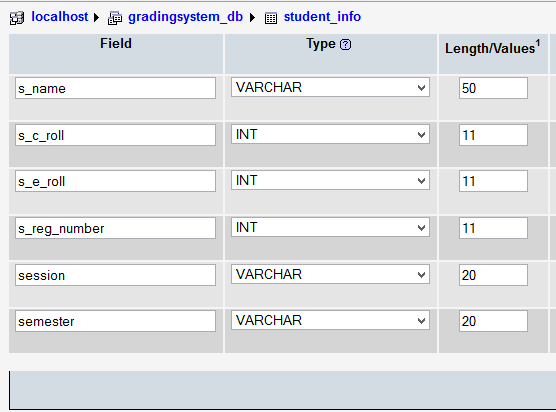
1. Go to the Phpmyadmin and create table as name “**admin\_login**” like this(Fig-12).



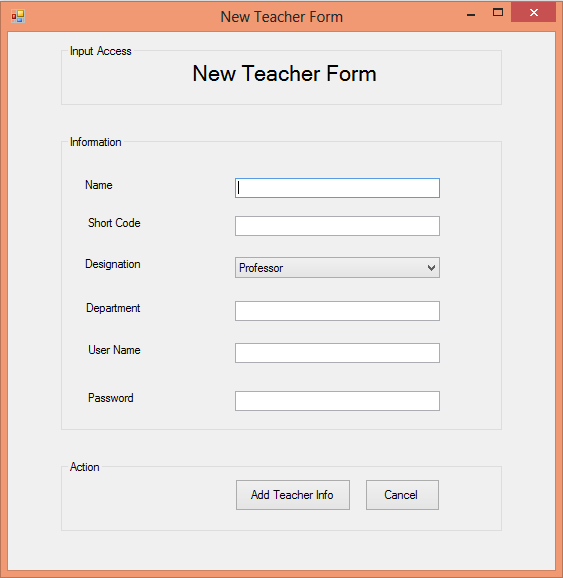
**12**

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

**13**



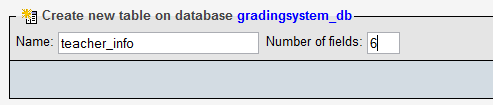
1. Now we click Add Teacher Info Button(Fig-9) and see the following design which you will make(Fig-14).This form name is **addTeacherForm.cs** which create in Admin folder (Fig-10).



**14**

1. Go to the Phpmyadmin and create table as name “**teacher\_info**” like this(Fig-12).

**15**

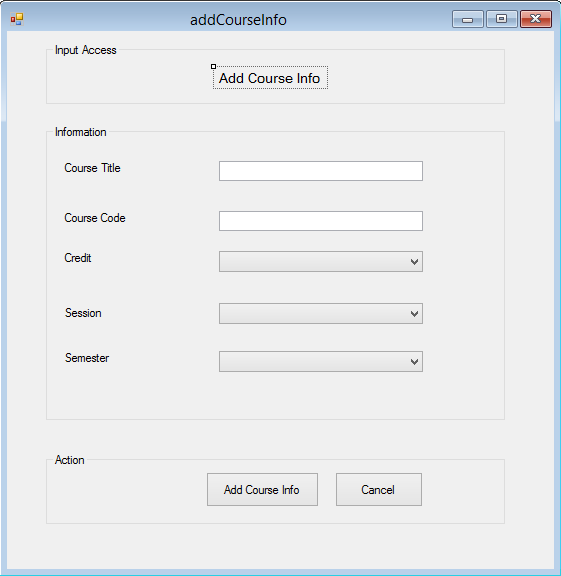


**Task 1:** Draw the database design for ***teacher\_info*** table.

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

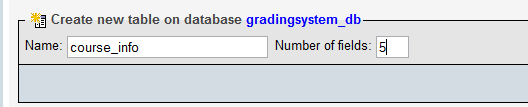
1. Now we click Add Course Info Button(Fig-9) and see the following design which you will make(Fig-16).This form name is **addCourseInfo.cs** which create in Admin folder (Fig-10).

**16**



1. Go to the Phpmyadmin and create table as name “**teacher\_info**” like this(Fig-17).

**17**



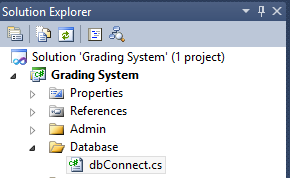
**Task 2:** Draw the database design for ***course\_info*** table.

**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)

**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 + ";";

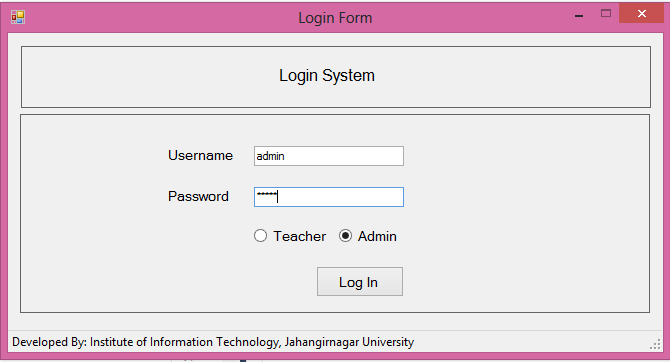
connection = new MySqlConnection(connectionString);

return connection;

}

}

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



MySqlConnection con = Database.dbConnect.Initialize();

if (usernameTB.Text == "" || passwordTB.Text == "")

{

MessageBox.Show("Username or password is empty. try again.", ".....");

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");

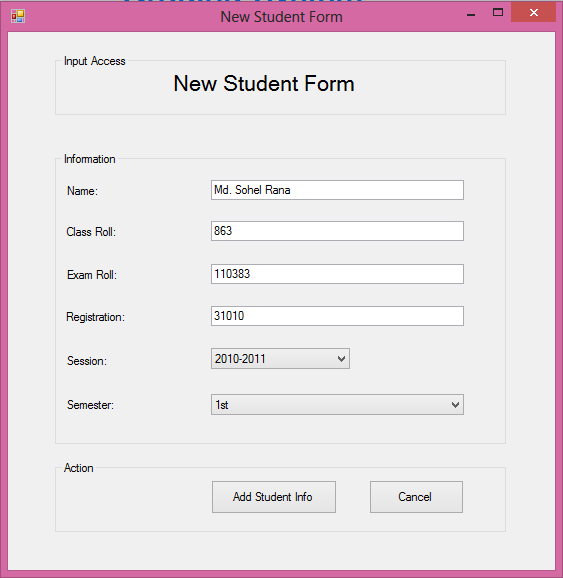
}

reader.Close();

con.Close();

**Task 5: Add,Delete,Update Student Information into database in our project**

**Practice-1: Add Student Information into student\_info table**

****

// string roll = sClassRollTB.Text;

MySqlConnection con = 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+"');";

//dbo.inupdel(sql, "New Student Has Been Created");

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)

{

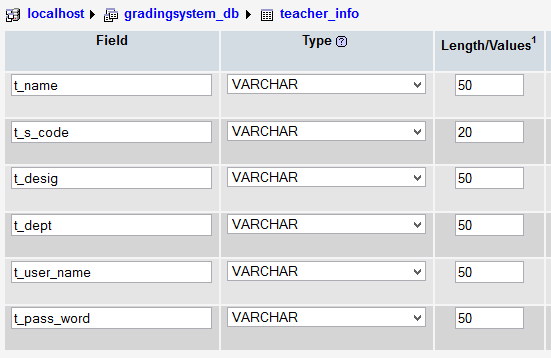
MessageBox.Show("Error Message:" + ex.Message);

con.Close();

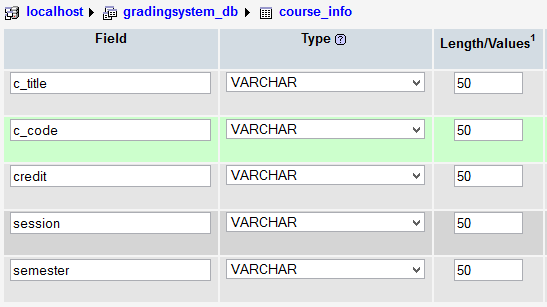
}

**Appendix-A**

**Help 1:** Database Design for ***teacher\_info.***

****

**Help 2:** Database Design for ***course\_info.***

****