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.

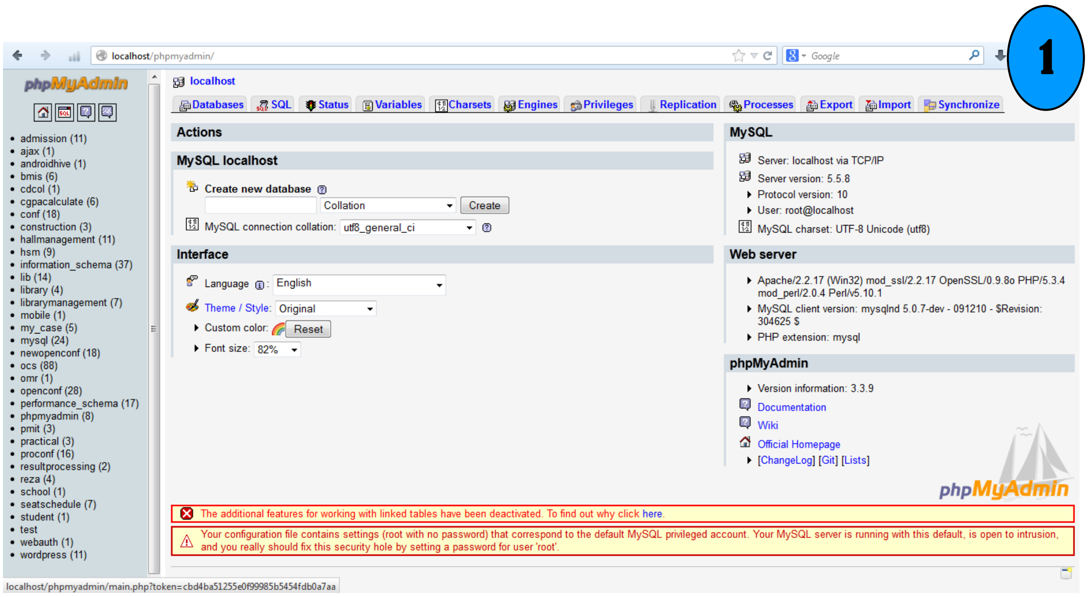
Apache HTTPD, MySQL, PHP, Perl, FileZilla FTP Server, phpMyAdmin, OpenSSL, Freetype, Webalizer, mod_perl, eAccelerator, mcrypt, SQLite, Mercury Mail Transport System, fake sendmail for windows, FPDF Class

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).



- admission (11)
- ajax (1)
- androidhive (1)
- bmis (6)
- cdcol (1)
- cgpacalculate (6)
- conf (18)
- construction (3)
- hallmanagement (11)
- hsm (9)
- information_schema (37)
- lib (14)
- library (4)
- librarymanagement (7)
- mobile (1)
- my_case (5)
- mysql (24)
- newopenconf (18)
- ocs (88)
- omr (1)
- openconf (28)
- performance_schema (17)
- phpmyadmin (8)
- pmit (3)
- practical (3)
- proconf (16)
- resultprocessing (2)
- reza (4)
- school (1)
- seatschedule (7)
- student (1)
- test
- webauth (1)
- wordpress (11)

Actions

MySQL localhost

Create new database
Collation
MySQL connection collation: utf8_general_ci

Interface

Language: English
Theme / Style: Original
Custom color:
Font size: 82%

MySQL

Server: localhost via TCP/IP
Server version: 5.5.8
Protocol version: 10
User: root@localhost
MySQL charset: UTF-8 Unicode (utf8)

Web server

Apache/2.2.17 (Win32) mod_ssl/2.2.17 OpenSSL/0.9.8o PHP/5.3.4 mod_perl/2.0.4 Perl/v5.10.1
MySQL client version: mysqlnd 5.0.7-dev - 091210 - $Revision: 304625 $
PHP extension: mysql

phpMyAdmin

Version information: 3.3.9
Documentation
Wiki
Official Homepage
[ChangeLog] [Git] [Lists]

The additional features for working with linked tables have been deactivated. To find out why click here.

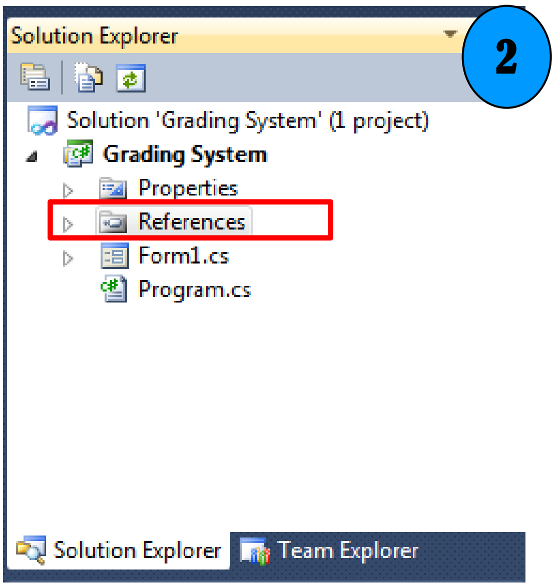
Your configuration file contains settings (root with no password) that correspond to the default MySQL privileged account. Your MySQL server is running with this default, is open to intrusion, and you really should fix this security hole by setting a password for user 'root'.

localhost/phpmyadmin/main.php?token=cbd4ba51255e0f99985b5454fdb0a7aa

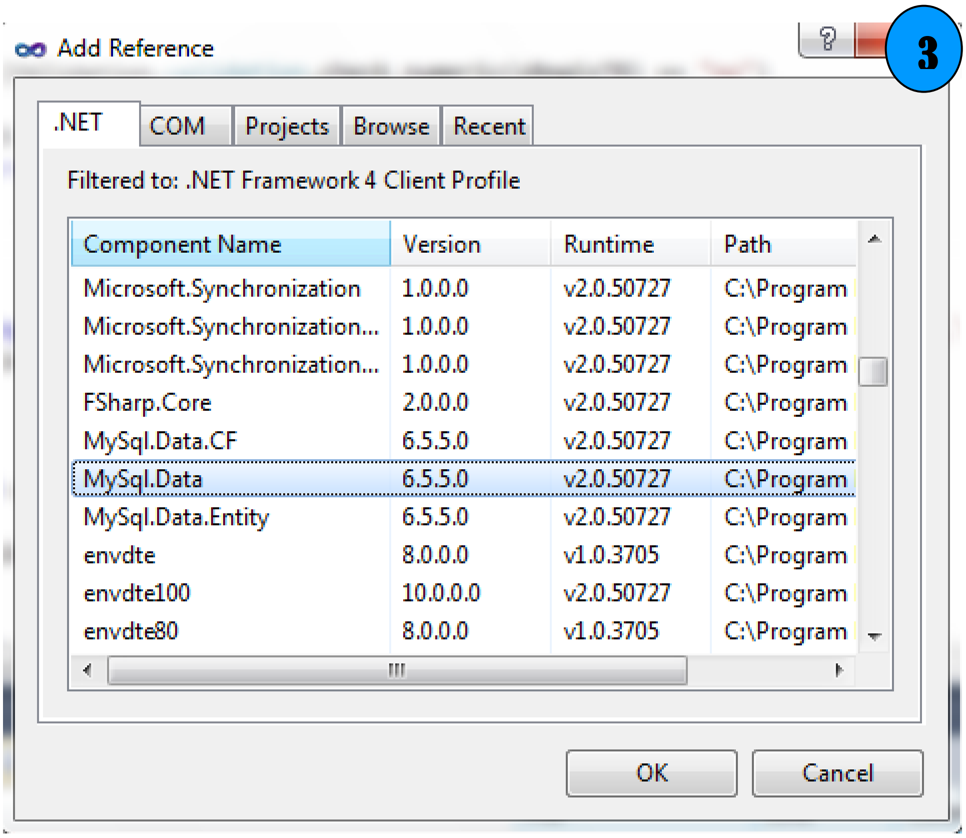
Task 3: Installation of MySQL Connector

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)



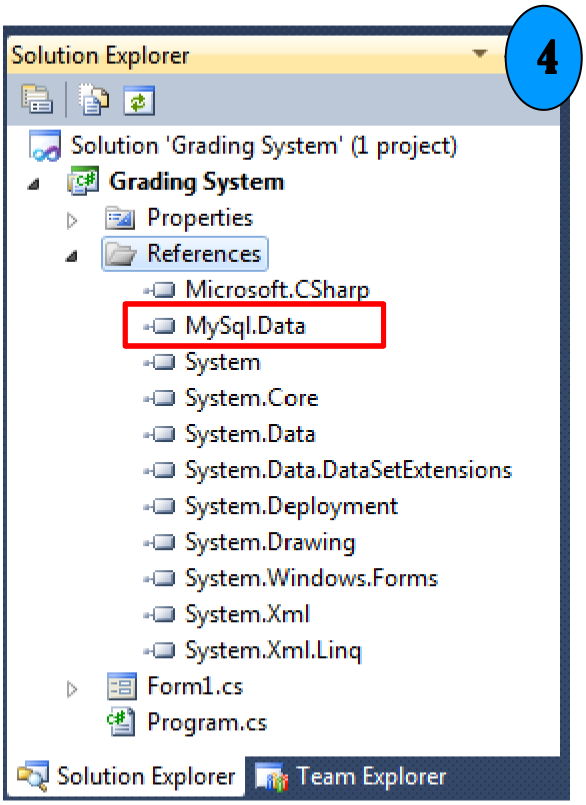
3. Right click on References and select > **Add Reference…** and see the dialog below(Fig-3).



Filtered to: .NET Framework 4 Client Profile

Component Name	Version	Runtime	Path
Microsoft.Synchronization	1.0.0.0	v2.0.50727	C:\Program
Microsoft.Synchronization...	1.0.0.0	v2.0.50727	C:\Program
Microsoft.Synchronization...	1.0.0.0	v2.0.50727	C:\Program
FSharp.Core	2.0.0.0	v2.0.50727	C:\Program
MySql.Data.CF	6.5.5.0	v2.0.50727	C:\Program
MySql.Data	6.5.5.0	v2.0.50727	C:\Program
MySql.Data.Entity	6.5.5.0	v2.0.50727	C:\Program
envdte	8.0.0.0	v1.0.3705	C:\Program
envdte100	10.0.0.0	v2.0.50727	C:\Program
envdte80	8.0.0.0	v1.0.3705	C:\Program

4. From **.NET** Tab, select the **Mysql.Data** component and click ok.
5. Now, see the references to see Mysql.Data (Fig-4).

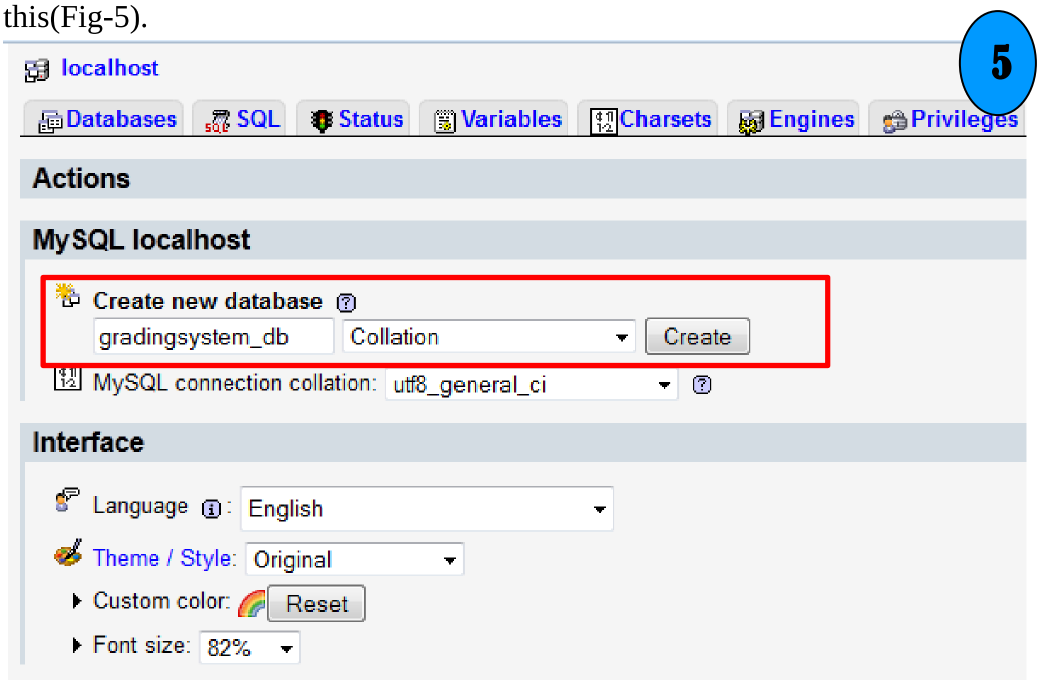


- Solution 'Grading System' (1 project)
  - **Grading System**
    - Properties
    - References
      - Microsoft.CSharp
      - MySql.Data
      - System
      - System.Core
      - System.Data
      - System.Data.DataSetExtensions
      - System.Deployment
      - System.Drawing
      - System.Windows.Forms
      - System.Xml
      - System.Xml.Linq
    - Form1.cs
    - Program.cs

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

localhost

Actions

MySQL localhost

Create new database: gradingsystem_db | Collation | Create

MySQL connection collation: utf8_general_ci

Interface

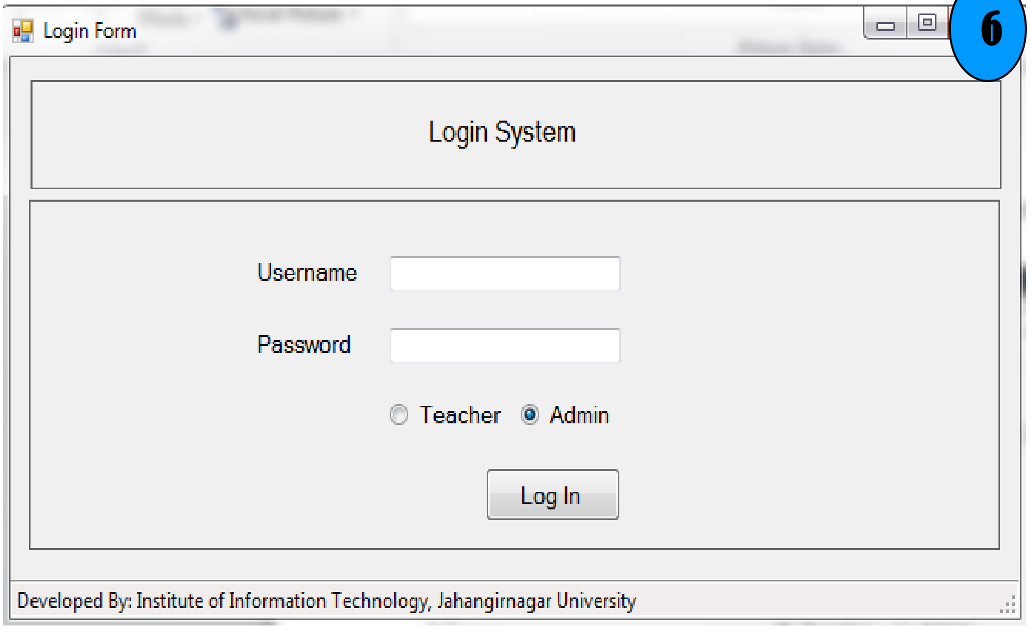
Language: English

Theme / Style: Original

Custom color: Reset

Font size: 82%

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).



6

Login Form

Login System

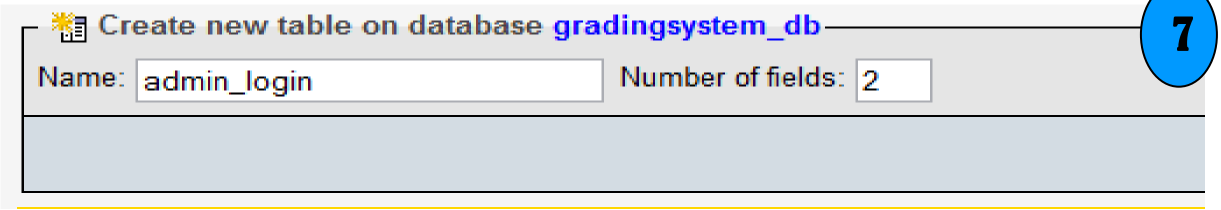
Username

Password

○ Teacher ● Admin

Developed By: Institute of Information Technology, Jahangirnagar University

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



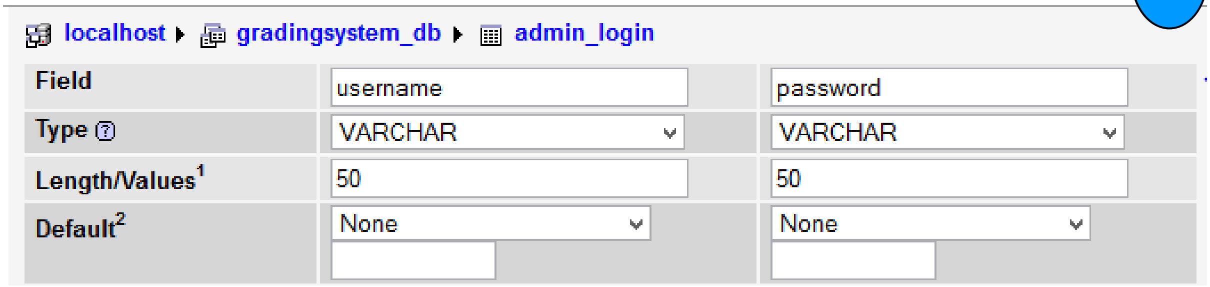
7

Create new table on database gradingsystem_db

Name: admin_login Number of fields: 2

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

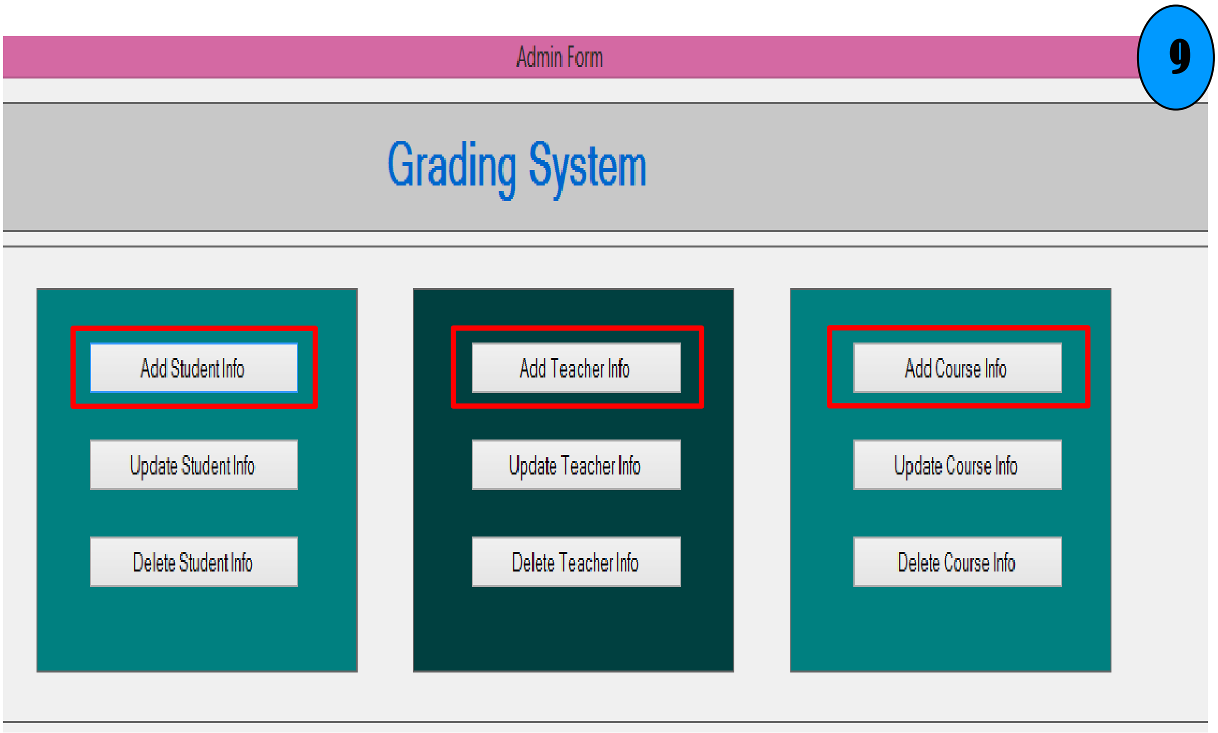
8



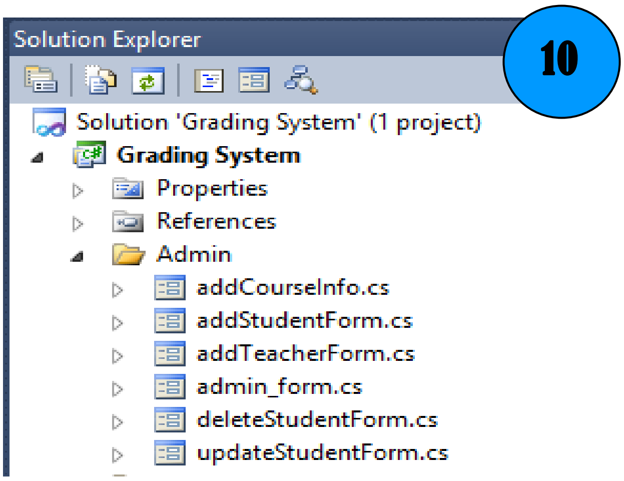
localhost ▸ gradingsystem_db ▸ admin_login

Field	username	password
Type	VARCHAR	VARCHAR
Length/Values[1]	50	50
Default[2]	None	None

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).



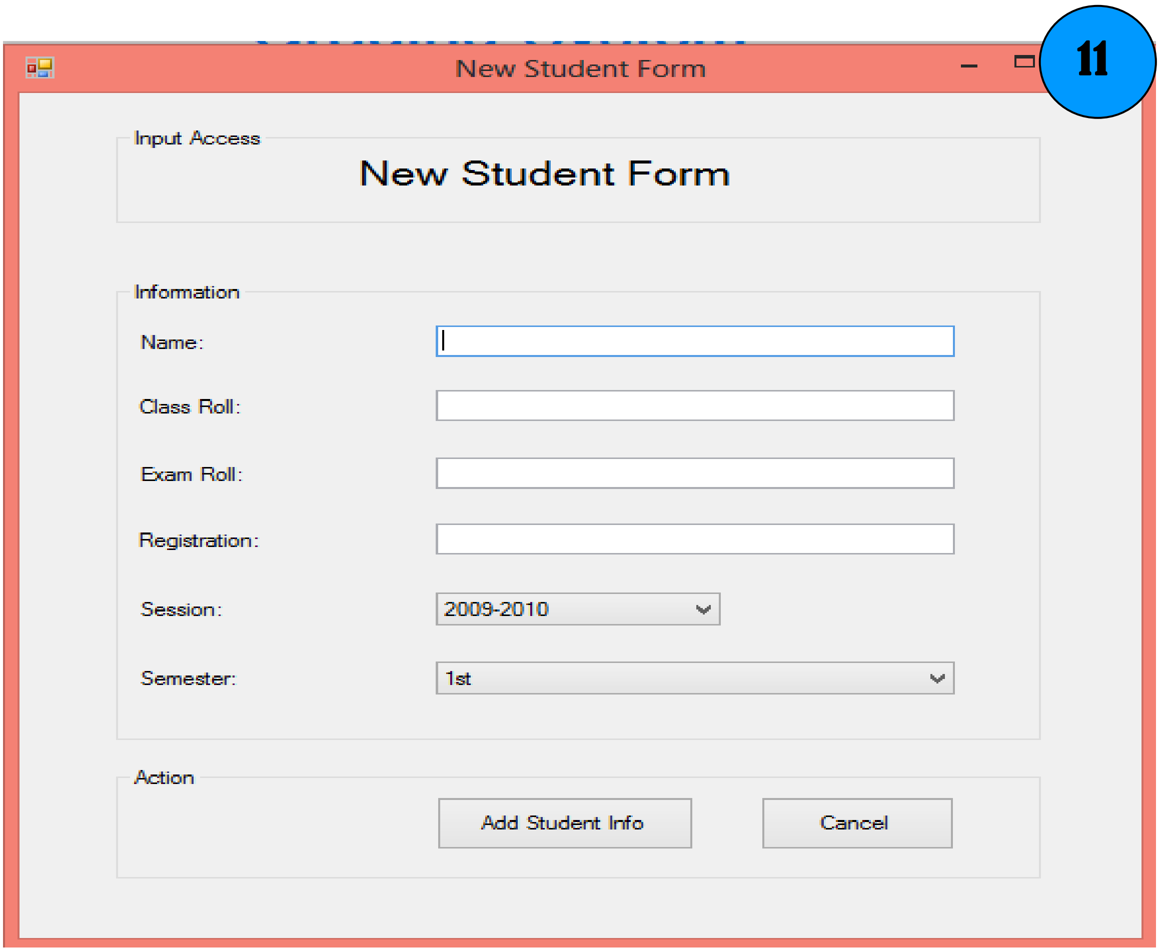
9



Solution Explorer

- Solution 'Grading System' (1 project)
  - Grading System
    - Properties
    - References
    - Admin
      - addCourseInfo.cs
      - addStudentForm.cs
      - addTeacherForm.cs
      - admin_form.cs
      - deleteStudentForm.cs
      - updateStudentForm.cs

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:

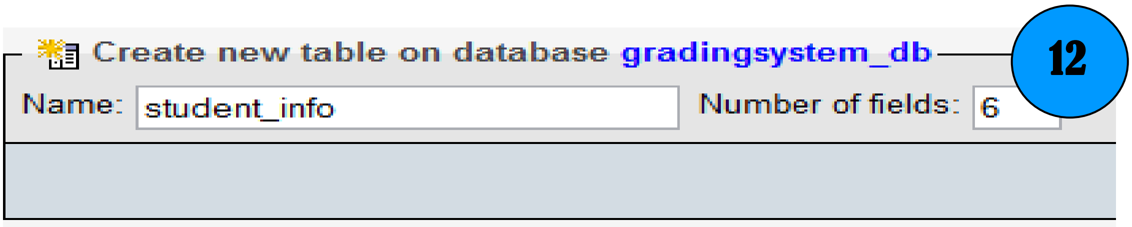
Registration:

Session: 2009-2010

Semester: 1st

Action

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

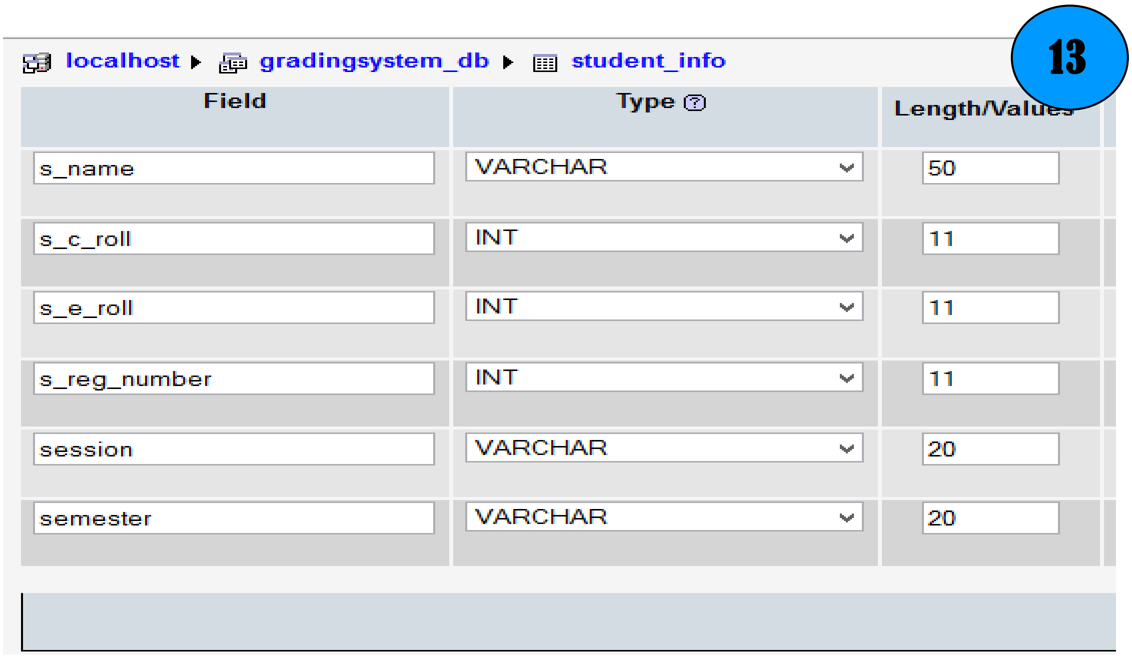


Create new table on database gradingsystem_db

Name: student_info

Number of fields: 6

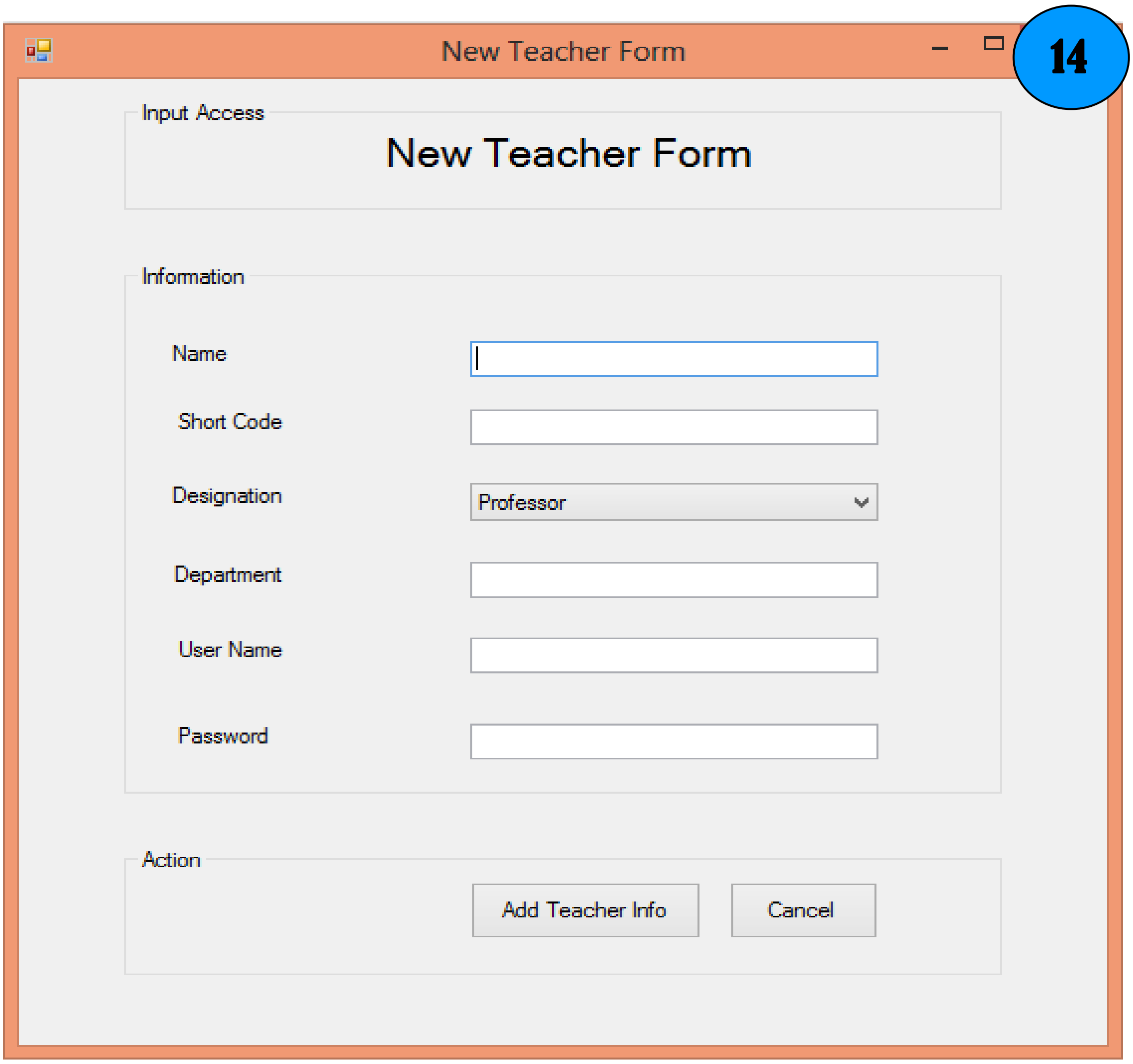
Now we add field name(column name) into our table which is like this(Fig-13)



localhost ▸ gradingsystem_db ▸ student_info

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

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).


14

New Teacher Form

Input Access

New Teacher Form

Information

Name:

Short Code:

Designation: Professor

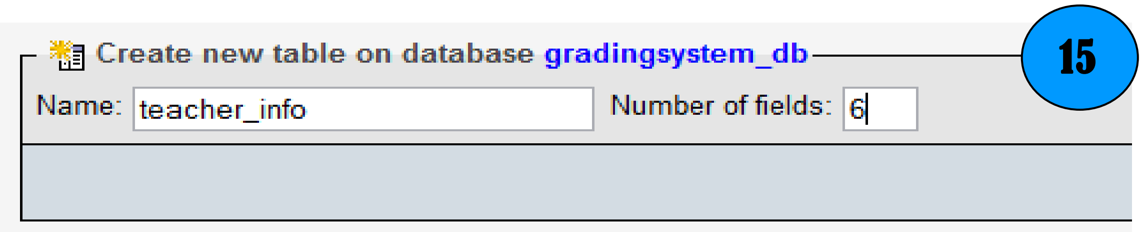
Department:

User Name:

Password:

Action

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


15

Create new table on database gradingsystem_db

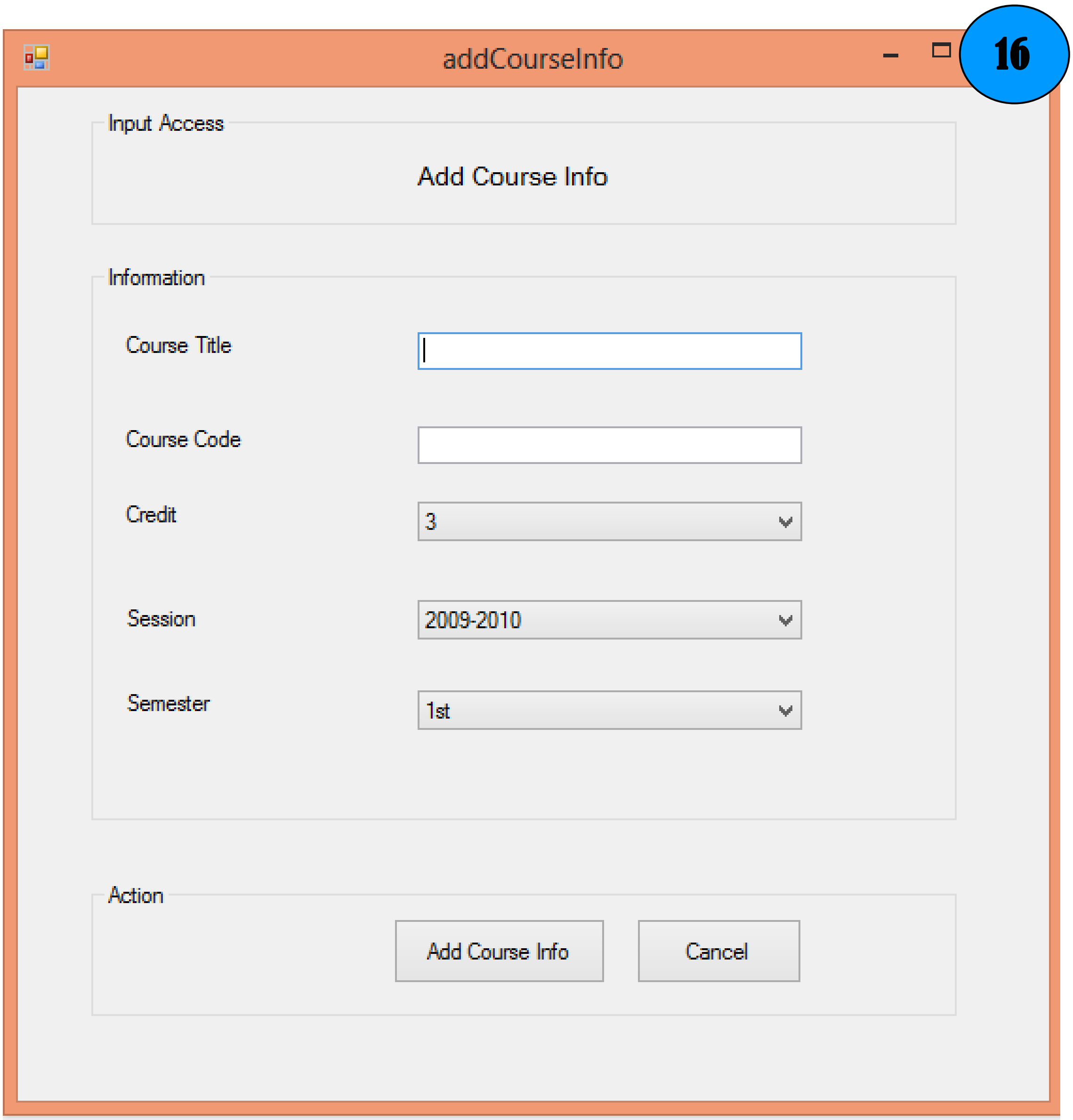
Name: teacher_info

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

Input Access

Add Course Info

Information

Course Title

Course Code

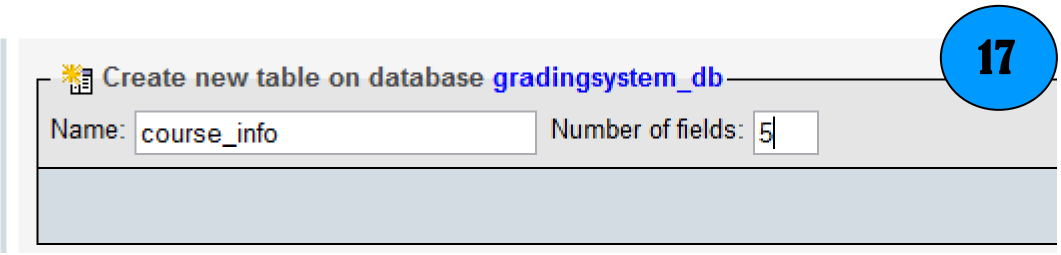
Credit: 3

Session: 2009-2010

Semester: 1st

Action

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



Create new table on database gradingsystem_db

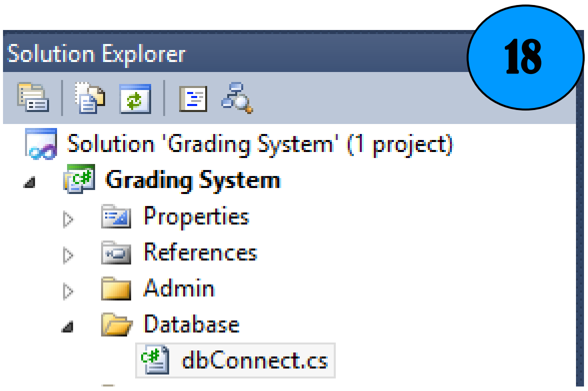
Name: course_info Number of fields: 5

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)

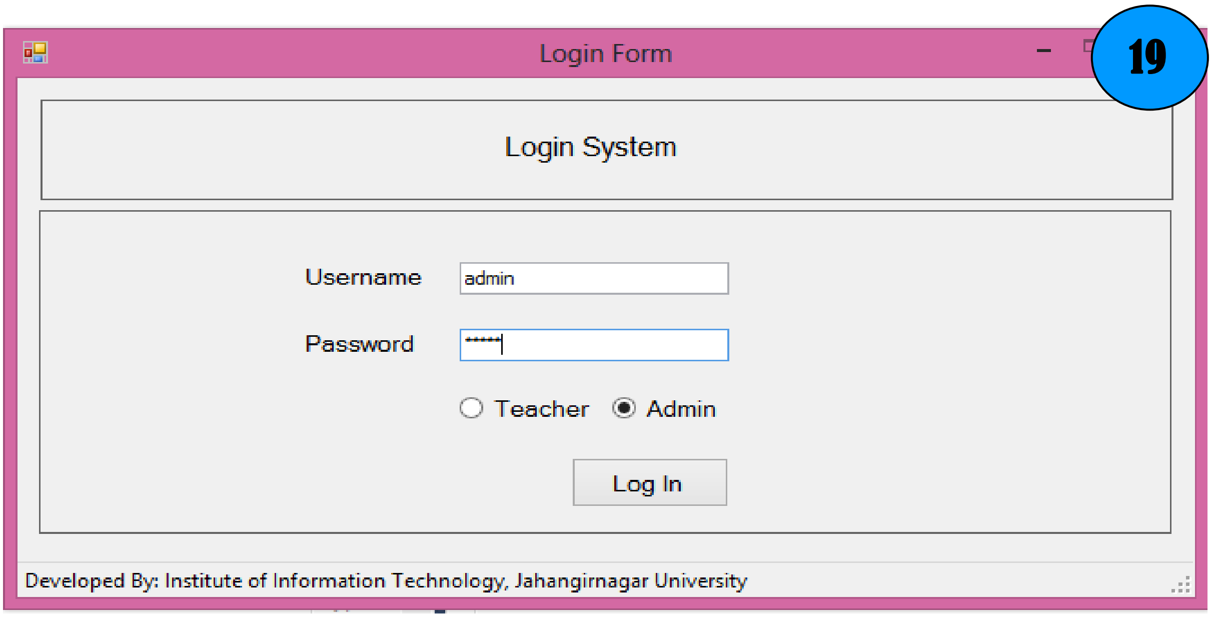


```
class dbConnect
  {

        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)



19

Login Form

Login System

Username: admin

Password: *****

○ Teacher ● Admin

Developed By: Institute of Information Technology, Jahangirnagar University

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();
```

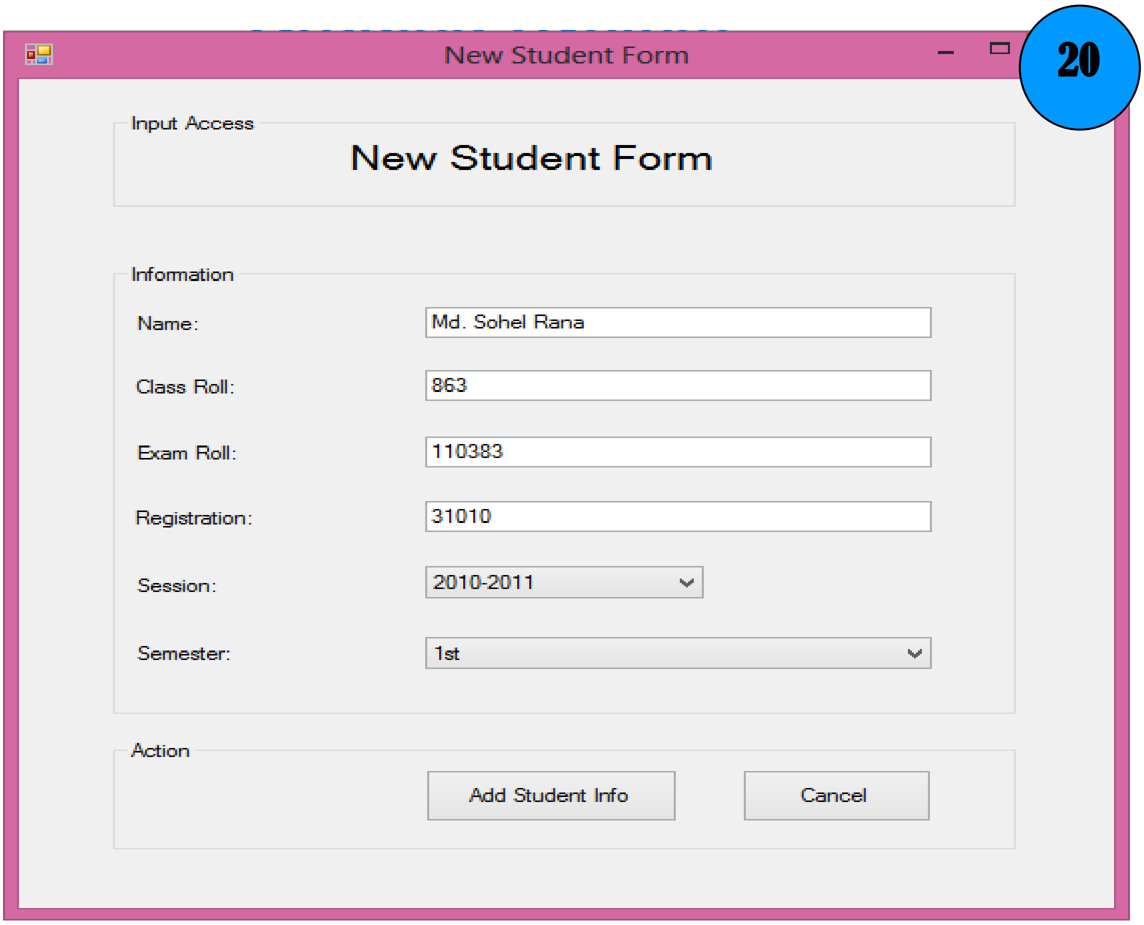
```
if (reader.Read())
 {
     if (adminRB.Checked)
     {
         Admin.admin_form aa = new Admin.admin_form();
         aa.Visible = true;
     }
     if (teacherRB.Checked)
     {

         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)



Name: Md. Sohel Rana

Class Roll: 863

Exam Roll: 110383

Registration: 31010

Session: 2010-2011

Semester: 1st

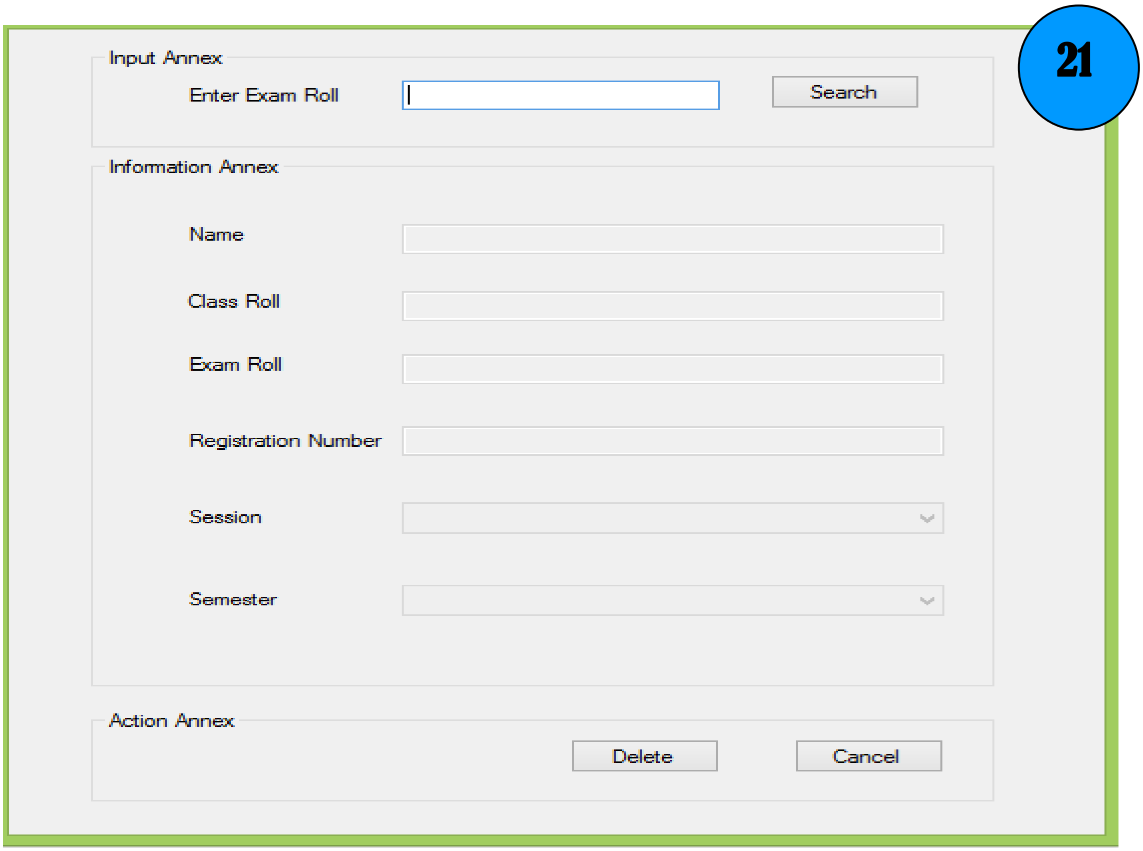
20

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");
```

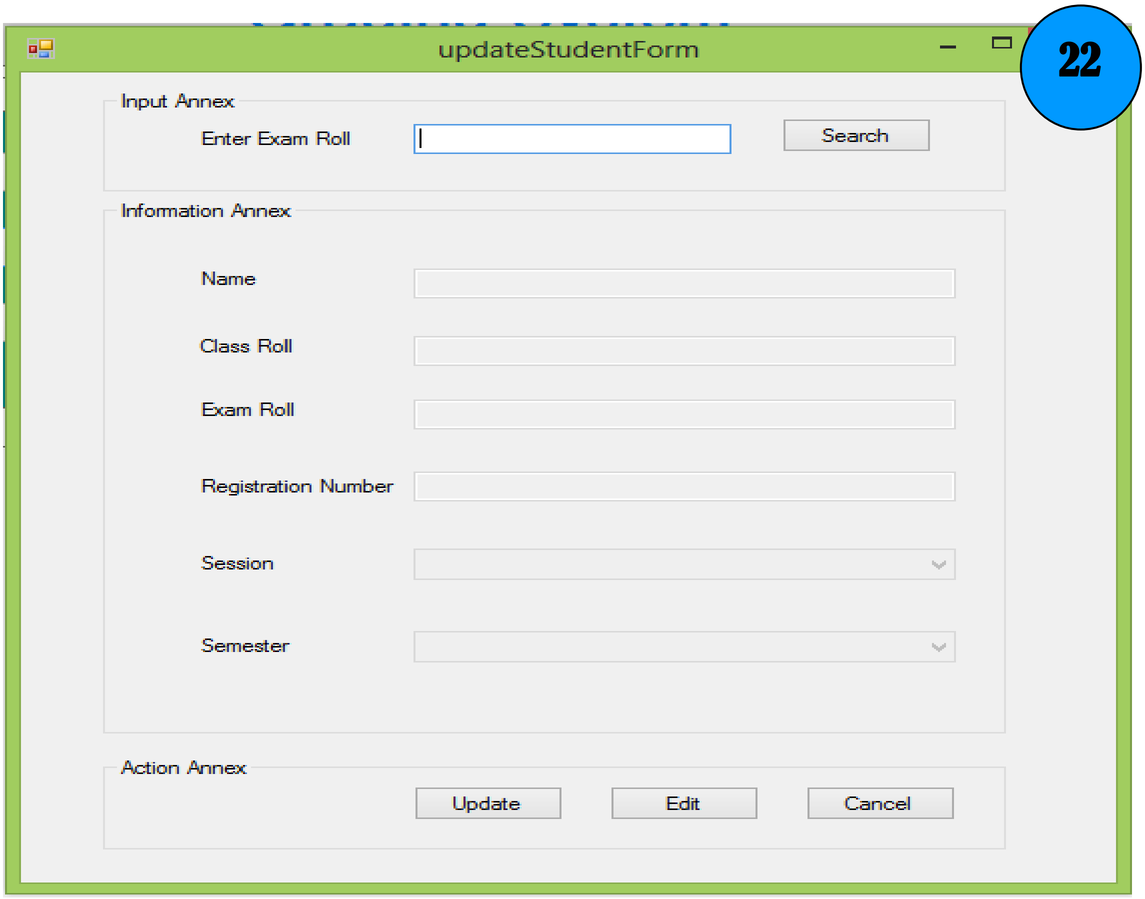
```
                return;
            }

                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).Form name will be deleteStudentForm.cs which create in Admin folder(Fig-10)



21

Practice-3: Update Student Information from **student_info** table(Do it). Make design the following picture(Fig-22). Form name will be updateStudentForm.cs which create in Admin folder(Fig-10)



updateStudentForm

Input Annex

Enter Exam Roll

Information Annex

Name

Class Roll

Exam Roll

Registration Number

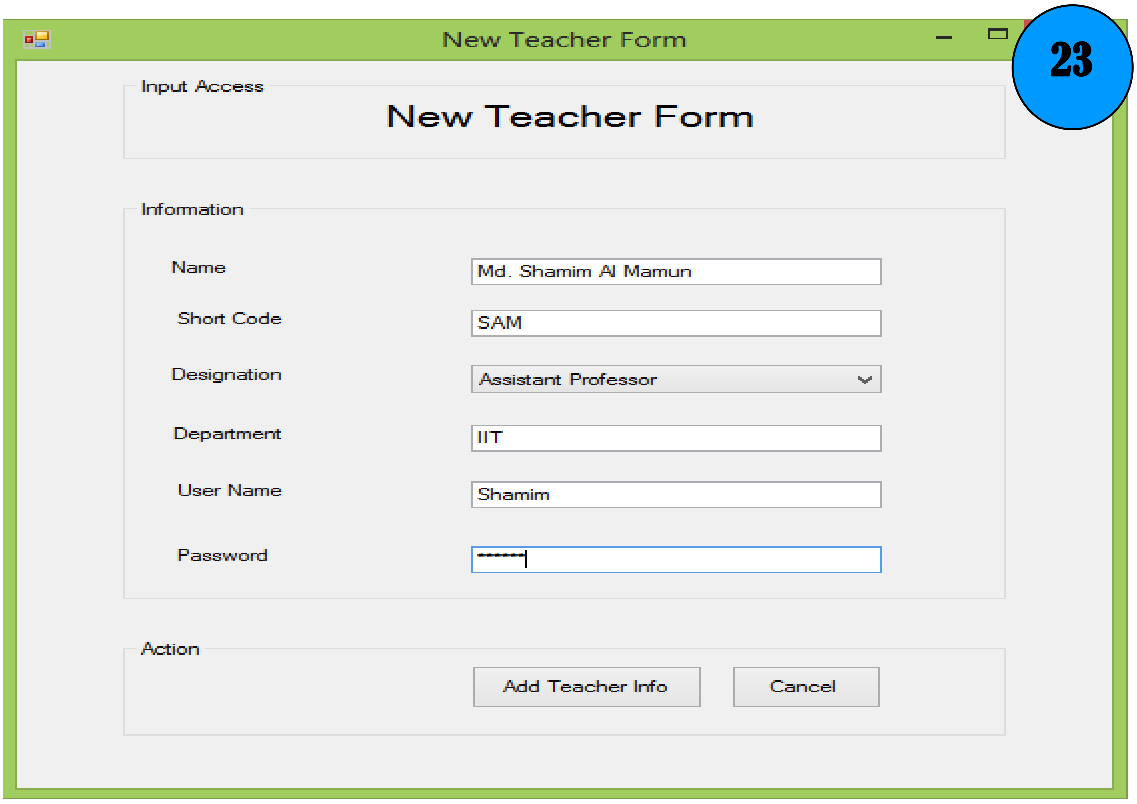
Session

Semester

Action Annex

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). Form name will be addTeacherForm.cs which create in Admin folder(Fig-10)



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	*******

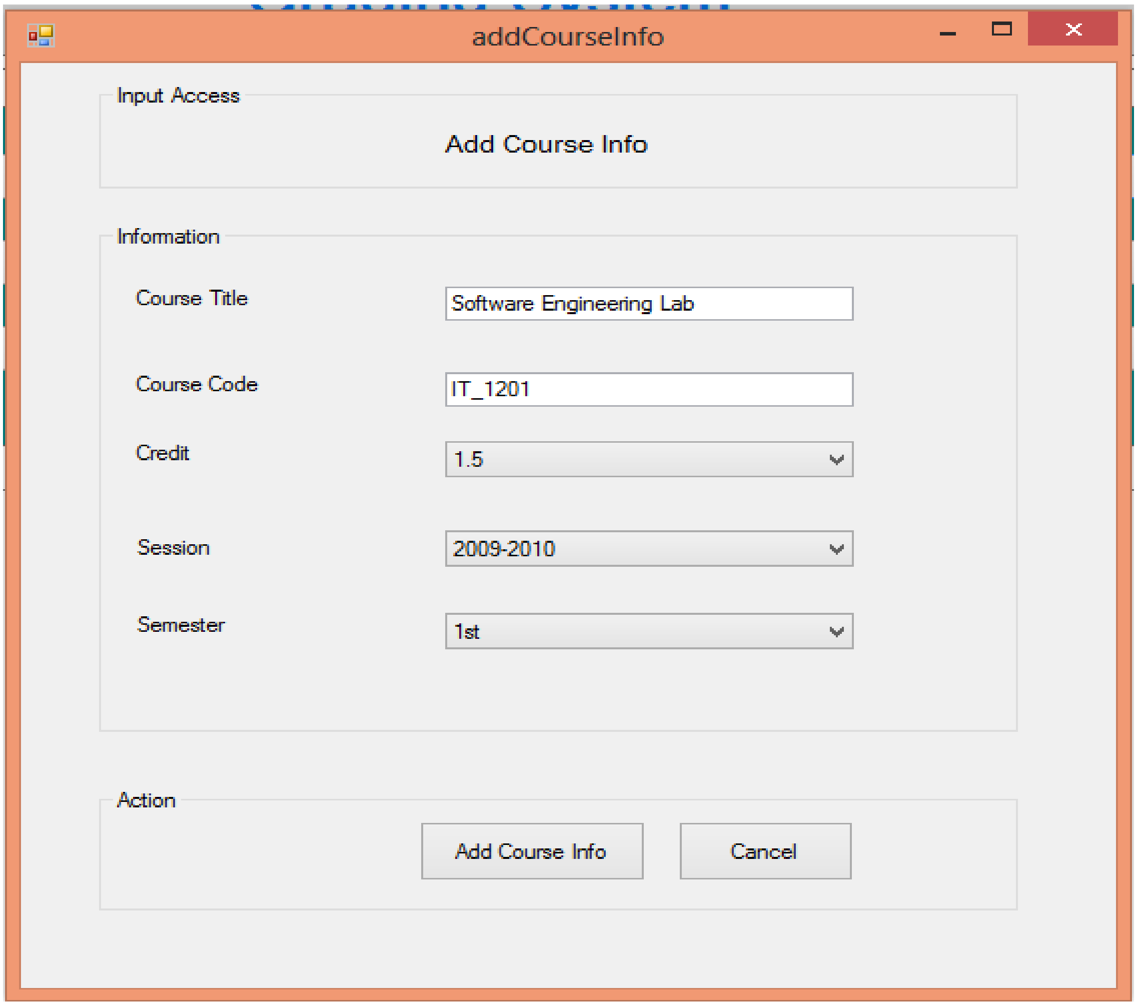
Action

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). Form name will be addCourseInfo.cs which create in Admin folder(Fig-10)



Course Title: Software Engineering Lab

Course Code: IT_1201

Credit: 1.5

Session: 2009-2010

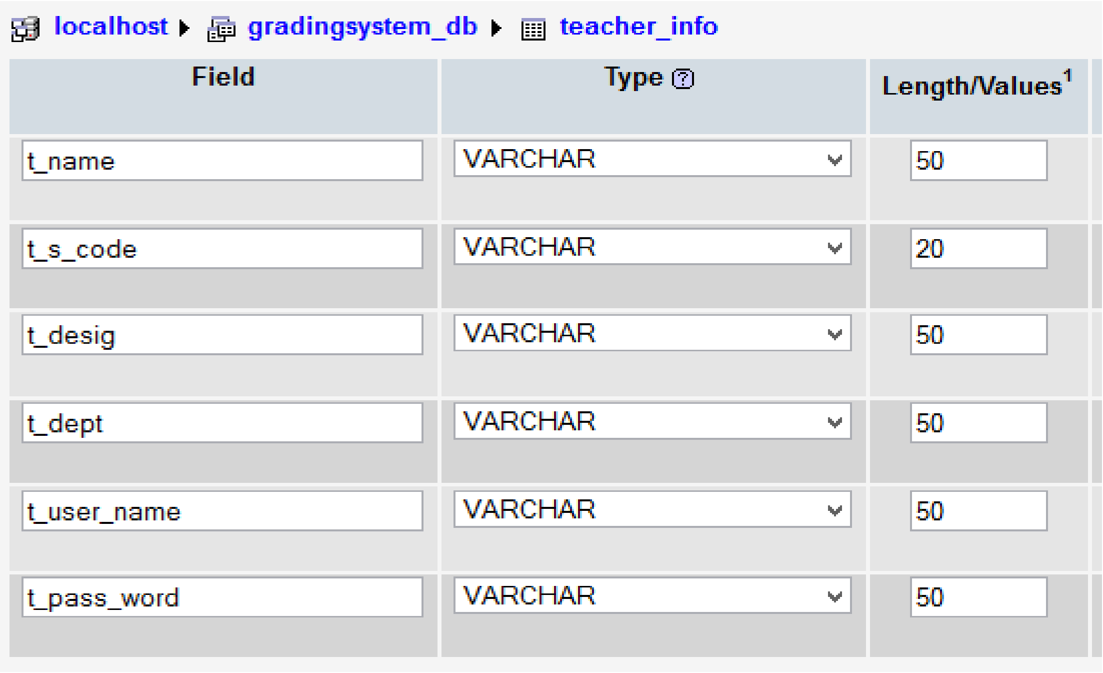
Semester: 1st

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

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

Appendix-A

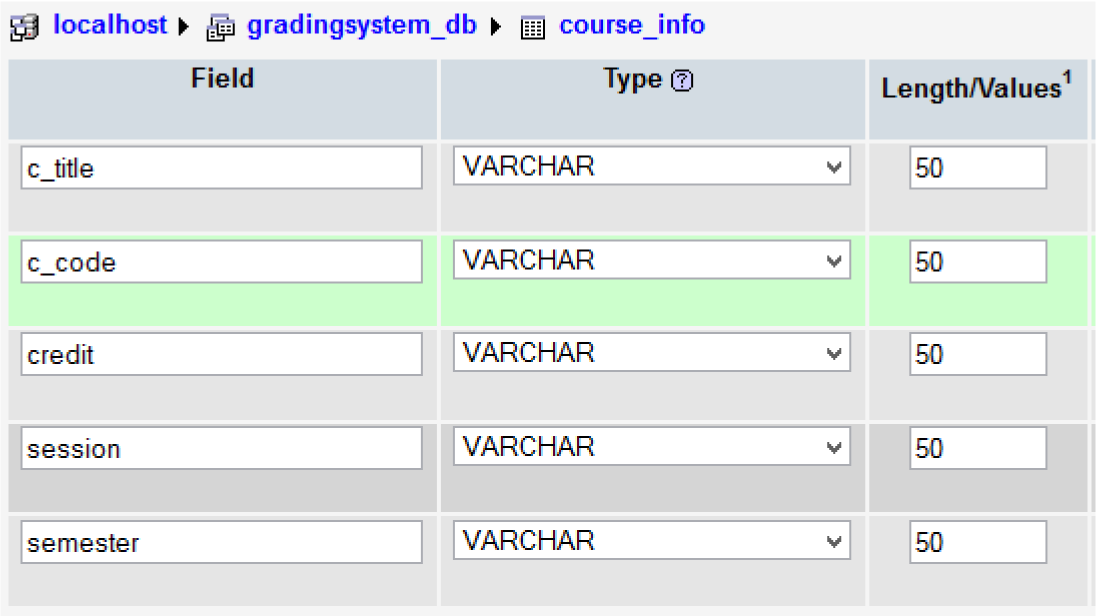
Help 1: Database Design for ***teacher_info.***



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