

**THE NATIONAL COLLEGE**

Autonomous

Jayanagar, Bangalore-560070

**PROJECT REPORT**

**ON**

**MEETUP**

**By**

**SAMARTHA L S 16NCJB643**

**LAVANYA R 16NCJB602**

**Under the guidance of**

**Prof. DEEPIKA S**

**MeetUp** ​project report submitted in partial fulfilment of the requirements of

**V Semester BCA**​, **THE NATIONAL**

**COLLEGE JAYANAGAR.**



**THE NATIONAL COLLEGE**

Autonomous

Jayanagar, Bangalore-560070

**CERTIFICATE**

This is to certify the project report titled “**MeetUp**”​ is a bonafide record of work done by **SAMARTHA L S (16NCJB643)** and **LAVANYA R (16NCJB602)** of **THE NATIONAL COLLEGE** ​, Jayanagar, Bengaluru, in partial fulfilment of the requirements of **V Semester BCA** during the year ​**2018-2019.**

**HEAD OF THE DEPARTMENT PROJECT GUIDE**

Examiners: ​Examination Centre

1. The National College,
2. Jayanagar.

Date of Examination:

**ACKNOWLEDGEMENT**

**MeetUp** ​is the project of many hands from the team. Our tribute for the successful completion of the project goes to all those who helped through their constant guidance and encouragement. The satisfaction that accompanies the success would be incomplete without thanking person who made it.

We are thankful to our beloved principal​ ​**Dr.B.R.PARINITHA**​, who encourage us to come with new and innovative ideas and for providing the environment with all facilities for completing the project.

We are also grateful to our Head of the Department ​**Prof. SHALINI C** and project guide​ ​**Prof. DEEPIKA S,**​ lecturer Department of computer science for her valuable guidance and constant support during our project development.

We extend our thanks to all our teaching staffs of department of computer science. Finally, we thank one and all who helped us directly and indirectly for the completion of our project.

**INDEX**

**CONTENT PAGE NO**

1. Admission and Fee Management System 1-6
   1. Introduction 5
   2. Abstract 5
   3. Objective and Scope 5
   4. Existing System 5
   5. Advantage 6
   6. Conclusion 6
   7. Software & Hardware Requirement 6
2. Design Specification 7-8
   1. Modular Description 7
   2. ER-Diagram 8
3. Introduction 9-11
   1. C#.NET 9
   2. Microsoft Sql Server 9
   3. Visual Studio 10
4. List of Tables 10-12
   1. Login 10
   2. Student 11
   3. Fees 12
   4. Fee\_paid 12
5. Source Code 12-34
   1. Login Form 12
   2. Homepage 15
   3. Change Password 16
   4. Create Student 18
   5. View and Update 23
   6. Pay Fee 27
   7. View Receipt 31
   8. View Due Balance 33

1. Screen Shots 35-40
2. BIBILOGRAPHY 41

**ADMISSION & FEE MANAGEMENT SYSTEM**

**INTRODUCTION**

The main purpose of this windows single user application is to develop COLLEGE ADMISSION & FEE PAYMENT MODULE by using C# and .NET language. This system mainly reduces human efforts and makes it easier to maintain records for a long time compared to the normal hand-written ones. The user can check his record details through the user search option. With the help of this system, fee calculations become an easy task. User can insert, delete, update and retrieve the data by a single click and thereby this software makes the maintenance and fee management of the college a handy task.

In the existing system, most of the records are maintained on paper. It becomes very inconvenient to modify data. Presently, the same data in different registers may have different values, which means the entries of the same data do not match. This inconsistent state does not supply the concrete information which poses a problem in the case information related to particular search record is needed.

**ABSTRACT**

In the growing world, there is a need for automation in each and every system as it not only reduces the efforts of the user but also has high accuracy and enhanced efficiency. This software helps to eliminate the errors and manual struggles faced by the user during computation. The untampered data can be stored for a longer period and can be easily accessed by a single click.

**OBJECTIVE AND SCOPE**

The main objective is to focus on various ways through which computer can be used for the benefit of the educational institution and thereby to enhance transparency and accountability. The aim and objectives of the project is to:

* Provide adequate security on records of the fee payment
* Reduce any attempt of misplacement of data
* Ease the problems of fee computations
* Make data validation easier and faster
* Computerize the system that will ensure prompt and accurate management of fee

**EXISTING SYSTEM**

The existing system is paper-based, involving high amount of paper work and man-power requirement; at times, the calculations may go wrong which may lead to computational problems at the end of the day or it may even be a mundane job to check a specific record. Even though computerized systems are used in many places, they are not that efficient and are very insecure leading to improper management. So, the current College admission system procedure is very efficient as it has many features which can overcome the problems faced in the current system.

**ADVANTAGE**

* Reduces the cost of handling the paper work associated with record keeping and decreases the manpower
* Effective use of time saving
* The user interface will be completely menu driven and user-friendly
* The user will not find any difficulties in options field
* Faster response will be present
* It shall also provide view, add, delete features for better College admission system
* It makes easier to view information about student’s fee structure without any complications and procedures

**CONCLUSION**

The college can handle full-scale computer and computer related resource. College admission system deals with all the activities done by computer such as registration and admission process and the data can be used by for other references by the staff, administration and other departments of the institutions.

**SOFTWARE REQUIREMENT**

**Processor :** 1 gigahertz (GHz) or faster 32-bit and 64-bit **Operating System :** Windows 7 and above  
**Front End :** Visual Studio 2010.NET 3.5 with C#  
**Back End :** SQL Server version 2008

**HARDWARE REQUIREMENT**

**Disk Space :** Minimum of 4.2gb for both 32-bit or 64-bit  
**RAM :** 1gb or above

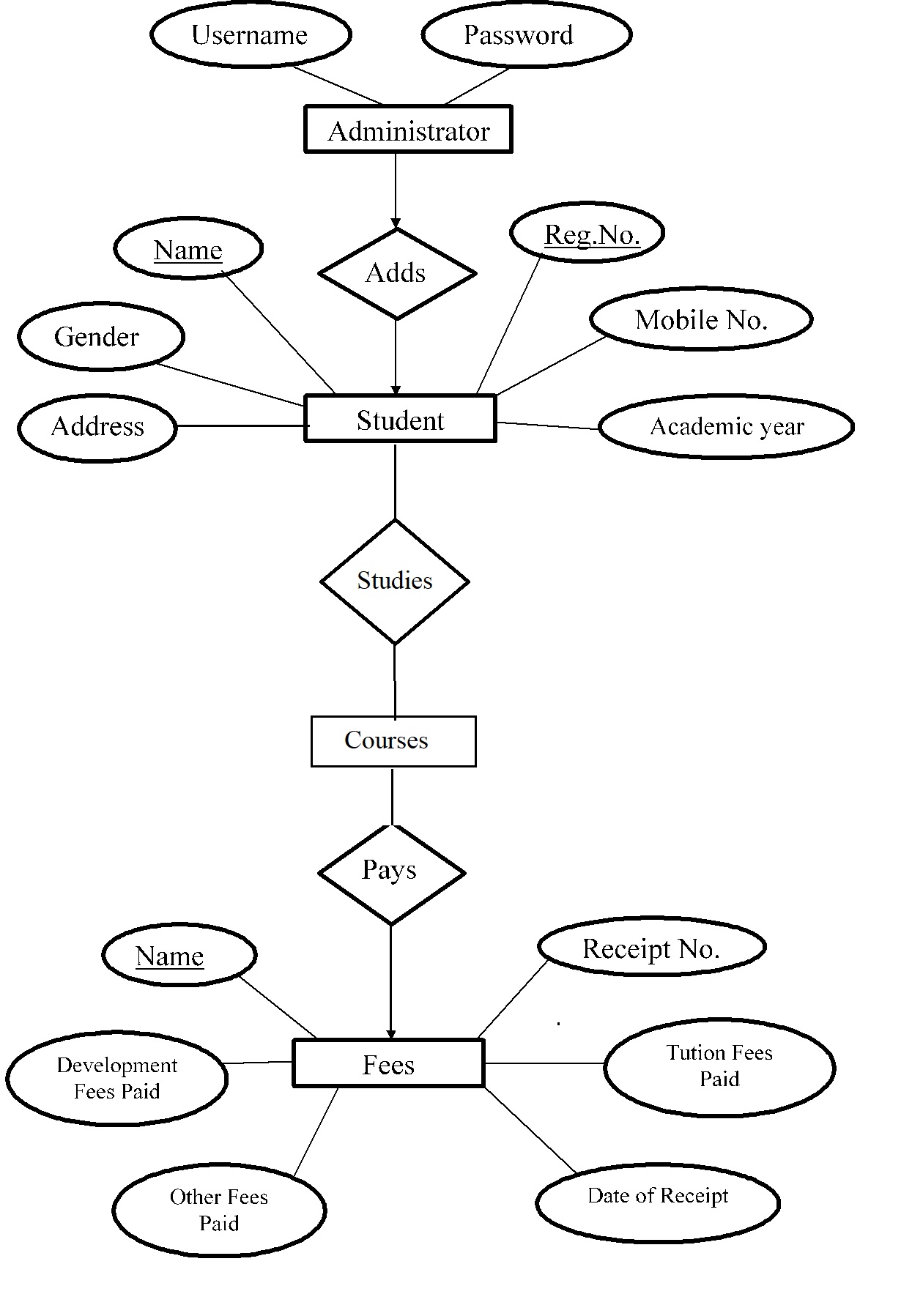
**DEISIGN SPECIFICATION**

**MODULAR DESCRIPTION**

There are nine modules in our ADMISSION & FEE MANAGEMENT SYSTEM project they are:

1. **Login Form:** Lets the user to authenticate himself to access to the admission and the fee data.
2. **Homepage:** Homepage is a collection of all the form links. It contains all the path for the forms, user can access different forms by a single click.
3. **Change Password:** This allows the user to change the password of the windows application as it is a security requirement
4. **Logout:** Logout allows the user to release the memory utilized by the software’s data and clears the system memory and goes back to the login form.
5. **Create Student:** This form of the software allows the user to create the student or admit the student by entering his/her details with a high validation function and the auto-generation of the regno takes place by concatenating the system date last 2 numbers of the year, campus name and course code with a serial number (For e.g.: 16NCJB643).
6. **View and Update Student Details:** View and update student details lets the user to update certain things like student’s e-Mail ID, Student Phone number, Parents Phone Number and the residential address. It also allows the user to search and view a particular student’s details by selecting the register number from the drop box.
7. **Pay Fee:** The student’s fee can be paid in this form by selecting the course and register number and the payment can be done in three installments and the payment can be done for a student.
8. **View Receipt**: Receipt helps to show the due amount, total fee, course, name, date of payment with the receipt number.
9. **Report**: The due amount, payed amount, register number, name and course of the student is shown in a DataGrid view.

**E-R DIAGRAM**

****

**INTRODUCTION**

**C#.NET**

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as null able value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviours that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

Encapsulated method signatures called delegates, which enable type-safe event notifications.

Properties, which serve as accessors for private member variables.

Attributes, which provide declarative metadata about types at run time.

Inline XML documentation comments.

Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

**MICROSOFT SQL SERVER**

Microsoft SQL Server is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a [software product](https://en.wikipedia.org/wiki/Software_product) with the primary function of storing and retrieving data as requested by other [software applications](https://en.wikipedia.org/wiki/Software_application)—which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](https://en.wikipedia.org/wiki/Concurrent_user).

[SQL Server Management Studio](https://en.wikipedia.org/wiki/SQL_Server_Management_Studio) is a [GUI](https://en.wikipedia.org/wiki/GUI) tool included with SQL Server 2005 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. SQL Server Management Studio replaces [Enterprise Manager](https://en.wikipedia.org/w/index.php?title=Enterprise_Manager&action=edit&redlink=1) as the primary management interface for Microsoft SQL Server since SQL Server 2005. A version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as *SQL Server Management Studio Express* (SSMSE).

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyse query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyse performance. It includes the query windows which provide a GUI based interface to write and execute queries.

[Data storage](https://en.wikipedia.org/wiki/Computer_data_storage) is a [database](https://en.wikipedia.org/wiki/Database), which is a collection of tables with [typed](https://en.wikipedia.org/wiki/Type_system) columns. SQL Server supports different data types, including [primary types](https://en.wikipedia.org/wiki/Primary_type) such as *Integer*, *Float*, *Decimal*, *Char* (including character strings), *Varchar* (variable length character strings), binary (for unstructured [blobs](https://en.wikipedia.org/wiki/Binary_large_object) of data), *Text* (for textual data) among others. The [rounding](https://en.wikipedia.org/wiki/Rounding) of floats to integers uses either Symmetric Arithmetic Rounding or Symmetric Round Down (*fix*) depending on arguments: SELECT Round(2.5, 0) gives 3.

Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs).

**VISUAL STUDIO**

Visual Studio is a complete set of development tools for building ASP.NET Web applications, XML Web Services, desktop applications, and mobile applications. Visual Basic, Visual C#, and Visual C++ all use the same integrated development environment (IDE), which enables tool sharing and eases the creation of mixed-language solutions. In addition, these languages use the functionality of the .NET Framework, which provides access to key technologies that simplify the development of ASP Web applications and XML Web Services.

Microsoft Visual Studio is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program) for [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), as well as [web sites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Native_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

Visual Studio includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (the [code completion](https://en.wikipedia.org/wiki/Code_completion) component) as well as [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring). [The integrated debugger](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a [code profiler](https://en.wikipedia.org/wiki/Profiling_(computer_programming)), forms designer for building [GUI](https://en.wikipedia.org/wiki/GUI) applications, [web designer](https://en.wikipedia.org/wiki/Web_designer), [class](https://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](https://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source control](https://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](https://en.wikipedia.org/wiki/Subversion_(software))) and adding new toolsets like editors and visual designers for [domain-specific languages](https://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development lifecycle](https://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Team Foundation Server](https://en.wikipedia.org/wiki/Team_Foundation_Server) client: Team Explorer).

Visual Studio supports different [programming languages](https://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](https://en.wikipedia.org/wiki/C_(programming_language)),[C++](https://en.wikipedia.org/wiki/C%2B%2B) and [C++/CLI](https://en.wikipedia.org/wiki/C%2B%2B/CLI) (via [Visual C++](https://en.wikipedia.org/wiki/Visual_C%2B%2B)), [VB.NET](https://en.wikipedia.org/wiki/VB.NET) (via [Visual Basic .NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET)), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (via [Visual C#](https://en.wikipedia.org/wiki/Visual_C_Sharp)), [F#](https://en.wikipedia.org/wiki/F_Sharp_(programming_language)) (as of Visual Studio 2010) and [TypeScript](https://en.wikipedia.org/wiki/TypeScript) (as of Visual Studio 2013 Update 2). Support for other languages such as [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), [Node.js](https://en.wikipedia.org/wiki/Node.js), and [M](https://en.wikipedia.org/wiki/MUMPS) among others is available via language services installed separately. It also supports [XML](https://en.wikipedia.org/wiki/XML)/[XSLT](https://en.wikipedia.org/wiki/XSLT), [HTML](https://en.wikipedia.org/wiki/HTML)/[XHTML](https://en.wikipedia.org/wiki/XHTML), [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) (and [J#](https://en.wikipedia.org/wiki/J_Sharp)) were supported in the past.

**LIST OF TABLES**

**LOGIN TABLE**

|  |  |
| --- | --- |
| **NAME** | **DATATYPE** |
| Username | varchar(10) |
| u\_password | varchar(10) |

**FEE\_PAID TABLE**

|  |  |
| --- | --- |
| **NAME** | **DATATYPE** |
| Regno | varchar(10) |
| s\_name | varchar(50) |
| s\_phone | numeric(10) |
| Dop | Smalldatetime |
| recipt\_no | numeric(5) |
| course\_name | varchar(5) |
| total\_fee | Int |
| Paid | Int |
| Due | int |

**STUDENT TABLE**

|  |  |
| --- | --- |
| **NAME** | **DATATYPE** |
| Regno | varchar(10) |
| s\_no | numeric(3) |
| Doj | Smalldatetime |
| course\_name | varchar(6) |
| Coursecode | Char |
| s\_name | varchar(50) |
| s\_address | Text |
| adhaar\_numb | numeric(12) |
| s\_phone | numeric(10) |
| email\_id | varchar(50) |
| p\_phone | numeric(10) |
| Dob | Date |
| f\_name | varchar(50) |
| m\_name | varchar(50) |
| Gender | varchar(6) |

**FEES TABLE**

|  |  |
| --- | --- |
| **NAME** | **DATATYPE** |
| sl\_no | int |
| course\_name | varchar(5) |
| total\_fee | Int |
| admission\_fee | Int |
| registration\_fee | Int |
| tuition\_fee | Int |
| lab\_fee | Int |
| library\_fee | Int |
| sports\_fee | Int |

**SOURCE CODE**

**LOGIN FORM**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Configuration;

namespace Fee\_Management

{

public partial class Login : Form

{

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

public Login()

{

InitializeComponent();

}

private void but\_login\_Click(object sender, EventArgs e)

{

try

{

if (txt\_password.TextLength > 8)

{

SqlDataAdapter sda = new SqlDataAdapter("select count(\*) from login where username = '" + txt\_login.Text + "' and u\_password = '" + txt\_password.Text + "'", con);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows[0][0].ToString() == "1")

{

this.Hide();

Homepage hp = new Homepage(txt\_login.Text);

hp.Show();

}

else

{

MessageBox.Show("Please check the username or password");

txt\_login.Clear();

txt\_password.Clear();

}

}

else

{

MessageBox.Show("Minimum Characters should be 8");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

con.Close();

}

}

private void but\_exit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void txt\_password\_TextChanged(object sender, EventArgs e)

{

if(txt\_password.TextLength < 8)

{

ep\_pass.SetError(this.txt\_password, "Minimum 8 Character should be Entered");

}

else

{

ep\_pass.Clear();

}

}

}

}

**HOMEPAGE**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Fee\_Management

{

public partial class Homepage : Form

{

public Homepage()

{

InitializeComponent();

}

public Homepage(String User)

{

InitializeComponent();

lab\_user.Text = User;

}

private void but\_create\_Click(object sender, EventArgs e)

{

Create\_Student cs = new Create\_Student();

cs.Show();

}

private void but\_pay\_Click(object sender, EventArgs e)

{

Pay\_Fee pf = new Pay\_Fee();

pf.Show();

}

private void but\_logout\_Click(object sender, EventArgs e)

{

Login l = new Login();

l.Show();

}

private void link\_logout\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

this.Close();

Login l = new Login();

l.Show();

}

private void link\_change\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

this.Hide();

Change\_Password cp = new Change\_Password();

cp.Show();

}

private void but\_update\_Click(object sender, EventArgs e)

{

Edit\_Student es = new Edit\_Student();

es.Show();

}

private void but\_due\_Click(object sender, EventArgs e)

{

View\_Due vd = new View\_Due();

vd.Show();

}

private void but\_view\_Click(object sender, EventArgs e)

{

Acknodwledge a = new Acknodwledge();

a.Show();

}

}

}

**CHANGE PASSWORD**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Fee\_Management

{

public partial class Change\_Password : Form

{

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

public Change\_Password()

{

InitializeComponent();

}

private void but\_update\_Click(object sender, EventArgs e)

{

try

{

if (txt\_new.TextLength > 8)

{

SqlDataAdapter sda = new SqlDataAdapter("select count(\*) from login where username ='" + txt\_username.Text + "' and u\_password ='" + txt\_current.Text + "'", con);

DataTable dt = new DataTable();

sda.Fill(dt);

error\_password.Clear();

if (dt.Rows[0][0].ToString() == "1")

{

if (txt\_new.Text == txt\_confirm.Text)

{

SqlDataAdapter cc = new SqlDataAdapter("update login set u\_password = '" + txt\_new.Text + "' where username = '" + txt\_username.Text + "' and u\_password = '" + txt\_current.Text + "'", con);

DataTable df = new DataTable();

cc.Fill(df);

MessageBox.Show("Password Changed Successfully", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

Login l = new Login();

l.Show();

}

else

{

error\_password.SetError(txt\_new, "Unmatch Password");

error\_password.SetError(txt\_confirm, "Unmatch Password");

}

}

else

{

error\_password.SetError(txt\_username, "Incorrect Username");

error\_password.SetError(txt\_current, "Inccorrect Password");

}

}

else

{

error\_password.SetError(txt\_new, "Minimum character 8");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

con.Close();

}

}

private void txt\_new\_TextChanged(object sender, EventArgs e)

{

if (txt\_new.TextLength < 8)

{

error\_password.SetError(txt\_new, "Minimum character 8");

}

else

{

error\_password.Clear();

}

}

private void but\_exit\_Click(object sender, EventArgs e)

{

this.Close();

Homepage hp = new Homepage();

hp.Show();

}

}

}

**CREATE STUDENT**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Text.RegularExpressions;

namespace Fee\_Management

{

public partial class Create\_Student : Form

{

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

public Create\_Student()

{

InitializeComponent();

}

void DataClear()

{

txt\_sname.Clear();

rb\_female.Checked = false;

rb\_male.Checked = false;

combo\_course.SelectedIndex = -1;

txt\_address.Clear();

txt\_sphone.Clear();

txt\_email.Clear();

txt\_adhar.Clear();

txt\_fname.Clear();

txt\_mname.Clear();

txt\_pnumb.Clear();

error\_mail.Clear();

error\_phone.Clear();

}

private void txt\_email\_Leave(object sender, EventArgs e)

{

string pattern = "^([0-9a-zA-Z]([-\\.\\w]\*[0-9a-zA-Z])\*@([0-9a-zA-Z][-\\w]\*[0-9a-zA-Z]\\.)+[a-zA-Z]{2,9})$";

if (Regex.IsMatch(txt\_email.Text, pattern))

{

error\_mail.Clear();

}

else

{

error\_mail.SetError(this.txt\_email, "Please check the mail id entered");

return;

}

}

private void txt\_sphone\_TextChanged(object sender, EventArgs e)

{

string pattern = (@"^((\+){0,1}91(\s){0,1}(\-){0,1}(\s){0,1}){0,1}(9|8|7)[0-9](\s){0,1}(\-){0,1}(\s){0,1}[1-9]{1}[0-9]{7}$");

if (Regex.IsMatch(txt\_sphone.Text, pattern))

{

error\_phone.Clear();

}

else

{

error\_phone.SetError(this.txt\_sphone, "Check the phone number entered");

}

}

private void txt\_pnumb\_TextChanged(object sender, EventArgs e)

{

string pattern = (@"^((\+){0,1}91(\s){0,1}(\-){0,1}(\s){0,1}){0,1}(9|8|7)[0-9](\s){0,1}(\-){0,1}(\s){0,1}[1-9]{1}[0-9]{7}$");

if (Regex.IsMatch(txt\_pnumb.Text, pattern))

{

error\_phone.Clear();

}

else

{

error\_phone.SetError(this.txt\_pnumb, "Check the phone number entered");

}

}

private void but\_submit\_Click(object sender, EventArgs e)

{

string pattern1 = "^([0-9a-zA-Z]([-\\.\\w]\*[0-9a-zA-Z])\*@([0-9a-zA-Z][-\\w]\*[0-9a-zA-Z]\\.)+[a-zA-Z]{2,9})$";

string pattern2 = (@"^((\+){0,1}91(\s){0,1}(\-){0,1}(\s){0,1}){0,1}(9|8|7)[0-9](\s){0,1}(\-){0,1}(\s){0,1}[1-9]{1}[0-9]{7}$");

if (!Regex.IsMatch(txt\_email.Text, pattern1))

{

error\_mail.SetError(this.txt\_email, "Please check the mail id entered");

return;

}

if (!Regex.IsMatch(txt\_sphone.Text, pattern2))

{

error\_phone.SetError(this.txt\_sphone, "Check the phone number entered");

return;

}

if (!Regex.IsMatch(txt\_pnumb.Text, pattern2))

{

error\_phone.SetError(this.txt\_pnumb, "Check the phone number entered");

return;

}

string gender;

if (rb\_male.Checked)

{

gender = "Male";

}

else

{

gender = "Female";

}

if (String.IsNullOrEmpty(txt\_sname.Text) || String.IsNullOrEmpty(txt\_address.Text) || String.IsNullOrEmpty(txt\_email.Text) || String.IsNullOrEmpty(txt\_fname.Text) || String.IsNullOrEmpty(txt\_mname.Text) || String.IsNullOrEmpty(txt\_pnumb.Text) || String.IsNullOrEmpty(txt\_sphone.Text) || String.IsNullOrEmpty(combo\_course.Text) || txt\_adhar.TextLength < 12)

{

MessageBox.Show("Fields cannot be left blank", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

try

{

if (combo\_course.SelectedIndex == 0)

{

con.Open();

SqlCommand sc = new SqlCommand("insert into student(coursecode,s\_no,s\_name,s\_address,s\_phone,email\_id,adhaar\_numb,f\_name,m\_name,p\_phone,course\_name,gender,dob)values('B',NEXT VALUE FOR seq\_b,'" + txt\_sname.Text + "','" + txt\_address.Text + "'," + txt\_sphone.Text + ",'" + txt\_email.Text + "'," + txt\_adhar.Text + ",'" + txt\_fname.Text + "','" + txt\_mname.Text + "'," + txt\_pnumb.Text + ",'" + combo\_course.SelectedItem + "','" + gender + "','" + this.dob.Text + "');", con);

sc.ExecuteNonQuery();

SqlCommand sc1 = new SqlCommand("insert into fee\_paid(paid,s\_name,s\_phone,course\_name)values(0,'" + txt\_sname.Text + "'," + txt\_sphone.Text + ",'" + combo\_course.SelectedItem + "');", con);

sc1.ExecuteNonQuery();

SqlCommand sp = new SqlCommand("update student set regno= CONCAT(CONCAT(RIGHT(year(doj),2)+'NCJ',+coursecode),s\_no) from student", con);

sp.ExecuteNonQuery();

con.Close();

}

else if (combo\_course.SelectedIndex == 1)

{

con.Open();

SqlCommand sc = new SqlCommand("insert into student(coursecode,s\_no,s\_name,s\_address,s\_phone,email\_id,adhaar\_numb,f\_name,m\_name,p\_phone,course\_name,gender,dob)values('C',NEXT VALUE FOR seq\_c,'" + txt\_sname.Text + "','" + txt\_address.Text + "'," + txt\_sphone.Text + ",'" + txt\_email.Text + "'," + txt\_adhar.Text + ",'" + txt\_fname.Text + "','" + txt\_mname.Text + "'," + txt\_pnumb.Text + ",'" + combo\_course.SelectedItem + "','" + gender + "','" + this.dob.Text + "');", con);

sc.ExecuteNonQuery();

SqlCommand sc1 = new SqlCommand("insert into fee\_paid(paid,s\_name,s\_phone,course\_name)values(0,'" + txt\_sname.Text + "'," + txt\_sphone.Text + ",'" + combo\_course.SelectedItem + "');", con);

sc1.ExecuteNonQuery();

SqlCommand sp = new SqlCommand("update student set regno= CONCAT(CONCAT(RIGHT(year(doj),2)+'NCJ',+coursecode),s\_no) from student", con);

sp.ExecuteNonQuery();

con.Close();

}

else if (combo\_course.SelectedIndex == 2)

{

con.Open();

SqlCommand sc = new SqlCommand("insert into student(coursecode,s\_no,s\_name,s\_address,s\_phone,email\_id,adhaar\_numb,f\_name,m\_name,p\_phone,course\_name,gender,dob)values('S',NEXT VALUE FOR seq\_s,'" + txt\_sname.Text + "','" + txt\_address.Text + "'," + txt\_sphone.Text + ",'" + txt\_email.Text + "'," + txt\_adhar.Text + ",'" + txt\_fname.Text + "','" + txt\_mname.Text + "'," + txt\_pnumb.Text + ",'" + combo\_course.SelectedItem + "','" + gender + "','" + this.dob.Text + "');", con);

sc.ExecuteNonQuery();

SqlCommand sc1 = new SqlCommand("insert into fee\_paid(paid,s\_name,s\_phone,course\_name)values(0,'" + txt\_sname.Text + "'," + txt\_sphone.Text + ",'" + combo\_course.SelectedItem + "');", con);

sc1.ExecuteNonQuery();

SqlCommand sp = new SqlCommand("update student set regno= CONCAT(CONCAT(RIGHT(year(doj),2)+'NCJ',+coursecode),s\_no) from student", con);

sp.ExecuteNonQuery();

con.Close();

}

else

{

con.Open();

SqlCommand sc = new SqlCommand("insert into student(coursecode,s\_no,s\_name,s\_address,s\_phone,email\_id,adhaar\_numb,f\_name,m\_name,p\_phone,course\_name,gender,dob)values('A',NEXT VALUE FOR seq\_a,'" + txt\_sname.Text + "','" + txt\_address.Text + "'," + txt\_sphone.Text + ",'" + txt\_email.Text + "'," + txt\_adhar.Text + ",'" + txt\_fname.Text + "','" + txt\_mname.Text + "'," + txt\_pnumb.Text + ",'" + combo\_course.SelectedItem + "','" + gender + "','" + this.dob.Text + "');", con);

sc.ExecuteNonQuery();

SqlCommand sc1 = new SqlCommand("insert into fee\_paid(paid,s\_name,s\_phone,course\_name)values(0,'" + txt\_sname.Text + "'," + txt\_sphone.Text + ",'" + combo\_course.SelectedItem + "');", con);

sc1.ExecuteNonQuery();

SqlCommand sp = new SqlCommand("update student set regno= CONCAT(CONCAT(RIGHT(year(doj),2)+'NCJ',+coursecode),s\_no) from student", con);

sp.ExecuteNonQuery();

con.Close();

}

MessageBox.Show("Student had been Admitted", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);

DataClear();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message,"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

con.Close();

}

}

private void but\_clear\_Click(object sender, EventArgs e)

{

DataClear();

}

private void but\_back\_Click(object sender, EventArgs e)

{

this.Close();

}

public static void main(string[] args)

{

Application.Run(new Create\_Student());

}

private void txt\_adhar\_TextChanged(object sender, EventArgs e)

{

if(txt\_adhar.TextLength < 12)

{

error\_mail.SetError(txt\_adhar, "Incorrect Aadhaar Number");

}

else

{

error\_mail.Clear();

}

}

}

}

**VIEW AND UPDATE STUDENT DETAILS**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Text.RegularExpressions;

namespace Fee\_Management

{

public partial class Edit\_Student : Form

{

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

public Edit\_Student()

{

InitializeComponent();

}

void DataClear()

{

cb\_reg.SelectedIndex = 0;

cb\_course.Text = " ---Select Course---";

txt\_sname.Clear();

txt\_address.Clear();

txt\_sphone.Clear();

txt\_email.Clear();

txt\_adhar.Clear();

txt\_fname.Clear();

txt\_mname.Clear();

txt\_pnumb.Clear();

txt\_doj.Clear();

error\_phone.Clear();

error\_email.Clear();

}

public void BCA()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'B'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

public void BCom()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'C'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

public void BSC()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'S'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

public void BA()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'A'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

private void cb\_course\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cb\_course.SelectedIndex == 0)

{

BCA();

}

else if (cb\_course.SelectedIndex == 1)

{

BCom();

}

else if (cb\_course.SelectedIndex == 2)

{

BSC();

}

else

{

BA();

}

DataClear();

}

private void but\_update\_Click(object sender, EventArgs e)

{

string pattern1 = "^([0-9a-zA-Z]([-\\.\\w]\*[0-9a-zA-Z])\*@([0-9a-zA-Z][-\\w]\*[0-9a-zA-Z]\\.)+[a-zA-Z]{2,9})$";

string pattern2 = (@"^((\+){0,1}91(\s){0,1}(\-){0,1}(\s){0,1}){0,1}(9|8|7)[0-9](\s){0,1}(\-){0,1}(\s){0,1}[1-9]{1}[0-9]{7}$");

if (!Regex.IsMatch(txt\_email.Text, pattern1))

{

error\_email.SetError(this.txt\_email, "Please check the mail id entered");

return;

}

if (!Regex.IsMatch(txt\_sphone.Text, pattern2))

{

error\_phone.SetError(this.txt\_sphone, "Check the phone number entered");

return;

}

if (!Regex.IsMatch(txt\_pnumb.Text, pattern2))

{

error\_phone.SetError(this.txt\_pnumb, "Check the phone number entered");

return;

}

if (String.IsNullOrEmpty(txt\_address.Text) || String.IsNullOrEmpty(txt\_email.Text) || String.IsNullOrEmpty(txt\_pnumb.Text) || String.IsNullOrEmpty(txt\_sphone.Text))

{

MessageBox.Show("Fields cannot be left blank", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

try

{

con.Open();

SqlCommand sc = new SqlCommand("update student set s\_address = '" + txt\_address.Text + "',s\_phone = " + txt\_sphone.Text + ",email\_id = '" + txt\_email.Text + "', p\_phone = " + txt\_pnumb.Text + " where regno = '" + cb\_reg.Text + "'", con);

sc.ExecuteNonQuery();

SqlCommand sc1 = new SqlCommand("update fee\_paid set s\_phone = " + txt\_sphone.Text + " where s\_name = '" + txt\_sname.Text + "'", con);

sc1.ExecuteNonQuery();

MessageBox.Show("Information Updated");

DataClear();

con.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

con.Close();

}

}

private void txt\_sphone\_TextChanged(object sender, EventArgs e)

{

string pattern = (@"^((\+){0,1}91(\s){0,1}(\-){0,1}(\s){0,1}){0,1}9|8|7[0-9](\s){0,1}(\-){0,1}(\s){0,1}[1-9]{1}[0-9]{7}$");

if (Regex.IsMatch(txt\_sphone.Text, pattern))

{

error\_phone.Clear();

}

else

{

error\_phone.SetError(this.txt\_sphone, "Check the phone number entered");

return;

}

}

private void txt\_pnumb\_TextChanged(object sender, EventArgs e)

{

string pattern = (@"^((\+){0,1}91(\s){0,1}(\-){0,1}(\s){0,1}){0,1}9|8|7[0-9](\s){0,1}(\-){0,1}(\s){0,1}[1-9]{1}[0-9]{7}$");

if (Regex.IsMatch(txt\_pnumb.Text, pattern))

{

error\_phone.Clear();

}

else

{

error\_phone.SetError(this.txt\_pnumb, "Check the phone number entered");

return;

}

}

private void txt\_email\_TextChanged(object sender, EventArgs e)

{

string pattern = "^([0-9a-zA-Z]([-\\.\\w]\*[0-9a-zA-Z])\*@([0-9a-zA-Z][-\\w]\*[0-9a-zA-Z]\\.)+[a-zA-Z]{2,9})$";

if (Regex.IsMatch(txt\_email.Text, pattern))

{

error\_email.Clear();

}

else

{

error\_email.SetError(this.txt\_email, "Please check the mail id entered");

return;

}

}

private void but\_clear\_Click(object sender, EventArgs e)

{

DataClear();

}

private void but\_back\_Click(object sender, EventArgs e)

{

this.Close();

}

private void cb\_reg\_SelectedIndexChanged(object sender, EventArgs e)

{

con.Open();

SqlCommand sc = new SqlCommand("select \* from student where regno ='" + cb\_reg.Text + "'", con);

SqlDataAdapter sda = new SqlDataAdapter(sc);

SqlDataReader dr = sc.ExecuteReader();

if (dr.Read())

{

txt\_sname.Text = (dr["s\_name"].ToString());

txt\_address.Text = (dr["s\_address"].ToString());

txt\_sphone.Text = (dr["s\_phone"].ToString());

txt\_email.Text = (dr["email\_id"].ToString());

txt\_adhar.Text = (dr["adhaar\_numb"].ToString());

txt\_fname.Text = (dr["f\_name"].ToString());

txt\_mname.Text = (dr["m\_name"].ToString());

txt\_pnumb.Text = (dr["p\_phone"].ToString());

txt\_doj.Text = (dr["doj"].ToString());

}

con.Close();

}

}

}

**PAY FEE**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Fee\_Management

{

public partial class Pay\_Fee : Form

{

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

public Pay\_Fee()

{

InitializeComponent();

}

void dataclear()

{

txt\_adm.Text = "";

txt\_lab.Text = "";

txt\_lib.Text = "";

txt\_pay.Clear();

txt\_reg.Text = "";

cb\_reg.SelectedIndex = 0;

txt\_sports.Text = "";

txt\_tut.Text = "";

cb\_course.SelectedIndex = -1;

txt\_tot.Text = "";

txt\_sname.Clear();

txt\_sphone.Clear();

}

public void BCA()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'B'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

public void BCom()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'C'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

public void BSC()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'S'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

public void BA()

{

DataRow dr;

SqlCommand cmd = new SqlCommand("Select \* from student where coursecode = 'A'", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_reg.DisplayMember = "regno";

cb\_reg.DataSource = dt;

}

private void but\_back\_Click(object sender, EventArgs e)

{

this.Close();

}

private void but\_clear\_Click(object sender, EventArgs e)

{

dataclear();

}

private void but\_Pay\_Click(object sender, EventArgs e)

{

try

{

con.Open();

SqlCommand sc = new SqlCommand("update fee\_paid set regno = '" + cb\_reg.Text + "',recipt\_no = next value for due\_seq, total\_fee = " + txt\_tot.Text + ",paid = paid + " + txt\_pay.Text + ", dop = getdate() where s\_phone = " + txt\_sphone.Text + "", con);

sc.ExecuteNonQuery();

SqlCommand ss = new SqlCommand("update fee\_paid set due= total\_fee-paid", con);

ss.ExecuteNonQuery();

MessageBox.Show("Payment Successfull", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);

con.Close();

dataclear();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

con.Close();

}

}

private void cb\_course\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cb\_course.SelectedIndex == 0)

{

BCA();

con.Open();

SqlCommand sc1 = new SqlCommand("select \* from fees where sl\_no = 1", con);

sc1.ExecuteNonQuery();

SqlDataReader dt1 = sc1.ExecuteReader();

if (dt1.Read())

{

txt\_adm.Text = (dt1["admission\_fee"].ToString());

txt\_reg.Text = (dt1["registration\_fee"].ToString());

txt\_tut.Text = (dt1["tuition\_fee"].ToString());

txt\_lab.Text = (dt1["lab\_fee"].ToString());

txt\_sports.Text = (dt1["sports\_fee"].ToString());

txt\_lib.Text = (dt1["library\_fee"].ToString());

txt\_tot.Text = (dt1["total\_fee"].ToString());

}

con.Close();

}

else if (cb\_course.SelectedIndex == 1)

{

BCom();

con.Open();

SqlCommand sc2 = new SqlCommand("select \* from fees where sl\_no = 2", con);

sc2.ExecuteNonQuery();

SqlDataReader dt2 = sc2.ExecuteReader();

if (dt2.Read())

{

txt\_adm.Text = (dt2["admission\_fee"].ToString());

txt\_reg.Text = (dt2["registration\_fee"].ToString());

txt\_tut.Text = (dt2["tuition\_fee"].ToString());

txt\_lab.Text = (dt2["lab\_fee"].ToString());

txt\_sports.Text = (dt2["sports\_fee"].ToString());

txt\_lib.Text = (dt2["library\_fee"].ToString());

txt\_tot.Text = (dt2["total\_fee"].ToString());

}

con.Close();

}

else if (cb\_course.SelectedIndex == 2)

{

BSC();

con.Open();

SqlCommand sc3 = new SqlCommand("select \* from fees where sl\_no = 3", con);

sc3.ExecuteNonQuery();

SqlDataReader dt3 = sc3.ExecuteReader();

if (dt3.Read())

{

txt\_adm.Text = (dt3["admission\_fee"].ToString());

txt\_reg.Text = (dt3["registration\_fee"].ToString());

txt\_tut.Text = (dt3["tuition\_fee"].ToString());

txt\_lab.Text = (dt3["lab\_fee"].ToString());

txt\_sports.Text = (dt3["sports\_fee"].ToString());

txt\_lib.Text = (dt3["library\_fee"].ToString());

txt\_tot.Text = (dt3["total\_fee"].ToString());

}

con.Close();

}

else if (cb\_course.SelectedIndex == 3)

{

BA();

con.Open();

SqlCommand sc4 = new SqlCommand("select \* from fees where sl\_no = 4", con);

sc4.ExecuteNonQuery();

SqlDataReader dt4 = sc4.ExecuteReader();

if (dt4.Read())

{

txt\_adm.Text = (dt4["admission\_fee"].ToString());

txt\_reg.Text = (dt4["registration\_fee"].ToString());

txt\_tut.Text = (dt4["tuition\_fee"].ToString());

txt\_lab.Text = (dt4["lab\_fee"].ToString());

txt\_sports.Text = (dt4["sports\_fee"].ToString());

txt\_lib.Text = (dt4["library\_fee"].ToString());

txt\_tot.Text = (dt4["total\_fee"].ToString());

}

con.Close();

}

else

{

con.Close();

dataclear();

}

}

private void cb\_reg\_SelectedIndexChanged(object sender, EventArgs e)

{

con.Open();

SqlCommand sc = new SqlCommand("select \* from student where regno = '" + cb\_reg.Text + "'", con);

sc.ExecuteNonQuery();

SqlDataReader dr = sc.ExecuteReader();

if(dr.Read())

{

txt\_sname.Text = (dr["s\_name"].ToString());

cb\_course.Text = (dr["course\_name"].ToString());

txt\_sphone.Text = (dr["s\_phone"].ToString());

}

con.Close();

}

}

}

**VIEW RECIEPT**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Fee\_Management

{

public partial class Acknodwledge : Form

{

public Acknodwledge()

{

InitializeComponent();

regno();

}

public void regno()

{

DataRow dr;

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

SqlCommand cmd = new SqlCommand("Select \* from fee\_paid", con);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dr = dt.NewRow();

dr.ItemArray = new object[] { "---Select Register No---" };

dt.Rows.InsertAt(dr, 0);

cb\_regno.DisplayMember = "regno";

cb\_regno.DataSource = dt;

}

private void cb\_regno\_SelectedIndexChanged(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

con.Open();

SqlCommand sc = new SqlCommand("select \* from fee\_paid where regno = '" + cb\_regno.Text + "'", con);

sc.ExecuteNonQuery();

SqlDataReader dr = sc.ExecuteReader();

while (dr.Read())

{

lab\_course.Text = (dr["course\_name"].ToString());

lab\_name.Text = (dr["s\_name"].ToString());

lab\_recno.Text = (dr["recipt\_no"].ToString());

lab\_date.Text = (dr["dop"].ToString());

lab\_paid.Text = (dr["paid"].ToString());

lab\_due.Text = (dr["due"].ToString());

}

con.Close();

con.Open();

if (lab\_course.Text == "BCA")

{

SqlCommand sc1 = new SqlCommand("select \* from fees where sl\_no = 1", con);

sc1.ExecuteNonQuery();

SqlDataReader dt1 = sc1.ExecuteReader();

while (dt1.Read())

{

lab\_admission.Text = (dt1["admission\_fee"].ToString());

lab\_regis.Text = (dt1["registration\_fee"].ToString());

lab\_tut.Text = (dt1["tuition\_fee"].ToString());

lab\_lab.Text = (dt1["lab\_fee"].ToString());

lab\_sports.Text = (dt1["sports\_fee"].ToString());

lab\_lib.Text = (dt1["library\_fee"].ToString());

lab\_total.Text = (dt1["total\_fee"].ToString());

}

}

else if (lab\_course.Text == "BCom")

{

SqlCommand sc2 = new SqlCommand("select \* from fees where sl\_no = 2", con);

sc2.ExecuteNonQuery();

SqlDataReader dt2 = sc2.ExecuteReader();

while (dt2.Read())

{

lab\_admission.Text = (dt2["admission\_fee"].ToString());

lab\_regis.Text = (dt2["registration\_fee"].ToString());

lab\_tut.Text = (dt2["tuition\_fee"].ToString());

lab\_lab.Text = (dt2["lab\_fee"].ToString());

lab\_sports.Text = (dt2["sports\_fee"].ToString());

lab\_lib.Text = (dt2["library\_fee"].ToString());

lab\_total.Text = (dt2["total\_fee"].ToString());

}

}

else if (lab\_course.Text == "BSC")

{

SqlCommand sc3 = new SqlCommand("select \* from fees where sl\_no = 3", con);

sc3.ExecuteNonQuery();

SqlDataReader dt3 = sc3.ExecuteReader();

while (dt3.Read())

{

lab\_admission.Text = (dt3["admission\_fee"].ToString());

lab\_regis.Text = (dt3["registration\_fee"].ToString());

lab\_tut.Text = (dt3["tuition\_fee"].ToString());

lab\_lab.Text = (dt3["lab\_fee"].ToString());

lab\_sports.Text = (dt3["sports\_fee"].ToString());

lab\_lib.Text = (dt3["library\_fee"].ToString());

lab\_total.Text = (dt3["total\_fee"].ToString());

}

}

else

{

SqlCommand sc4 = new SqlCommand("select \* from fees where sl\_no = 4", con);

sc4.ExecuteNonQuery();

SqlDataReader dt4 = sc4.ExecuteReader();

while (dt4.Read())

{

lab\_admission.Text = (dt4["admission\_fee"].ToString());

lab\_regis.Text = (dt4["registration\_fee"].ToString());

lab\_tut.Text = (dt4["tuition\_fee"].ToString());

lab\_lab.Text = (dt4["lab\_fee"].ToString());

lab\_sports.Text = (dt4["sports\_fee"].ToString());

lab\_lib.Text = (dt4["library\_fee"].ToString());

lab\_total.Text = (dt4["total\_fee"].ToString());

}

}

}

private void but\_print\_Click(object sender, EventArgs e)

{

PrintDialog pd = new PrintDialog();

pd.ShowDialog();

}

}

}

**VIEW DUE FEE**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Fee\_Management

{

public partial class View\_Due : Form

{

SqlConnection con = new SqlConnection(@"Data Source=SAMARTHA-LS\SQLEXPRESS;Initial Catalog=fee;Integrated Security=True;");

public View\_Due()

{

InitializeComponent();

view();

}

public void view()

{

SqlDataAdapter sda = new SqlDataAdapter("select regno,s\_name,paid,due from fee\_paid where due >0", con);

DataTable dt = new DataTable();

sda.Fill(dt);

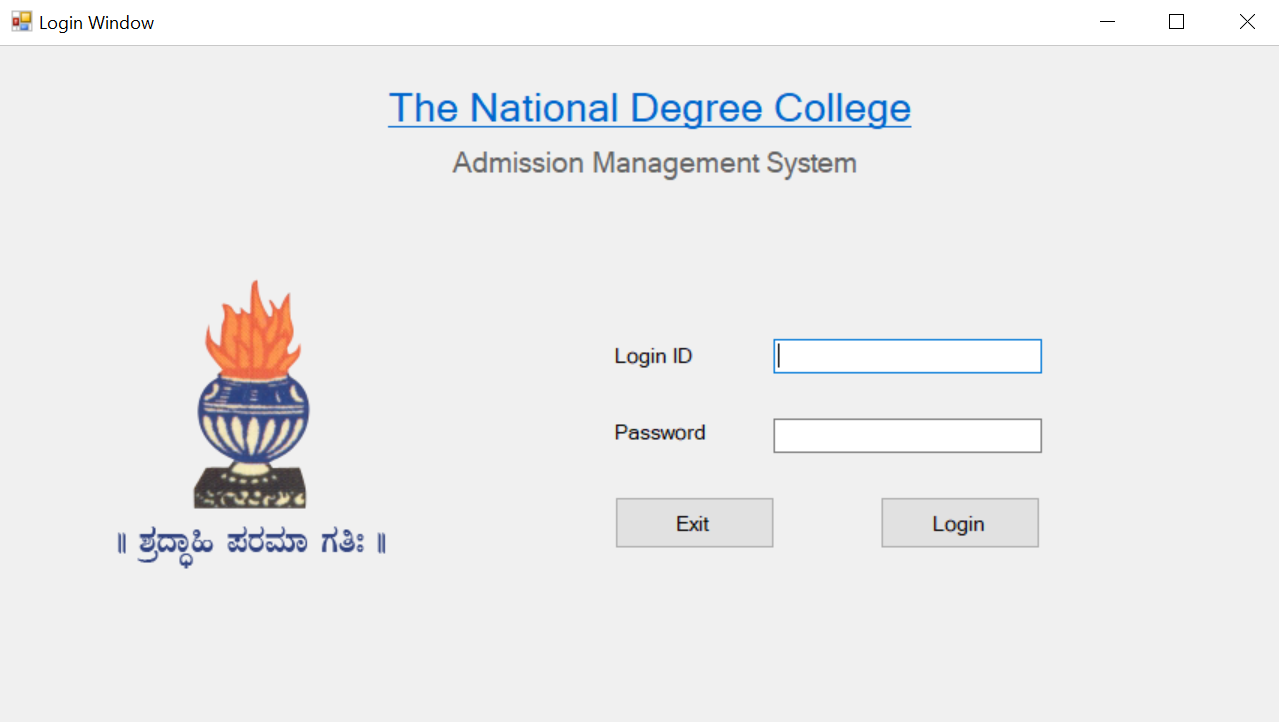
dgv\_due.DataSource = dt;

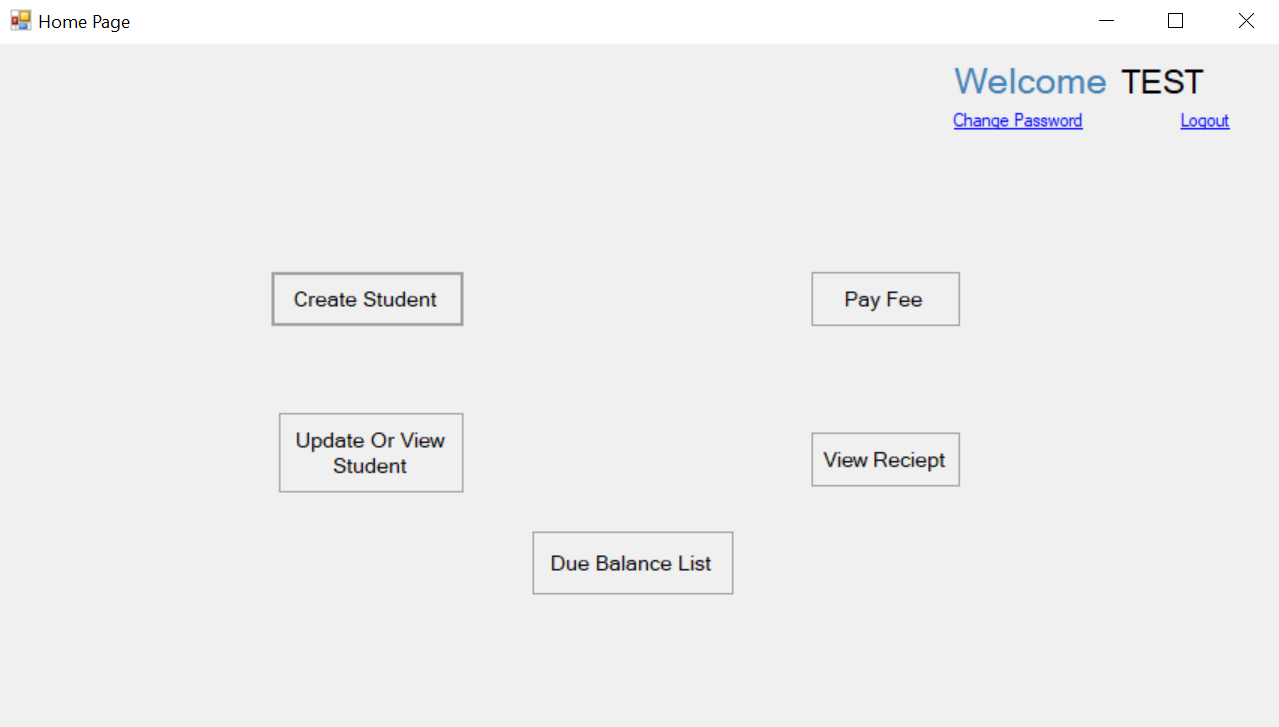
}

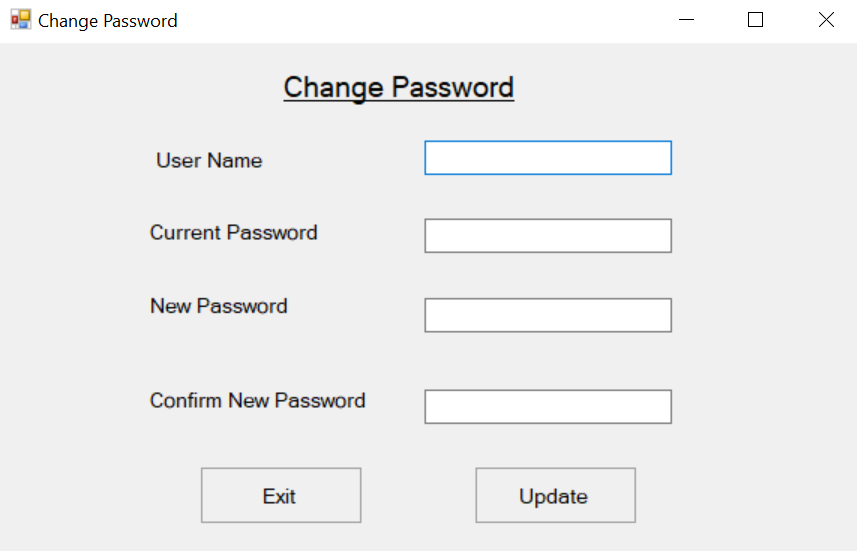
}

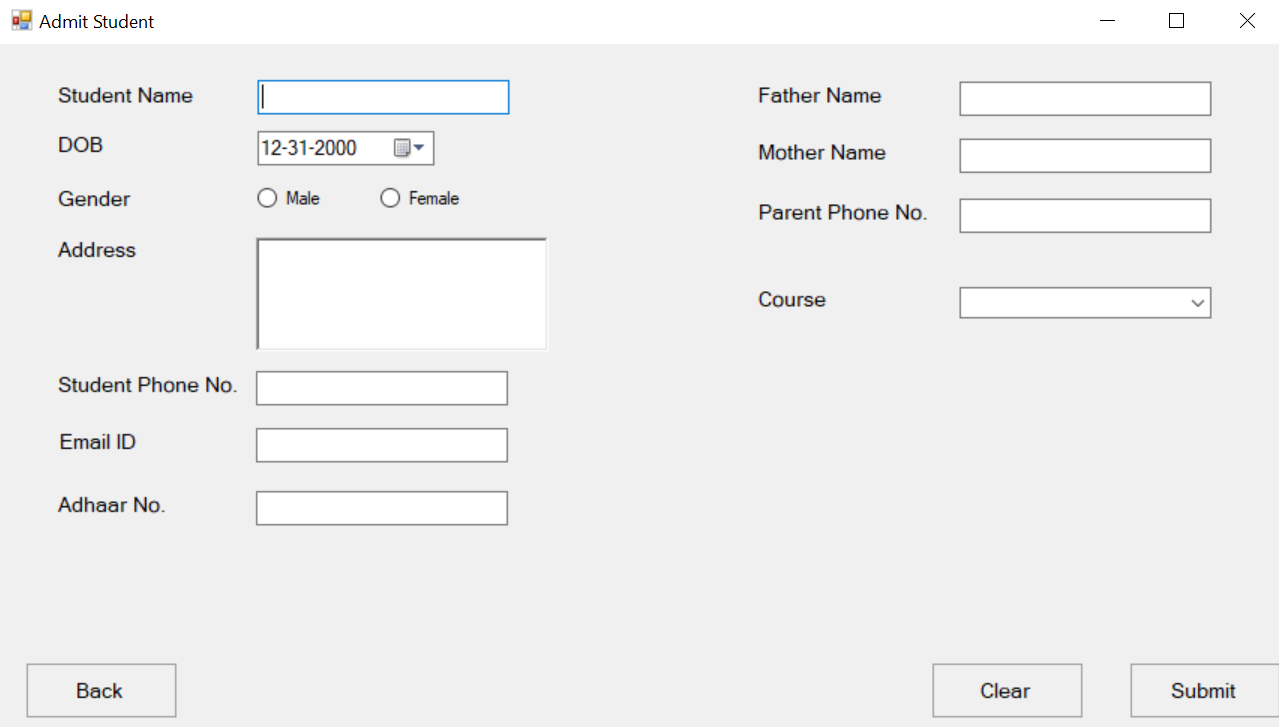
}

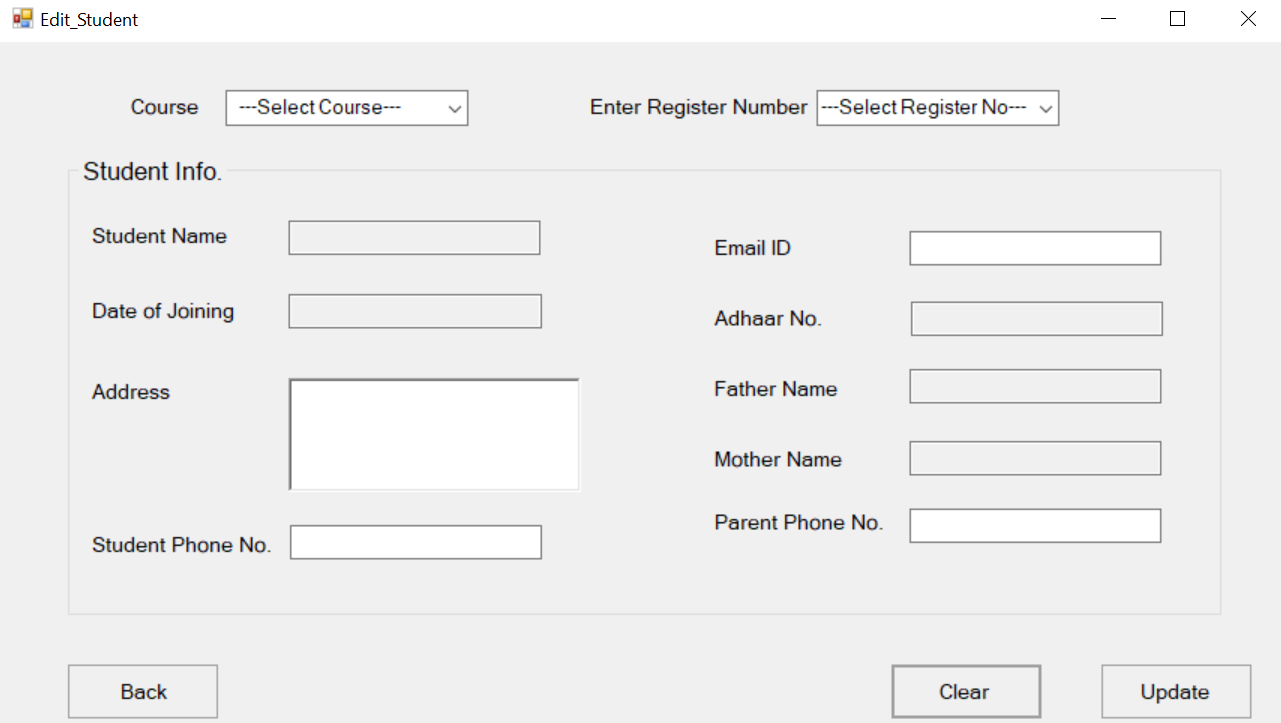
**SCREEN SHOTS**

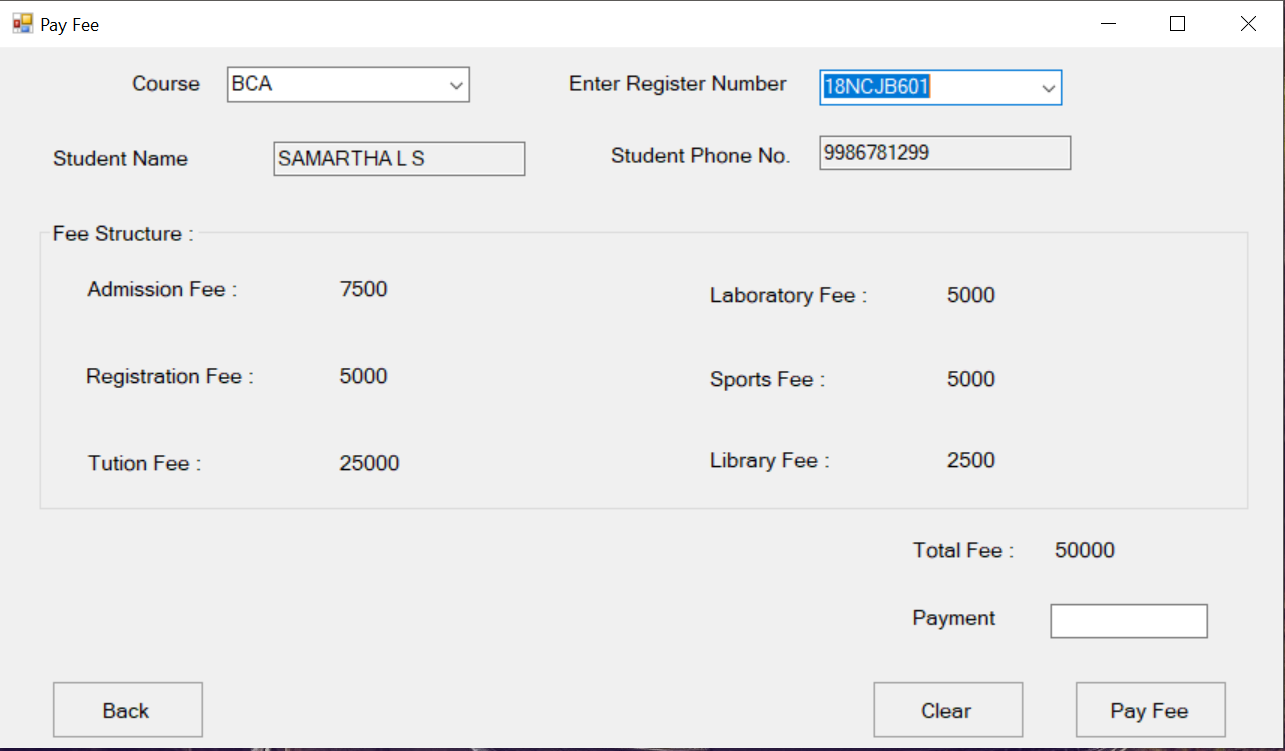


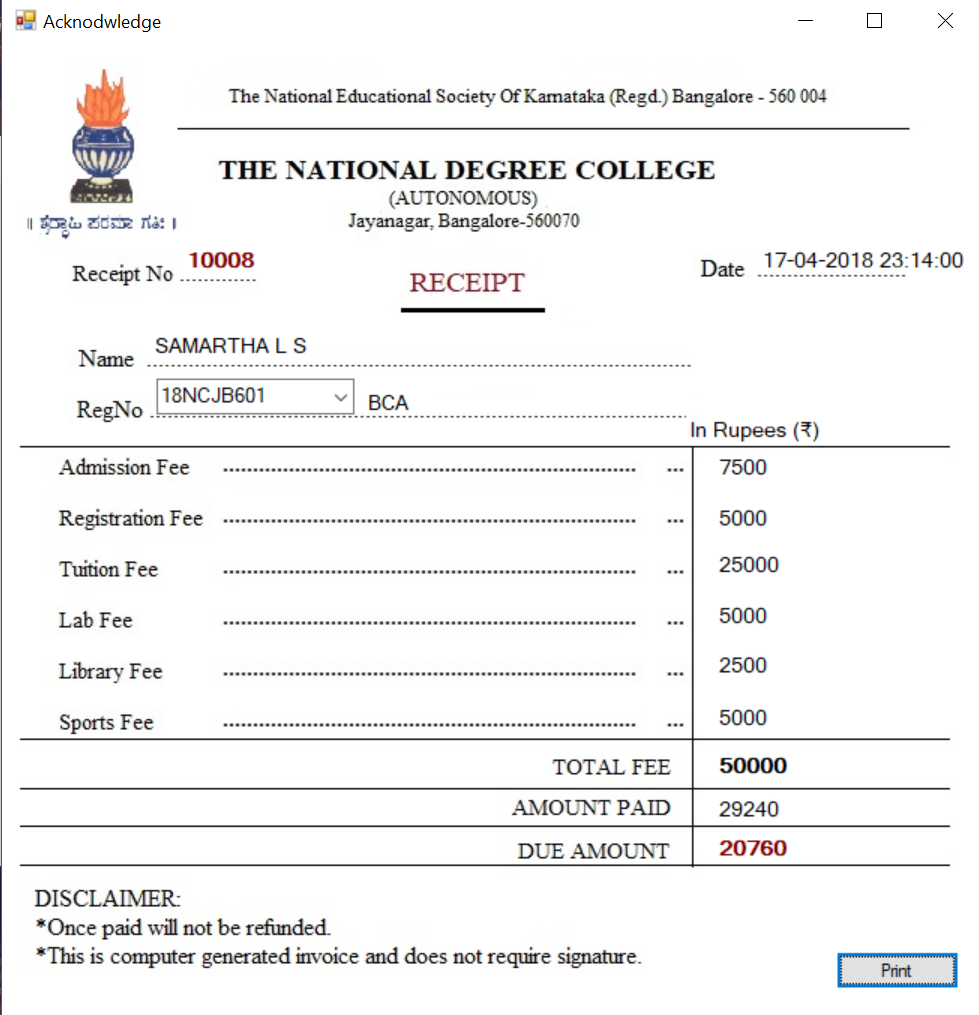


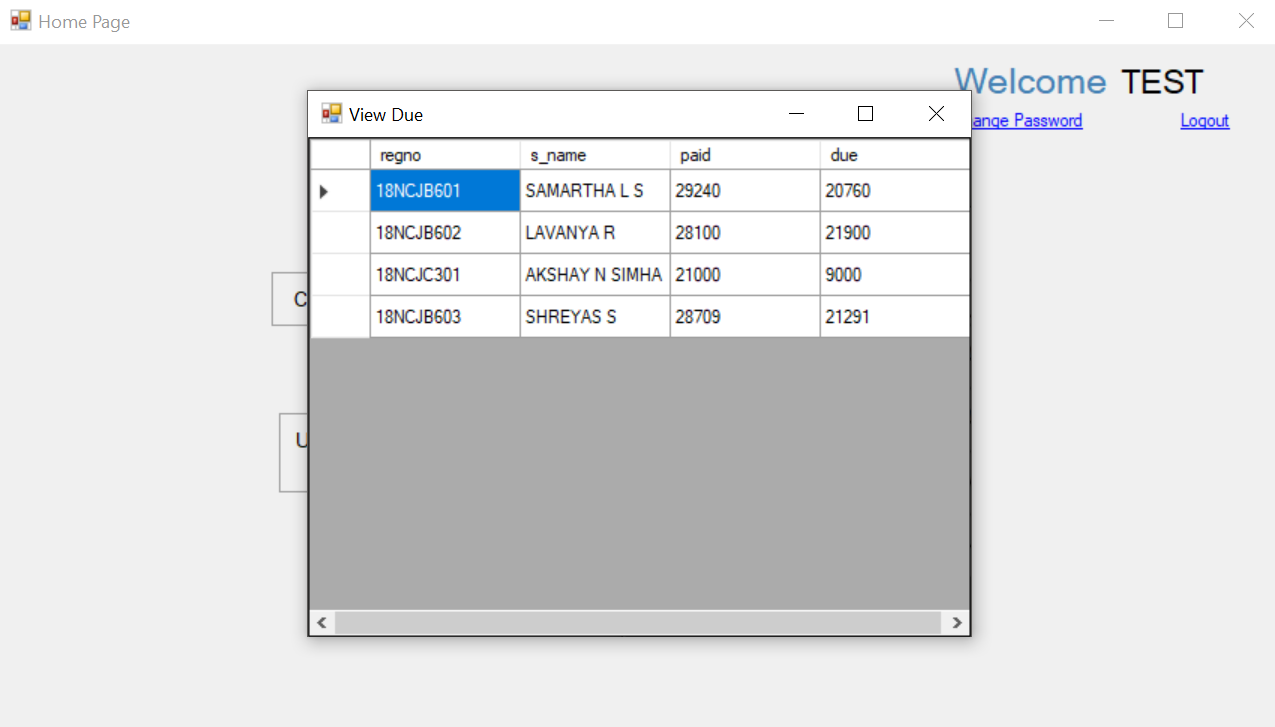












**BIBILOGRAPHY**

**REVIEW THROUGH BOOKS**

* Database Management Systems - Navathe.
* C# complete reference Balguru swamy.

**REVIEW THROUGH WEB REFERENCE**

* [***www.sitetpoint.com***](http://www.sitetpoint.com)
* [***www.w3schools.com***](http://www.w3schools.com)
* [***www.wikipedia.com***](http://www.wikipedia.com)
* [***www.tutorialspoint.com***](http://www.tutorialspoint.com)
* [***www.youtube.com***](http://www.youtube.com)