Informatics College Pokhara

informatics
college pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: **Sneha Gurung**
London Met ID: Enter ID Here

Submitted To: Ishwor Sapkota
Module Leader

Marking Scheme

Component Grade and Comments	
A. Implementation of Application	
User Interface and proper controls used for designing	missing controls in the interface
Manual data entry or import from csv	not properly saved or imported data
Data Validation	Only basic validation
Enrollment Report & weekly report in tabular format	very poorly executed reports and data not shown accurately
Course wise enrollment report & Chart display	Very poorly designed and only contains one report format with in appropriate data
Algorithm used for sorting & proper sorting of data	Default sorting provided by .net is used
B. Documentation	
User Manual for running the application	User Manual is below average. Is textual only.

Application architecture & description of the classes ad methods sued	average work with very limited explanation of the classes and methods used
Flow chart, algoriathms and data sctructures used	average work with very limited explanation and missing diagramatic representation.
Reflective essay	Very poorly written
C. Programming Style	
Clarity of code,Popper Naming convention & comments	Very poor code
System Usability	unusable system

Overall Grade:	**D+**	**D+**

Overall Comment:

Code should be self explainable with less comments. Need some proper naming of the componen and require to add comments on required area.

Good can explain the code.

Informatics College Pokhara

<u>Application Development</u>

<u>CS6004NP</u>

Coursework 1

Submitted By:		**Submitted To:**
Student Name:	Sneha Gurung	Mr. Ishwor Sapkota
London Met ID:	17031948	
Group:	L3C2	
Date:	10-Jan-2019	

Table of Contents

1. Introduction 1

1.1 Current Scenario 1

1.2 Proposed System 1

2. User Manual 2

3. Journal Article 14

4. System Architecture 16

4.1 Architectural Diagram 16

4.2 Class Diagram 17

4.3 Class Description 17

4.4 Flowchart 20

5. Sorting 22

6. Testing 24

Test case 1 24

Test case 2 24

Test case 3 25

Test case 4 26

Test case 5 27

7. Reflection 28

Bibliography 29

Appendix 30

List of Figure

Figure 1: Login fail 2
Figure 2: Login successful 3
Figure 3: Main form 4
Figure 4: Add record fail 5
Figure 5: Inserting all details 6
Figure 6: Add record successful 6
Figure 7: Retrieve record 7
Figure 8: sort by name 8
Figure 9: sort by date 8
Figure 10: importing excel file 9
Figure 11: Excel file 10
Figure 12: Excel data added manually 10
Figure 13: Course Enrol 11
Figure 14: Pie chart 11
Figure 15: clear button for retrieve record 12
Figure 16: clear button for course enrol 12
Figure 17: Architectural Diagram 16
Figure 18: Test case 1 24
Figure 19: Test case 2 25
Figure 20: Test case 3 25
Figure 21: Test case sort by name 26
Figure 22: Test case sort by date 26
Figure 23: Test case 5 27

List of Table

Table 1: Login 17
Table 2: Main Window 18
Table 3: Add record 18
Table 4: Retrieve record 19
Table 5: Generate Report 19

1. Introduction

This coursework is assigned under our module Application Development to address a Window Presentation Foundation Application for Student Information System. This application is to keep track of the student’s details, program enrol and registration date. This project is based on the concept of managing student’s information where user input records by providing student’s personal information like student ID, name, address, email, contact number and their course and making it available to retrieve student reports and enrol status. Likewise, it enables user to view the student details sorted by the first name of the student and another sorted by their registration date. In addition to this it also includes additional feature that displays weekly tabular report and chart showing total number of students enrolled on each program.

1.1 Current Scenario

If we examine the modern-day scenario there are numerous schools who keep report of their information in vintage traditional system which can be frequently paper based system. These devices are often slower and incontinent to perform and maintain. Especially in Nepal there is no systematic computerised system for maintaining student details.

1.2 Proposed System

The proposed system is a digital version of Student Management System which does all the task the tradition system along with some additional benefits of high security, easy to operate & maintain & reliable. Some of the advantages of digital student management system over the traditional system are given below:

- Eco-Friendly: paperwork can be avoided
- Efficient control over student data
- Monitor student performance
- Supervise multiple branches
- Cost-efficient and User-friendly
- Single solution for total College management
- Easy access to forums, attendance, timetable, marks, grades and examination schedule

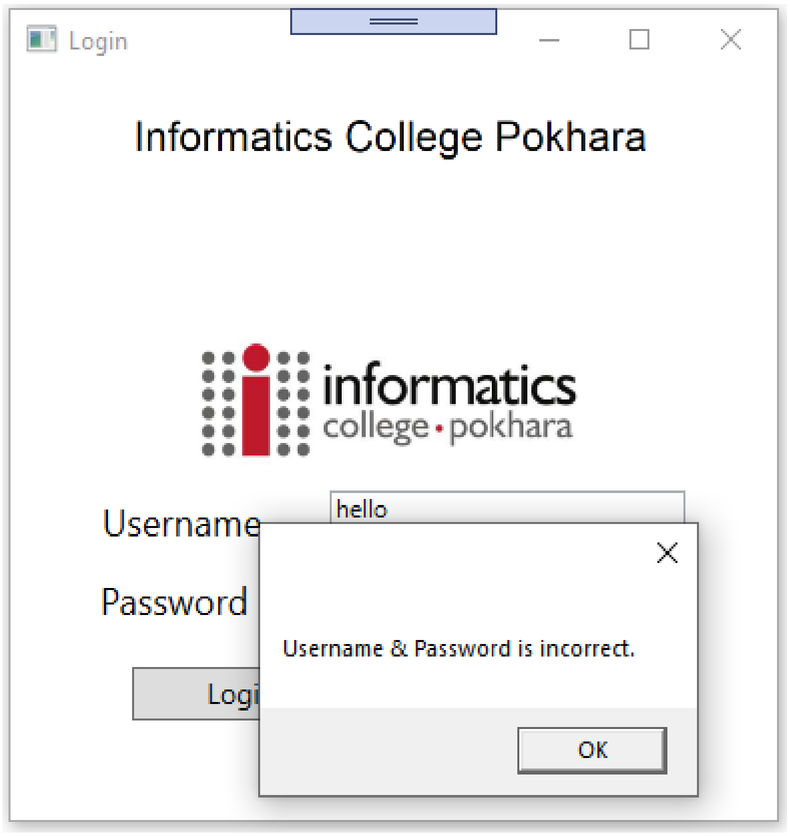
2. User Manual

The detailed information to run the program along with proper screenshot is as below:

When running the program

• Click the 'ApplicationDevelopmentCW1.sln'

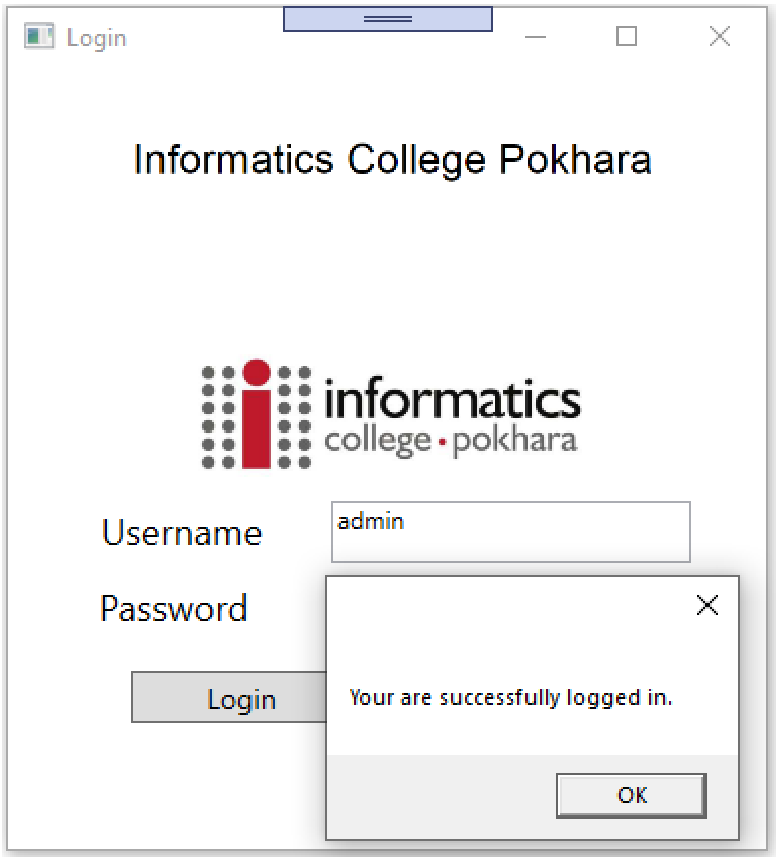
Log in



Username & Password is incorrect.

Figure 1: Login fail

- The username and password of this system is admin and admin respectively.
- As the end user operates the system the initial screen will be secure.
- Only valid username and password can provide access to the system and user can continue the next procedure.
- If the user enters an invalid username or password, thus system displays a pop-up message.
- User can click the ok button to try again or terminate the system.

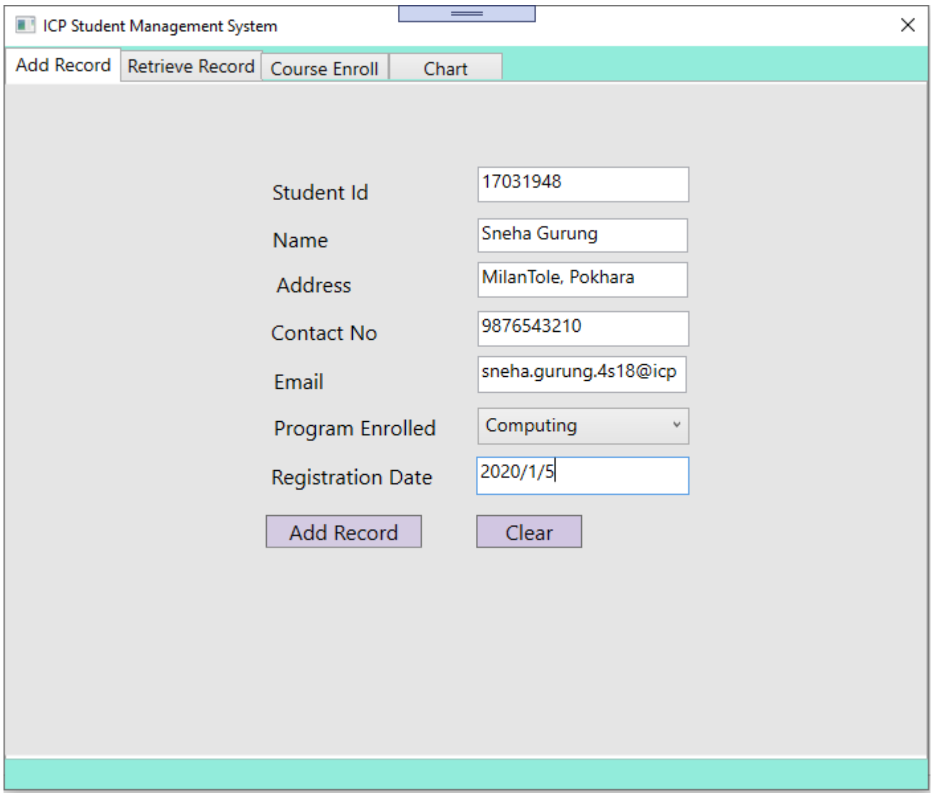


Your are successfully logged in.

Figure 2: Login successful

- When the user enters correct username and password which is admin and admin respectively, a popup message box is shown.
- User can click the ok button to continue the procedure. When the user clicks the ok button, the main form will show up and the login form will close itself.

Main Form



Student Id	17031948
Name	Sneha Gurung
Address	MilanTole, Pokhara
Contact No	9876543210
Email	sneha.gurung.4s18@icp
Program Enrolled	Computing
Registration Date	2020/1/5

Figure 3: Main form

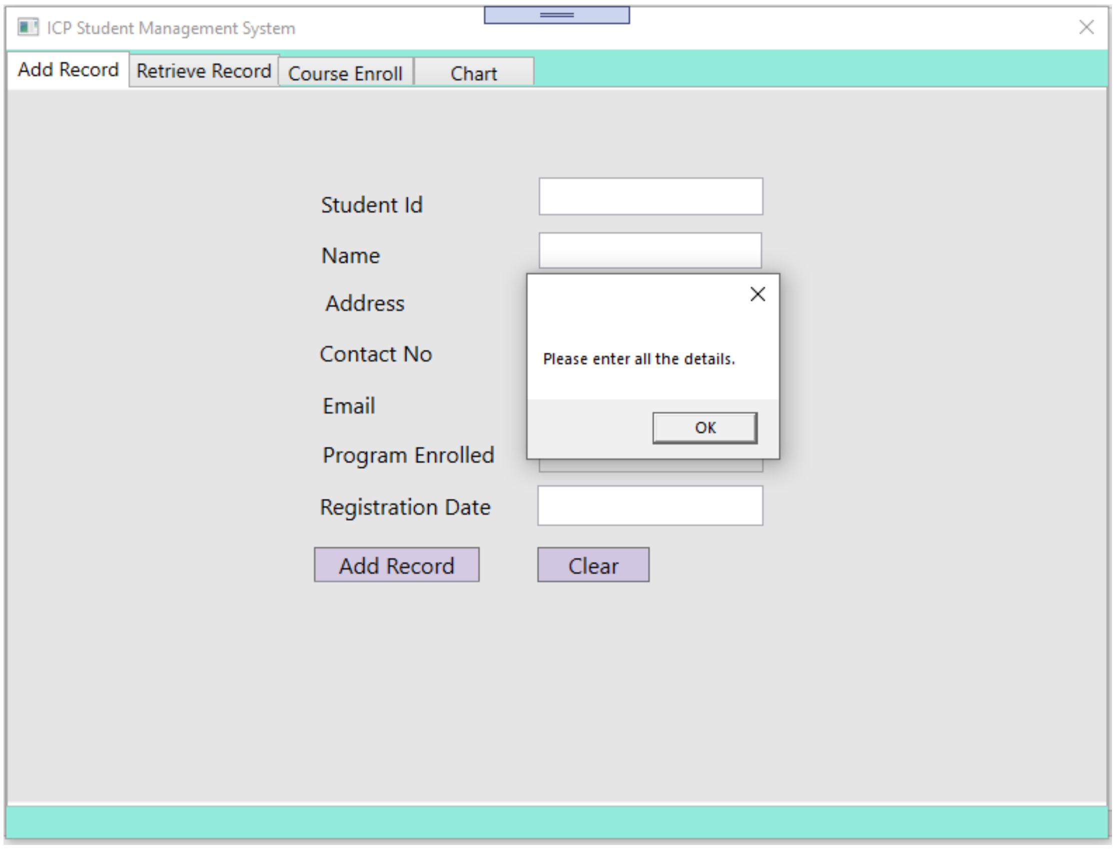
After logging into the system with correct credentials, the main screen of the system appears. The main screen provides four options in menu bar to click for which are Add Record, Retrieve Record, Course Enrol and Chart. The menus perform different tasks which are described as follows:

- Add Record: The Add Record screen allows user to insert a new student data verification before inserting student's information like student's ID, name, address, contact number, email address, course enrol and registration date. All the information is recorded by clicking the add to record button. User is accrued to fill all the information otherwise it not added to the record.
- Retrieve Record: The Retrieve Record contains a DataGrid view populated with all the student's information. All the user input records like student ID, name, address, email, contact number and their course are available and user can retrieve student reports from the DataGrid. Likewise, it enables user to view the student details sorted by the first

name of the student and another sorted by their registration date. User can also import a record from CSV to allow manually inputting details.

- Course Enrol: Course Enrol lets user to generate report of course enrol details. It calculates the number of students enrolled in a specific course or program. It displays weekly tabular report showing total number of students enrolled so far in each program.
- Chart: Chart includes a static form canvas of pie chart showing total number of students on each program computing, multimedia, networking and database.

Add Record

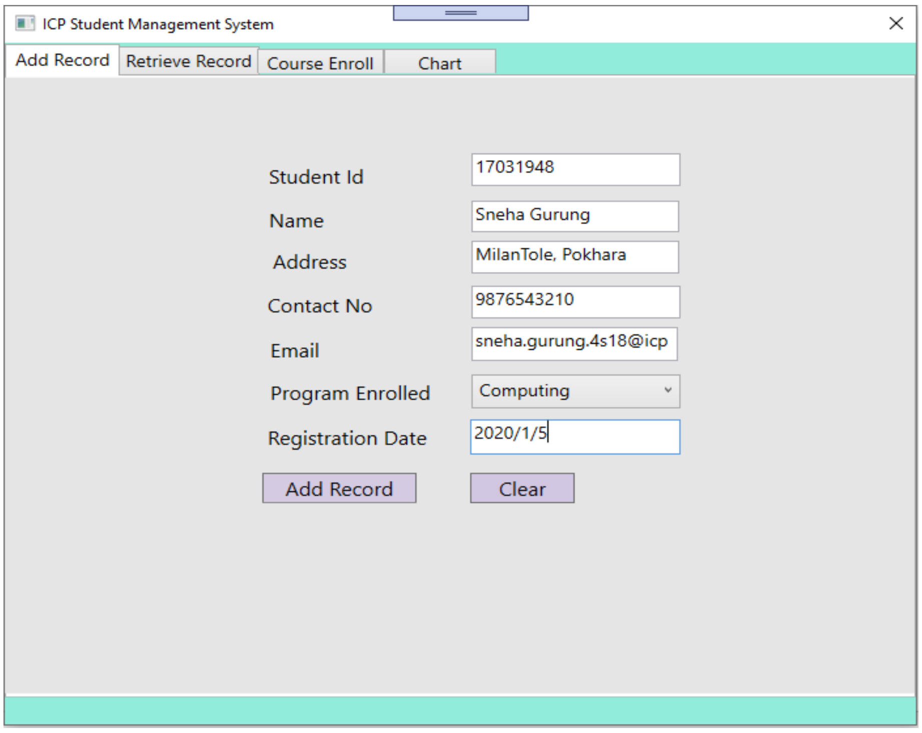


Please enter all the details.

Figure 4: Add record fail

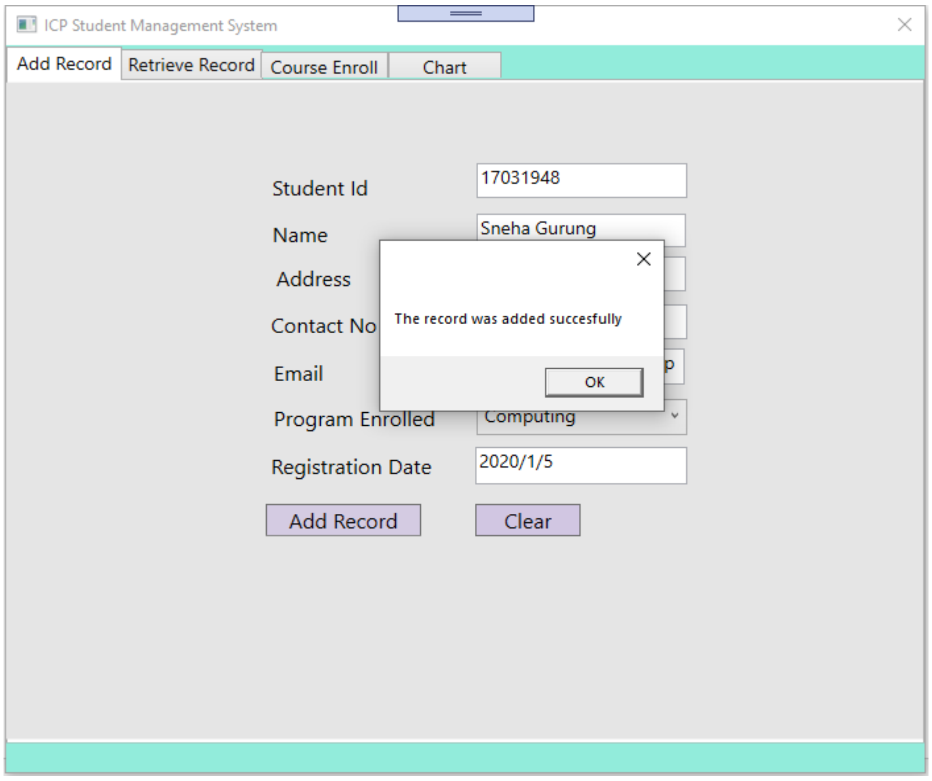
- Add Record screen allows user to insert a new student data verification before inserting student's information like student's ID, name, address, contact number, email address, course enrol and registration date. All the information is recorded by clicking the add to record button.

- User must fill all the required details to record the information. If not, a message box is displayed. User can click the ok button to fill all the required information again.



Student Id	17031948
Name	Sneha Gurung
Address	MilanTole, Pokhara
Contact No	9876543210
Email	sneha.gurung.4s18@icp
Program Enrolled	Computing
Registration Date	2020/1/5

Figure 5: Inserting all details



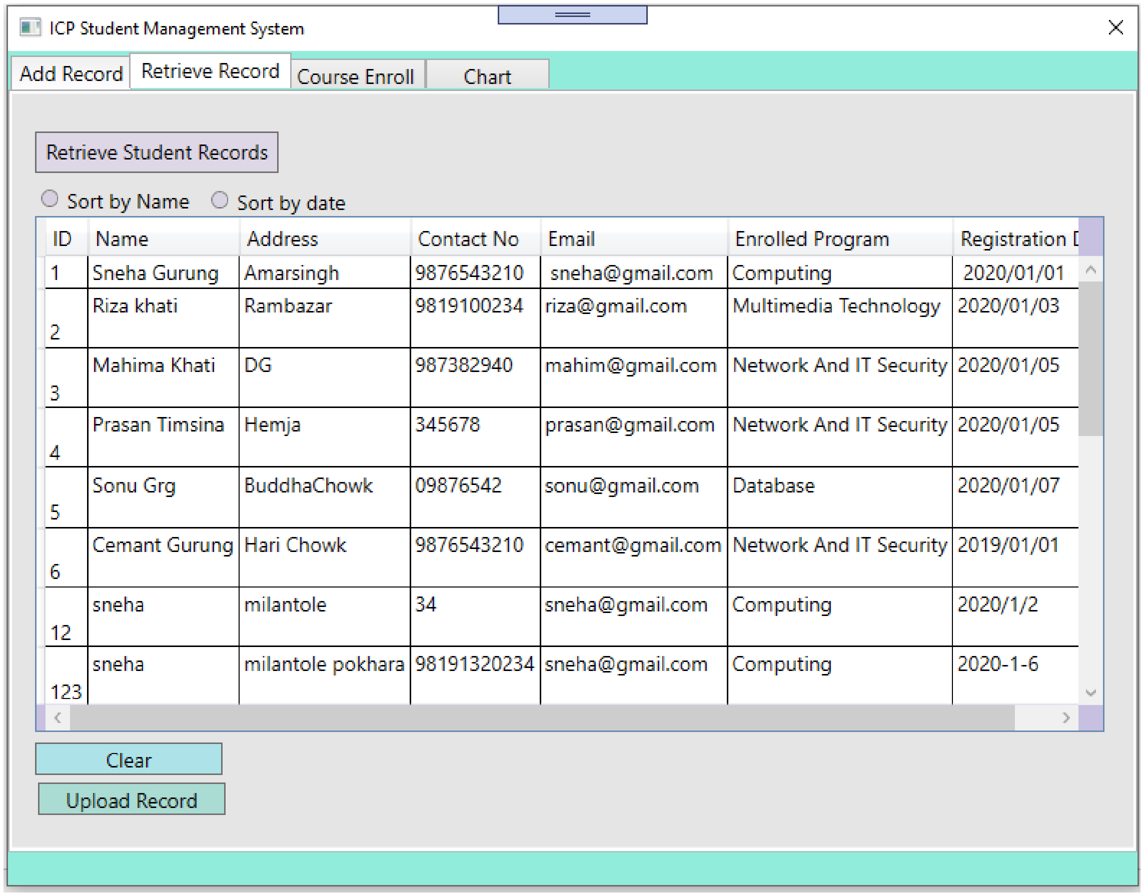
Student Id	17031948
Name	Sneha Gurung
Address	
Contact No	
Email	
Program Enrolled	Computing
Registration Date	2020/1/5

The record was added succesfully

Figure 6: Add record successful

- After providing all the required input parameters, user can simply click the Add Record button to save all the records.
- A popup message box is displayed, user can click the ok button to terminate the message box.

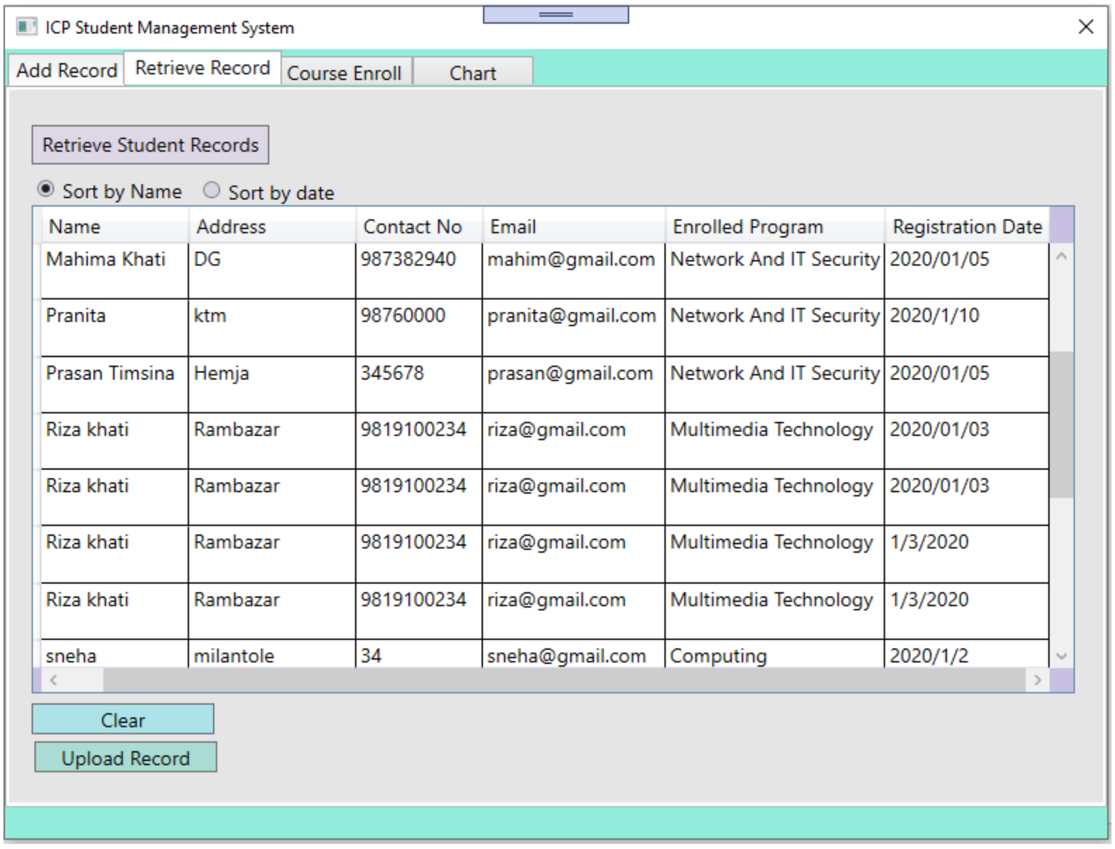
Retrieve Record



ID	Name	Address	Contact No	Email	Enrolled Program	Registration D
1	Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
2	Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	2020/01/03
3	Mahima Khati	DG	987382940	mahim@gmail.com	Network And IT Security	2020/01/05
4	Prasan Timsina	Hemja	345678	prasan@gmail.com	Network And IT Security	2020/01/05
5	Sonu Grg	BuddhaChowk	09876542	sonu@gmail.com	Database	2020/01/07
6	Cemant Gurung	Hari Chowk	9876543210	cemant@gmail.com	Network And IT Security	2019/01/01
12	sneha	milantole	34	sneha@gmail.com	Computing	2020/1/2
123	sneha	milantole pokhara	98191320234	sneha@gmail.com	Computing	2020-1-6

Figure 7: Retrieve record

- Retrieve Record contains a DataGrid view populated with all the student's information. All the user input records like student ID, name, address, email, contact number and their course are available and user can retrieve student reports from the DataGrid.
- User can simply click the Retrieve Student Records to see all the records of student's information.



ICP Student Management System

Add Record | Retrieve Record | Course Enroll | Chart

Retrieve Student Records

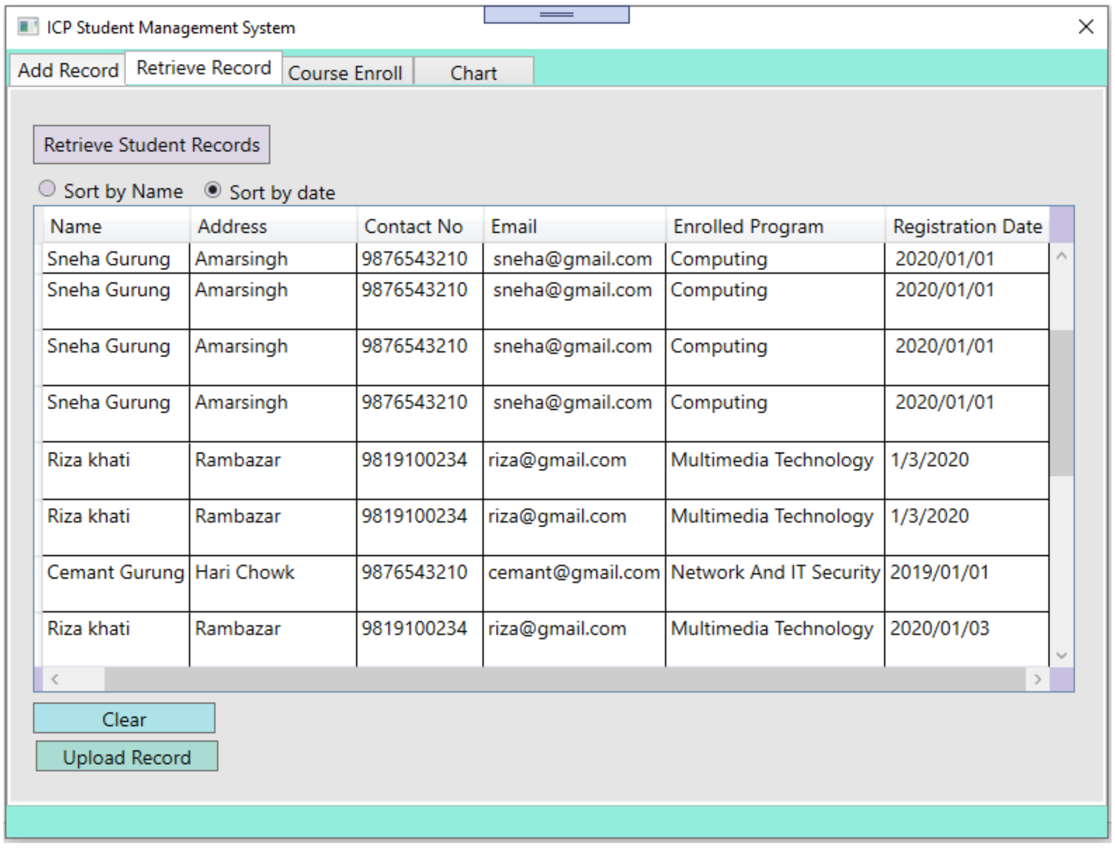
[✓ Sort by Name] Sort by date

Name	Address	Contact No	Email	Enrolled Program	Registration Date
Mahima Khati	DG	987382940	mahim@gmail.com	Network And IT Security	2020/01/05
Pranita	ktm	98760000	pranita@gmail.com	Network And IT Security	2020/1/10
Prasan Timsina	Hemja	345678	prasan@gmail.com	Network And IT Security	2020/01/05
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	2020/01/03
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	2020/01/03
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
sneha	milantole	34	sneha@gmail.com	Computing	2020/1/2

Clear

Upload Record

Figure 8: sort by name



ICP Student Management System

Add Record | Retrieve Record | Course Enroll | Chart

Retrieve Student Records

Sort by Name [✓ Sort by date]

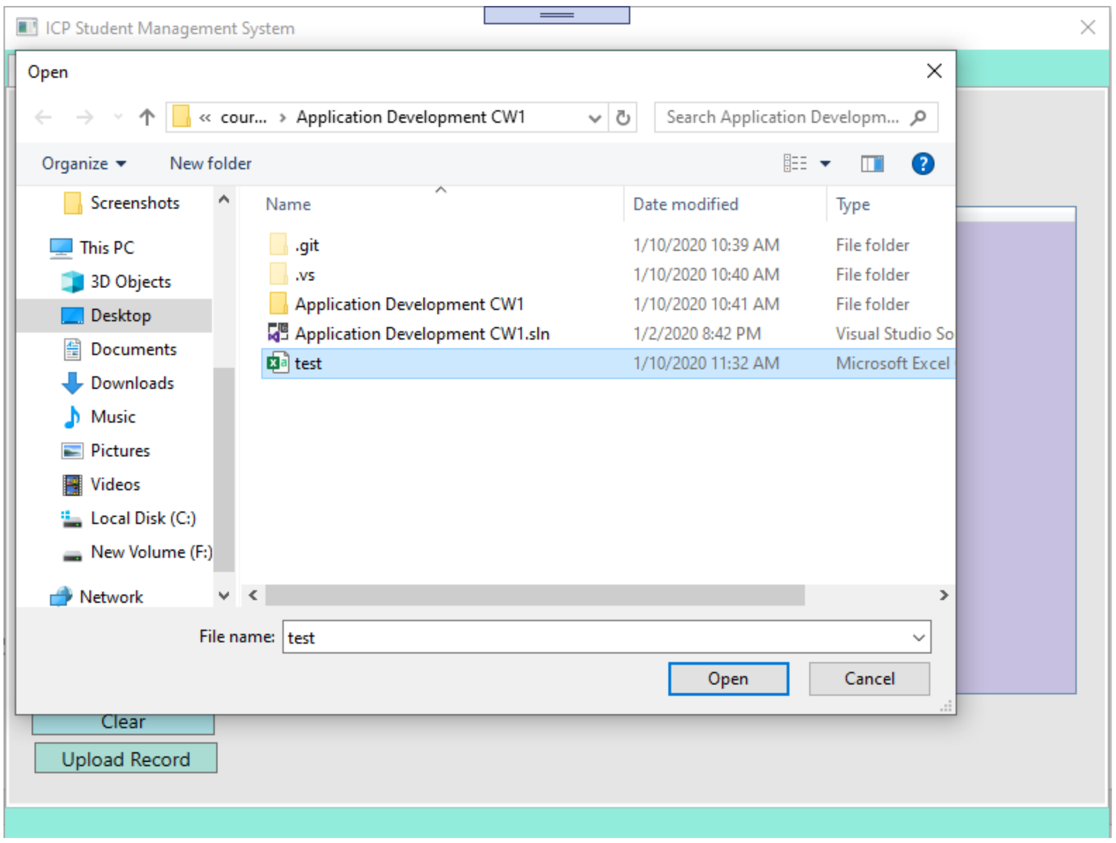
Name	Address	Contact No	Email	Enrolled Program	Registration Date
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
Cemant Gurung	Hari Chowk	9876543210	cemant@gmail.com	Network And IT Security	2019/01/01
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	2020/01/03

Clear

Upload Record

Figure 9: sort by date

- As all the user input records are available, user can retrieve student reports from the DataGrid sorted by the first name of the student and another sorted by their registration date.
- To display the record sorted by the first name of the student, user is required to click the radio button Sort by name before clicking the button Retrieve Student Record.
- User is required to click the radio button Sort by date to display the records sorted by registration date of the student

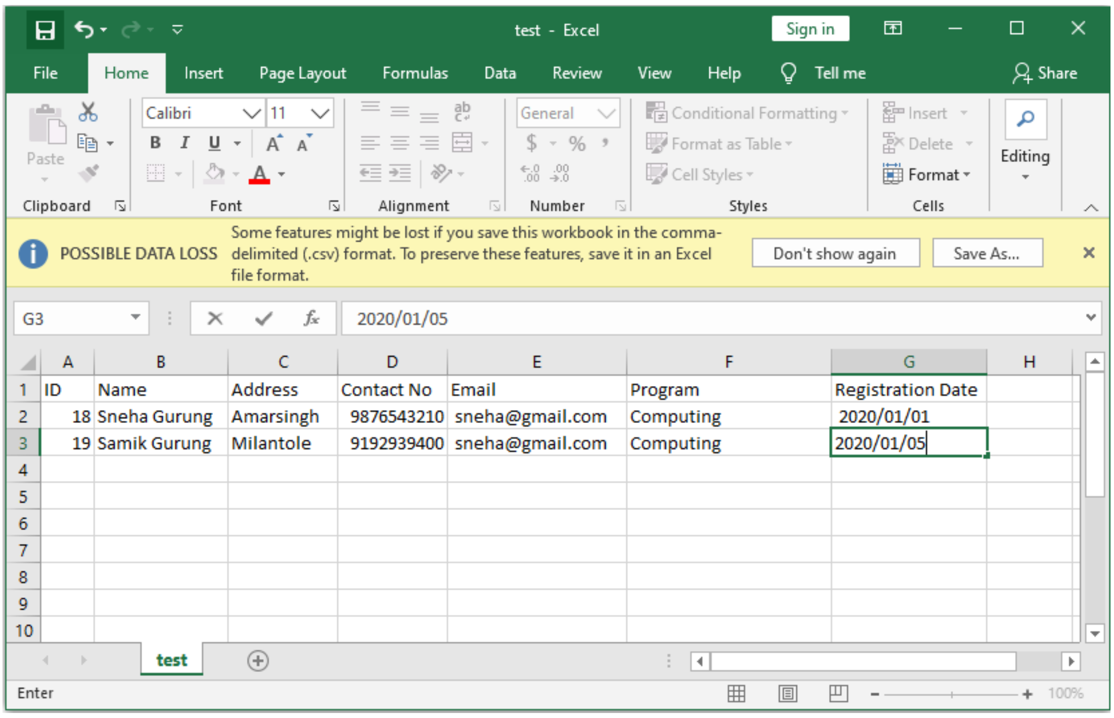


Name	Date modified	Type
.git	1/10/2020 10:39 AM	File folder
.vs	1/10/2020 10:40 AM	File folder
Application Development CW1	1/10/2020 10:41 AM	File folder
Application Development CW1.sln	1/2/2020 8:42 PM	Visual Studio So
test	1/10/2020 11:32 AM	Microsoft Excel

File name: test

Figure 10: importing excel file

- User import a record from a CSV files to allow manually inputting details like ID number, name, address, contact no, course enrol, registration date.
- Click the Upload Record button and you can select your file to input details manually.

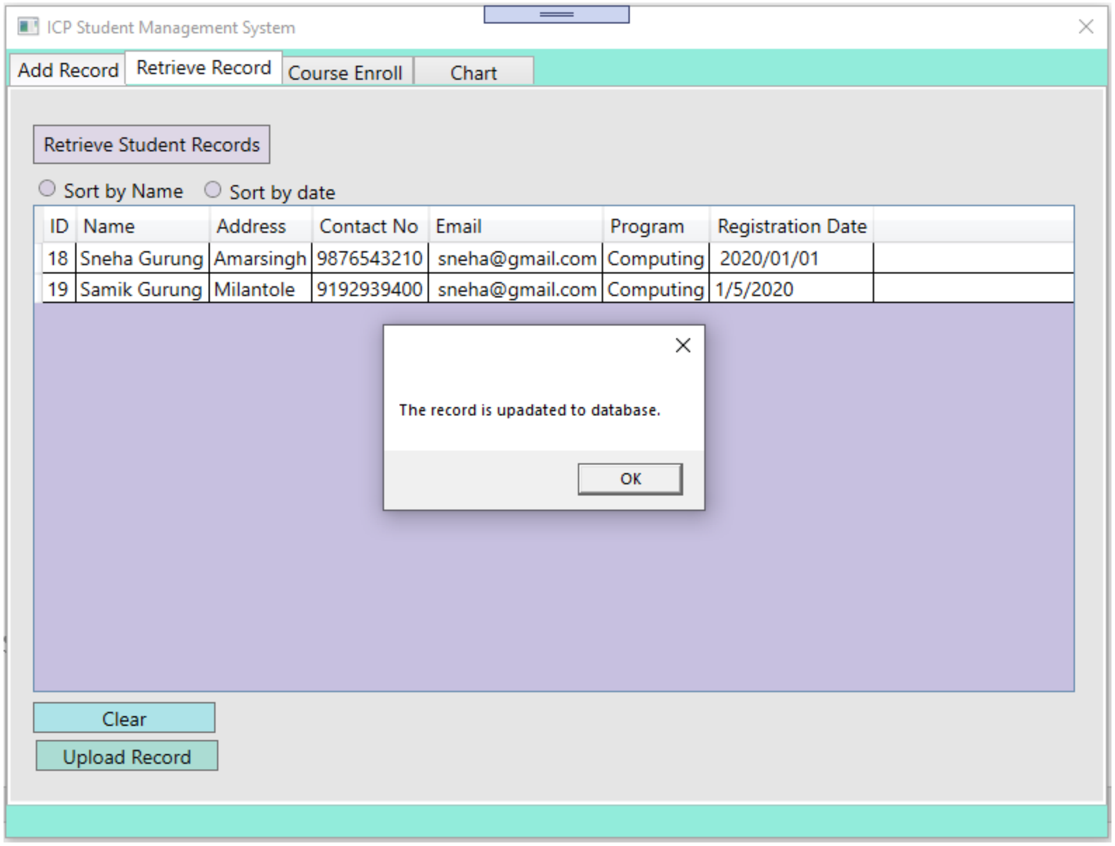


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

G3 2020/01/05

	A	B	C	D	E	F	G
1	ID	Name	Address	Contact No	Email	Program	Registration Date
2	18	Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
3	19	Samik Gurung	Milantole	9192939400	sneha@gmail.com	Computing	2020/01/05

Figure 11: Excel file



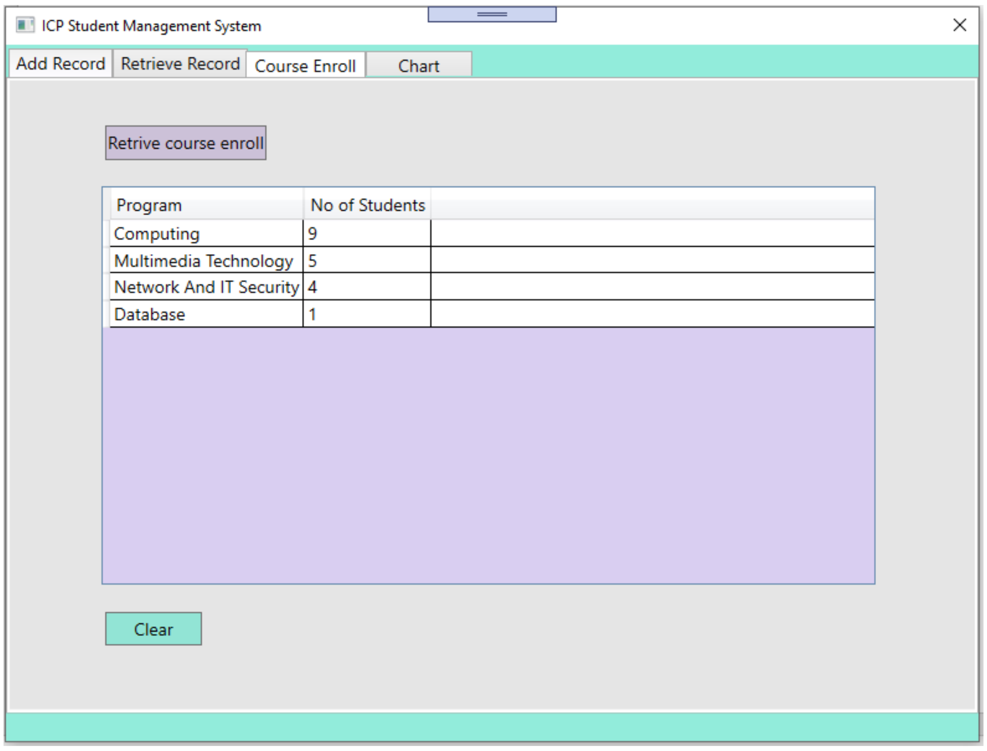
ID	Name	Address	Contact No	Email	Program	Registration Date
18	Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
19	Samik Gurung	Milantole	9192939400	sneha@gmail.com	Computing	1/5/2020

The record is upadated to database.

Figure 12: Excel data added manually

- After selecting the excel file, all the data involved in the files are added manually inputting details like ID number, name, address, contact no, email, course enrol and registration date.

Course Enrol

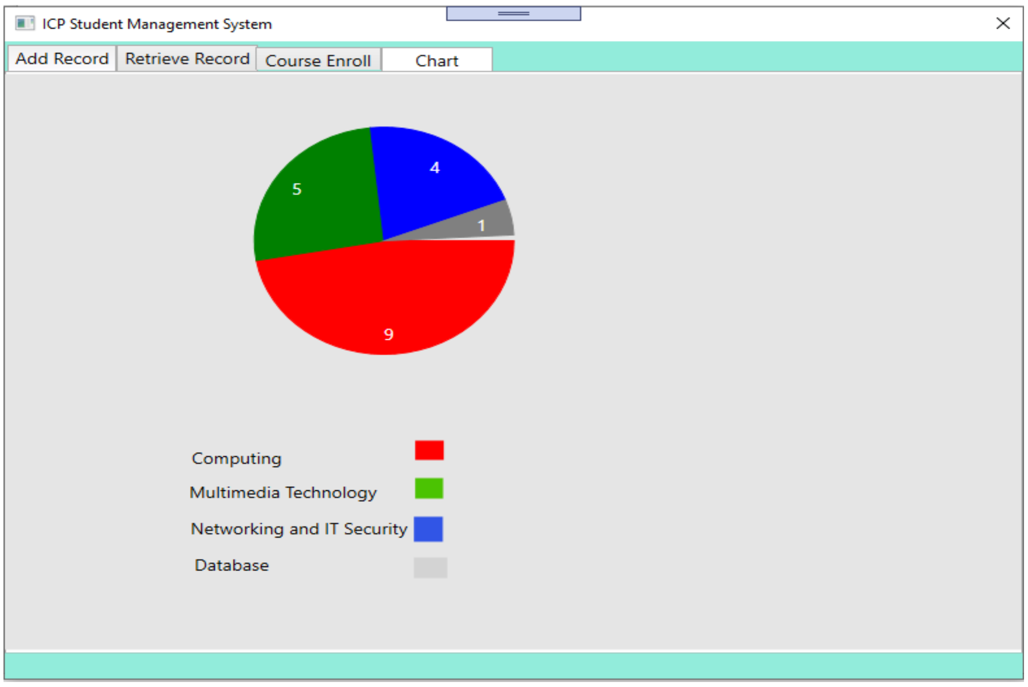


Program	No of Students
Computing	9
Multimedia Technology	5
Network And IT Security	4
Database	1

Figure 13: Course Enrol

- Course Enrol displays weekly tabular report showing total number of students enrolled so far in each program. It calculates the number of students enrolled in a specific course or program.

Chart



Computing	🟥
Multimedia Technology	🟩
Networking and IT Security	🟦
Database	⬜

Figure 14: Pie chart

- Chart includes a static form canvas of pie chart showing total number of students on each program computing, multimedia, networking and database.

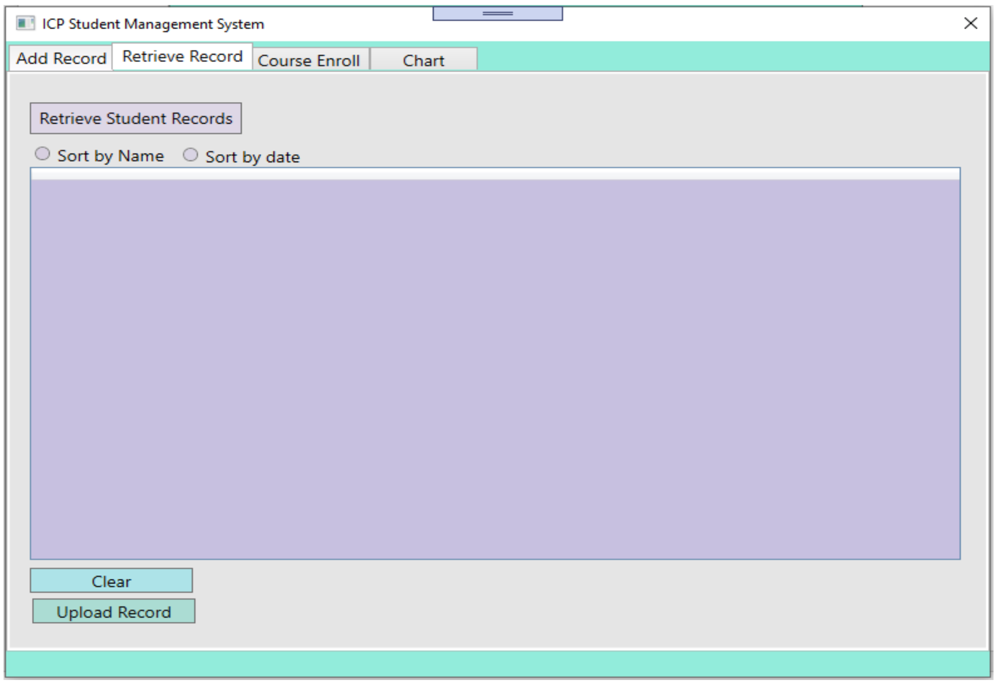


Figure 15: clear button for retrieve record

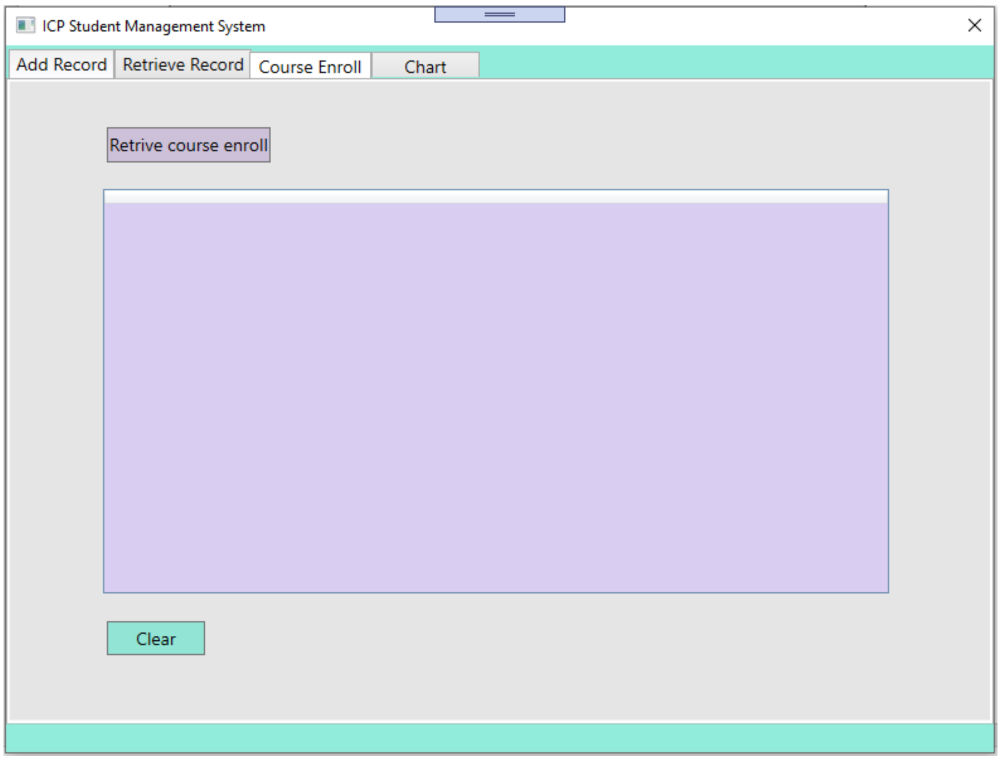


Figure 16: clear button for course enrol

- Both Retrieve Record and Course Enrol has a clear button which enables the DataGrid to hide the students details and course details respectively.

3. Journal Article

1. What is Student Management System Software?

The advantages of Student Management System software cannot be explained but could be summarized. The world has seen a tremendous growth in technology in past few decades than in centuries; with necessity in clouding and data management, many organizations have opened their gates to simplify procedures by reducing human effort. With SMS software much of paper work could be circumvented. Mishandling of data is biggest concerns in many organizations hence; to bypass misuses Management System applications are designed under regulate guidelines and directives. Collaborative coordination between faculty and students could be attained and, announcements can be published with a single click in bulk via these simple programs. Few applications give parents access to candidate's academic performances and behaviours. EduSys is reputed for facilitating numerous modules, from providing parents access to their wards profiles to tracking institutes-owned vehicles. SMS software makes library management flexible and orderly. To condense the merits of employing Student Management System software (edusys, n.d.).

2. Advantages of Student Management System

Managing an educational system requires careful planning and time management on everybody's part, and therefore the automated management system has become not only a norm but a necessity in today's world. These smart technologies improve the efficiency of teachers to educate their pupils in the most unique ways of bringing out the true creativity in them. With an enormous amount of information available on various channels and such little time to grasp it, time management becomes much more crucial. The same goes for an educational organization trying to manage a crowd of students, with efficiency and effectiveness only a system can provide. With such requirements, it comes important to include a managing tool to save everybody's precious time. A student management system brings in various features like admission, attendance, fee collection library, examination, timetable, transport tracking, student performance report etc. All these are low complexity tasks which require high accountability and accuracy, therefore these tasks can easily be delegated to our lesser counterparts; the machines (or software's). Student management system has become an essential part of school management, offering efficacy where it matters (fedena, 2017).

3. Top 7 benefits of student management system

   The first significant advantage of a student management system is that as a school you are able to keep proper track of data related to students. This includes areas such as fees being paid by students, examination records of the students, transport facilities being provided by the school and availed by the students, and usage of libraries and other school facilities by the students. As a school, you can access this data and more, by using a unique identification number of the student. In fact, apart from management people, students can also use these systems in order to keep track of their dues as well as class schedules. With the help of this system no longer would your admin guys need to type in all the data of the students as well as others in the school on excel sheets for long hours! It can all now be organized in a format that happens to be simple and accessible to one and all. This also means that you would be able to access data within a few seconds, something that would not have been possible without such a system in place. (fedena, 2017)

4. Student Management System – A Survey
   This is a paper published on the International research journal of computer science on the month of May of 2017. The author of the paper is Dr M Ramakrishna PG student & professor of computer science and engineering. In this paper professor illustrate about the current system, system design, proposed system, dataflow diagram etc. about the student management system. (irjcs, n.d.)

4. System Architecture

4.1 Architectural Diagram

The above figure represents the architecture of the developed system. At first, user needs to login to the system for which the user needs to input the correct credentials. After logging into the system with correct credentials, the system will display the main form which is the main panel of the developed system. The main screen provides four options in menu bar to click for which are Add Record, Retrieve Record, Course Enrol and Chart. Using the menu stripe on the main form, the user can record the students details with the registration date along Moreover, the user can generate a statistical pie chart report of the number of students on each program.

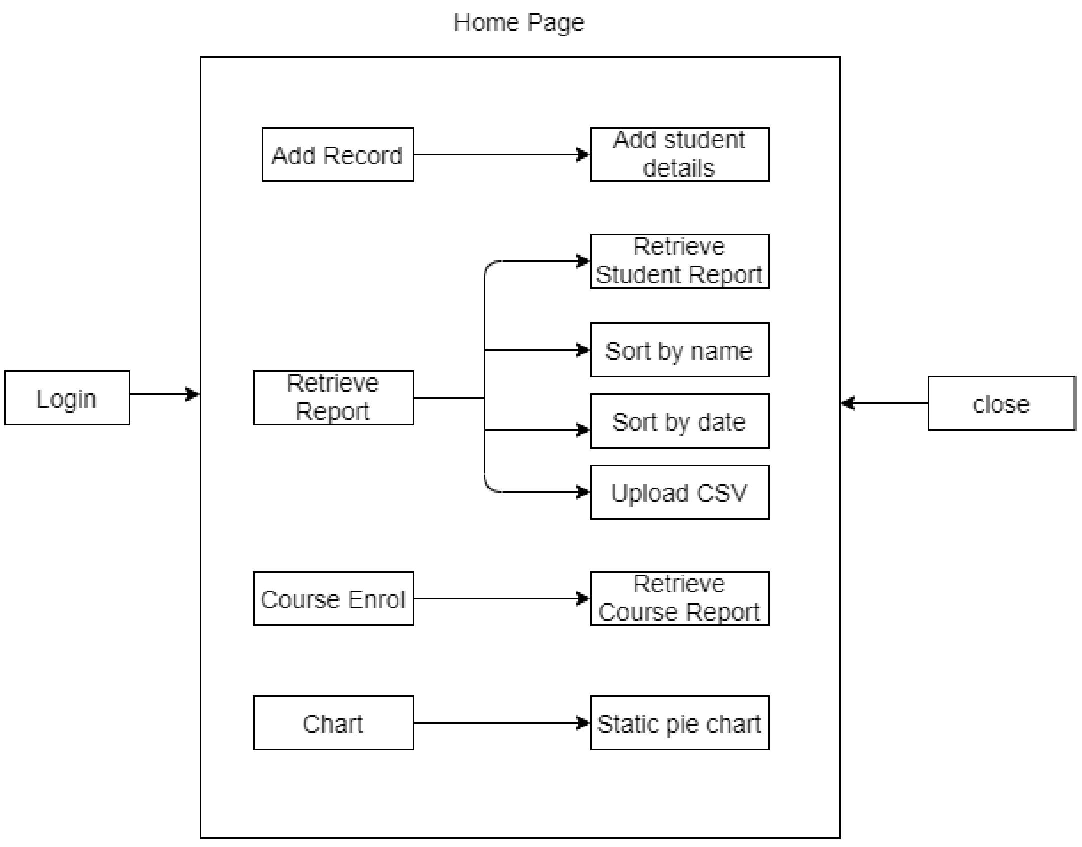


Figure 17: Architectural Diagram

4.2 Class Diagram

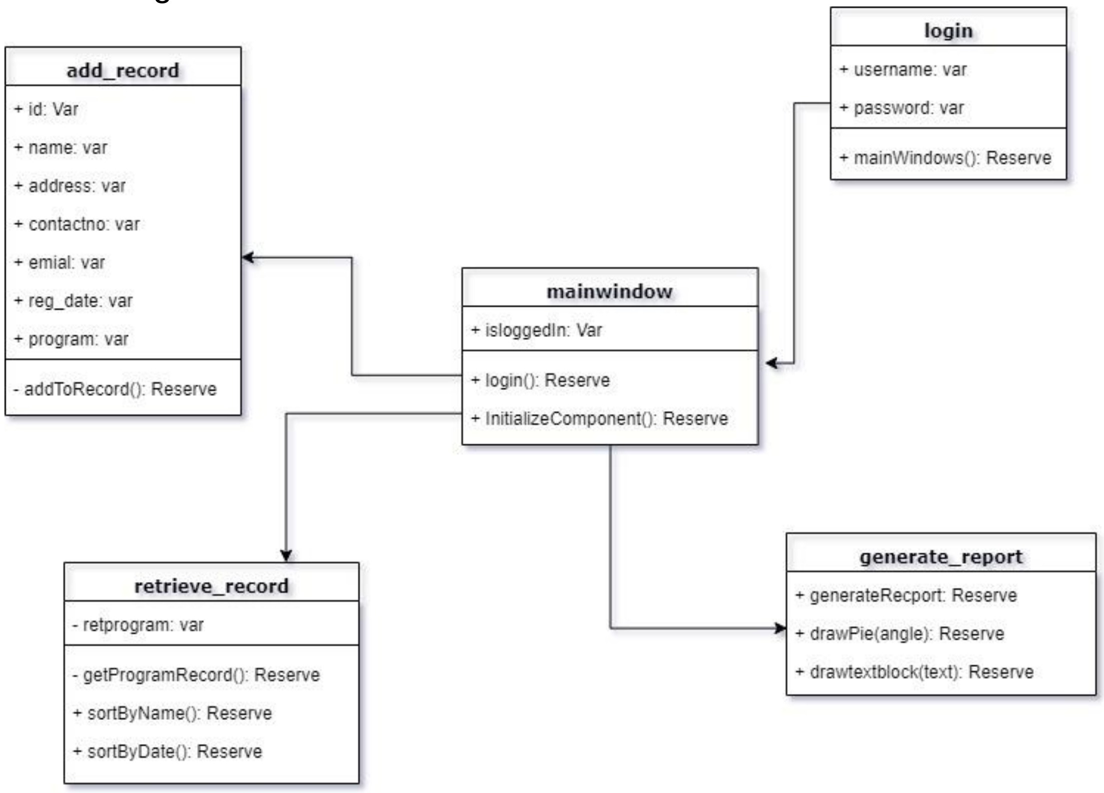


Figure 18: Class diagram

4.3 Class Description

Login

Methods	Description
mainWindow	The method redirect the program to the main widow after successful authentication.

Table 1: Login

mainWindow

Methods	Description
Login	The method opens the login window for authentication purpose.
InitializeComponent	This is built in method of the program. The method initialize

	of the wpf components of the method.

Table 2: Main Window

add_record

Methods	Description
addToRecord	This method is used to add record of student to the database. The method pulls all the data from the textbox verifies them and add them to the record.

Table 3: Add record

retrieve_record

Methods	Description
getStudentRecord	The method generates the student records available in the database.
sortByName	The method sort the student record in accordance to name of the students.
sortByDate	The method sort the student record in accordance to the registration date.

Table 4: Retrieve record

generateRecord

Methods	Description
generateRecord	The method generates the program record from the database.
drawPie	The method draws an arc on the canvas for drawing pie.
drawTextBlock	The method draws texts on the canvas for representation of the pie.

Table 5: Generate Report

4.4 Flowchart

Add to record

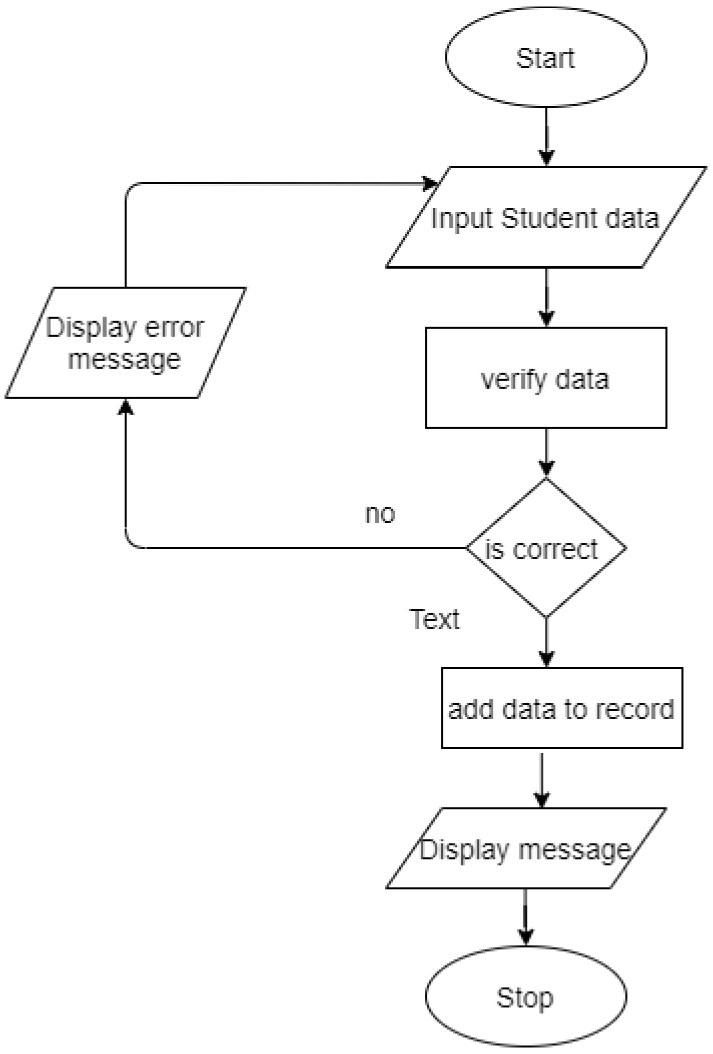


Figure 19: Flowchart of add data

Generate Report

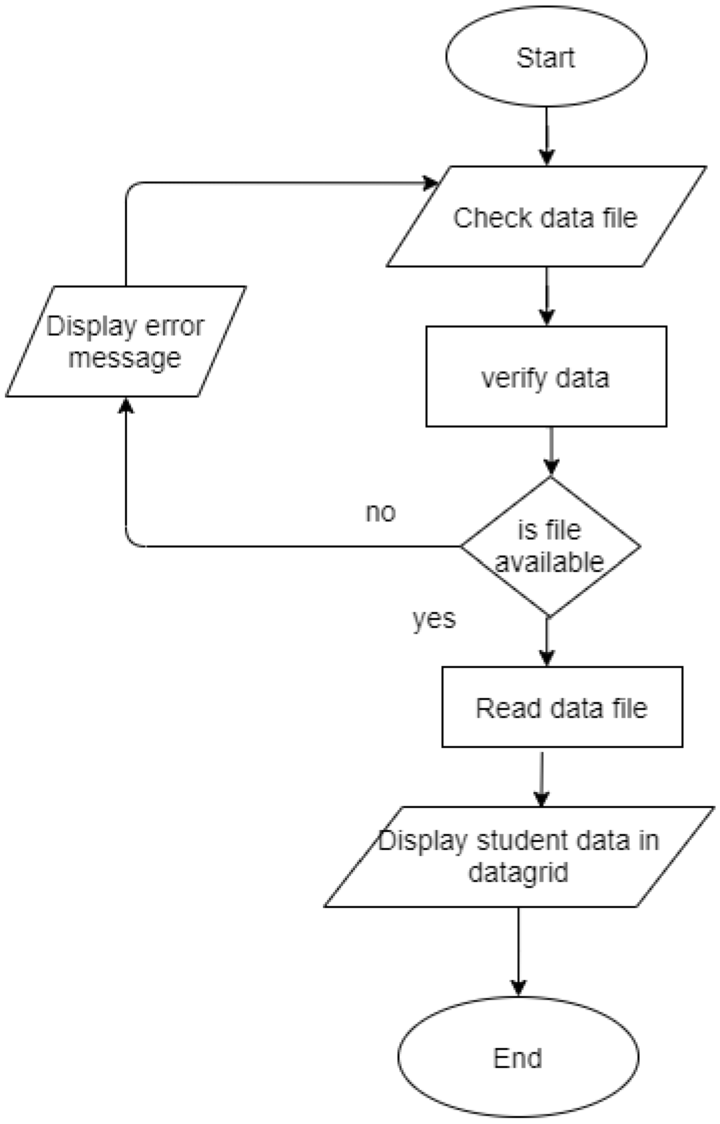


Figure 20: Flowchart for retrieve report

5. Sorting

For this project I have used Bubble sort algorithm. Bubble sort algorithm is a sorting algorithm. It is a step by step procedure that need to be sorted comparing each pair of elements and swapping them if they are in wrong order and arranging them in a certain order. Bubble sort algorithm starts by comparing the first two elements of an array and swapping if necessary, i.e. if you want to sort the elements of array in an ascending order and if the first element is greater than second then, you need to swap the elements but, if the first element is smaller than second, you mustn't swap the element. Then, again second and third elements are compared and swapped if it is necessary and this process go on until last and second last element is compared and swapped. This completes the first step of bubble sort.

If there are n elements to be sorted then, the process mentioned above should be repeated n-1 times to get required result. But, for better performance, in second step, last and second last elements are not compared because, the proper element is automatically placed at last after first step. Similarly, in third step, last and second last and second last and third last elements are not compared and so on.

This iteration process repeats over and over through the array comparing each pair of elements and swapping if necessary, to sort them in certain order. The process through the array is repeated until and unless no swaps are required. That's when it knows that the array is completely sorted.

1.2 Explanation of algorithm through example

i. First pass:

(7 2 5 3 9) - (2 7 5 3 9) swapping since 7>2

(2 7 5 3 9) - (2 5 7 3 9) swapping since 7>5

(2 5 7 3 9) - (2 5 3 7 9) swapping since 7>3

(2 5 3 7 9) - (2 5 3 7 9)

Taking an unsorted array (7 2 5 3 9) as an example to arrange the elements in an ascending order. Bubble sort algorithm starts with comparing the very first elements 7 and 2 and swaps, since 7 is greater than 2. Second and third elements are also compared where 7 is greater than 5 and swaps. Then again third and fourth elements are compared they are 7 and 3 and swaps because 7 is greater than 3. In addition to this, the fourth and the last element 7 and 9 respectively are compared. Now, since these elements are already in ascending order i.e. 9 is greater than 5 hence doesn't swap.

ii. Second pass:

(2 5 3 7 9) - (2 5 3 7 9)

(2 5 3 7 9) - (2 3 5 7 9) swapping since 5>3

(2 3 5 7 9) - (2 3 5 7 9)

(2 3 5 7 9) - (2 3 5 7 9)

Here only second and third elements which are 5 and 3 respectively gets swap as 5 is greater than 3. Since other pair of elements are already in ascending order it doesn't swap.

iii. Third pass:

(2 3 5 7 9) - (2 3 5 7 9)

(2 3 5 7 9) - (2 3 5 7 9)

(2 3 5 7 9) - (2 3 5 7 9)

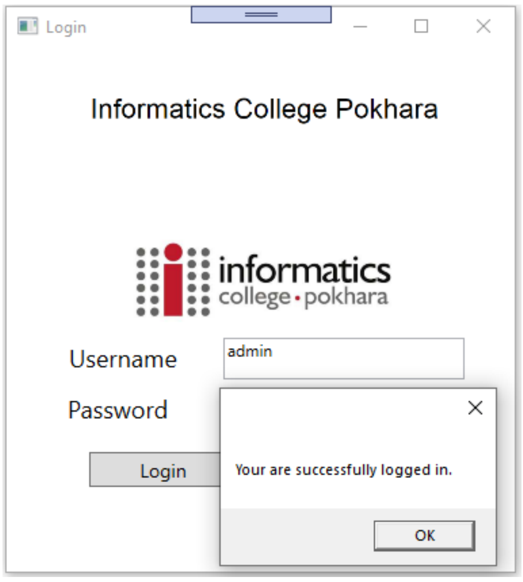
(2 3 5 7 9) - (2 3 5 7 9)

As all these elements are already in an ascending order, bubble sort algorithm still has to go through the whole iteration process because it doesn't know if the elements are arranged completely or not. So, it needs to repeat the whole procedure one more time without any swaps. And when there is no swap required, then bubble sort algorithm knows that array is sorted completely. This is the end of the procedure. (programiz, n.d.)

6. Testing

Test case 1

Objective: To analyse whether logging in with right username or password is possible or not.



admin

Your are successfully logged in.

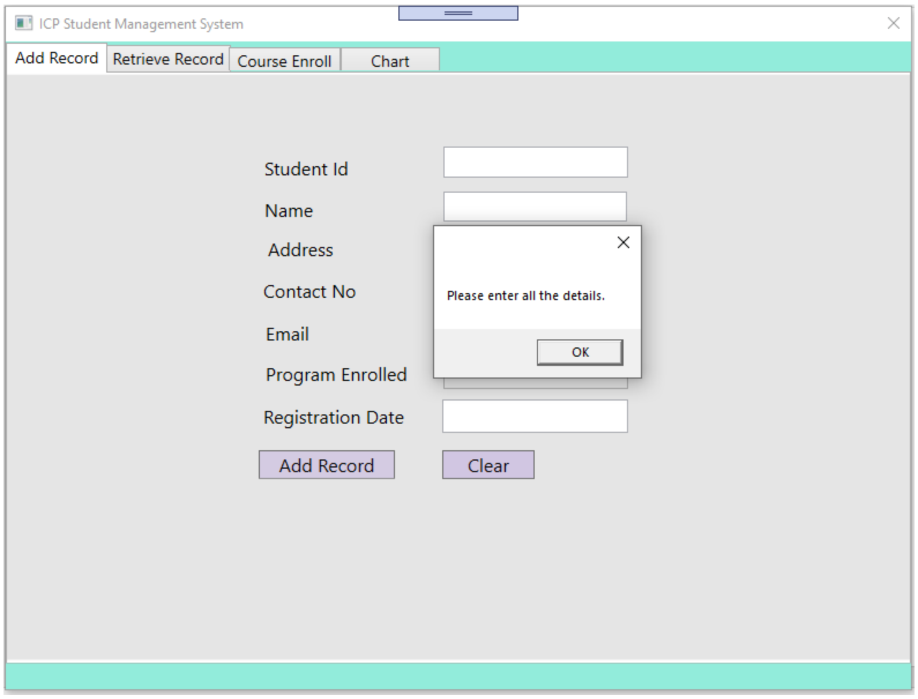
Figure 21: Test case 1

Result: successful log in with correct username and password is possible.

Status: Test is successful.

Test case 2

Objective: To analyse whether add record button works or not when the student’s details are empty.



Please enter all the details.

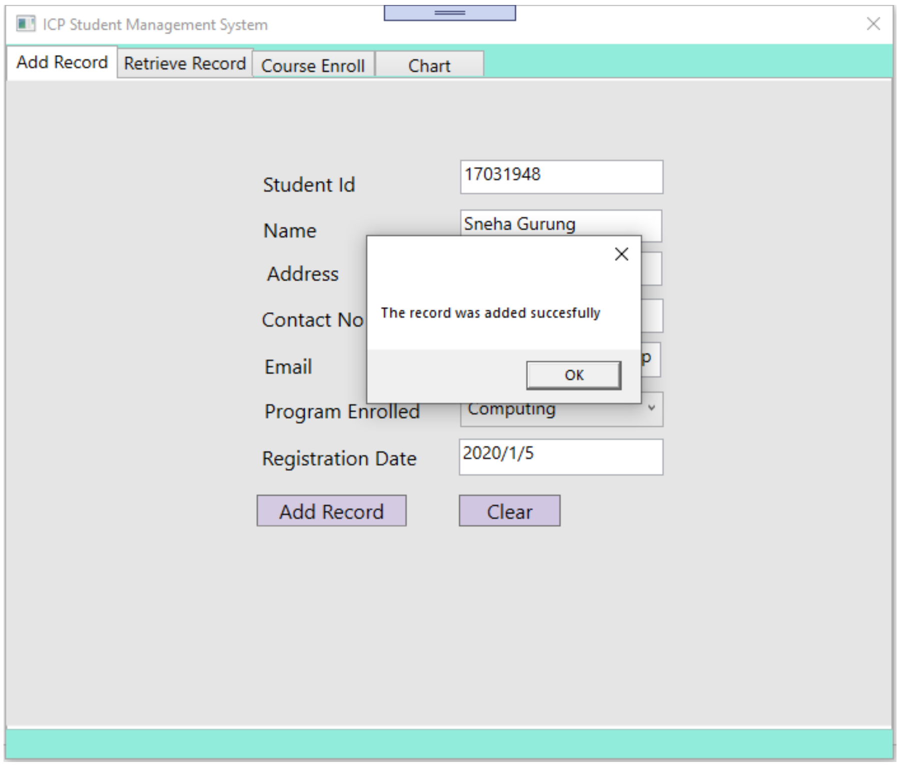
Figure 22: Test case 2

Result: Shows a popup message box to ask user to enter all the details.

Status: Test is successful.

Test case 3

Objective: To analyse whether the add record button works or not



Student Id: 17031948

Name: Sneha Gurung

Program Enrolled: Computing

Registration Date: 2020/1/5

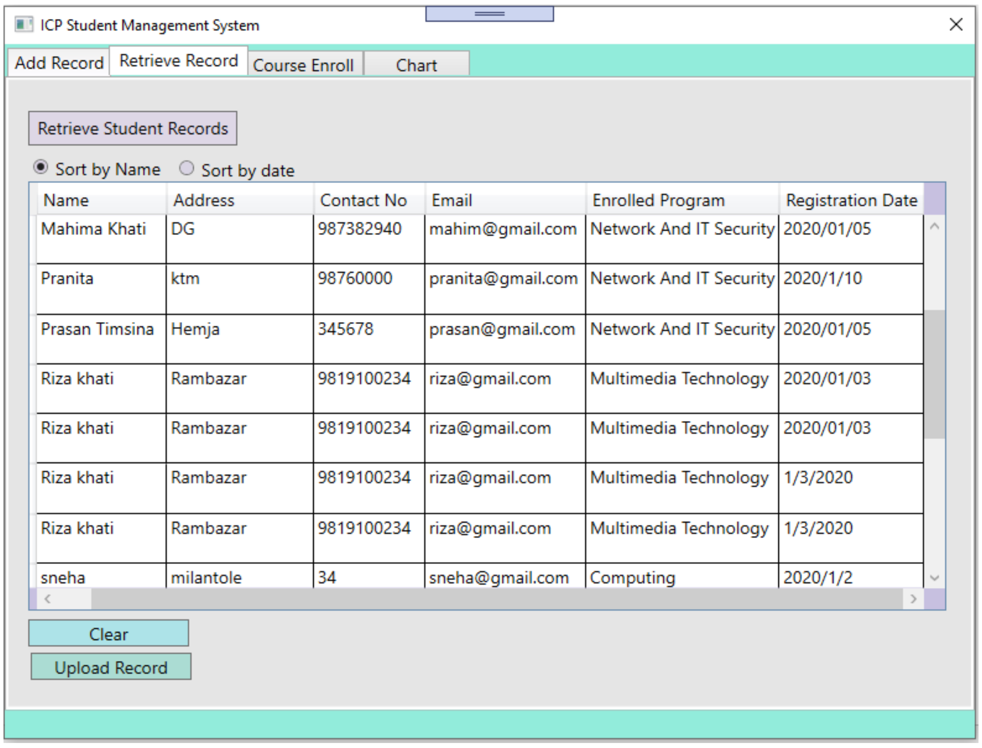
The record was added succesfully

Figure 23: Test case 3

Result: Successfully add. Status: Test is successful.

Test case 4

Objective: To check whether sorting buttons sorts data by its name and by the registration date displayed DataGrid.



ICP Student Management System

Add Record | Retrieve Record | Course Enroll | Chart

Retrieve Student Records

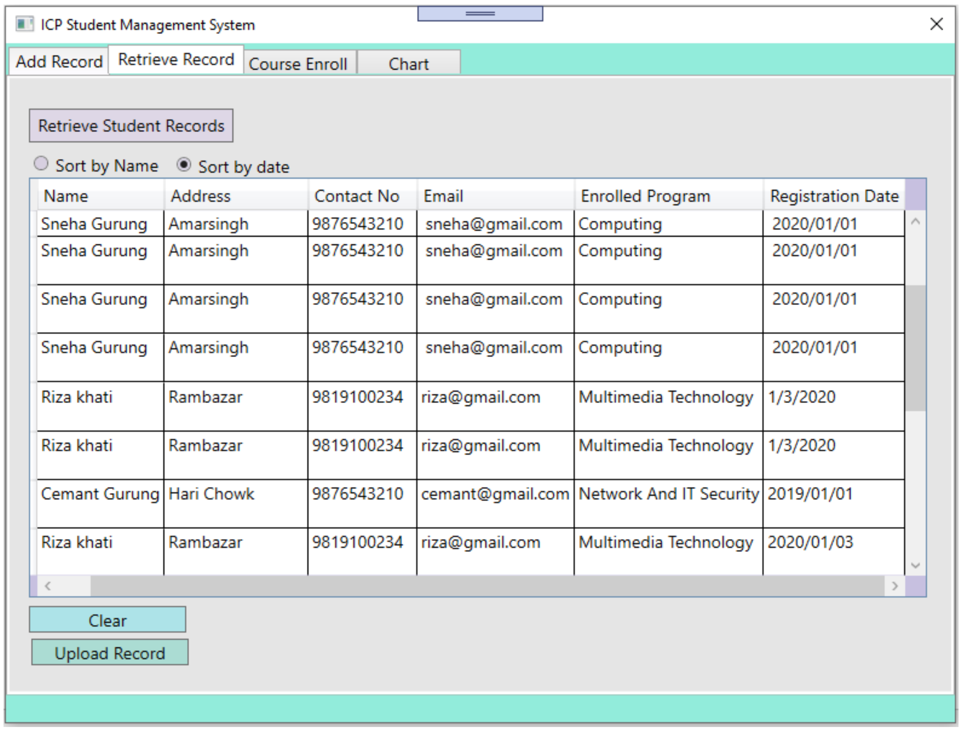
[✓ Sort by Name] Sort by date

Name	Address	Contact No	Email	Enrolled Program	Registration Date
Mahima Khati	DG	987382940	mahim@gmail.com	Network And IT Security	2020/01/05
Pranita	ktm	98760000	pranita@gmail.com	Network And IT Security	2020/1/10
Prasan Timsina	Hemja	345678	prasan@gmail.com	Network And IT Security	2020/01/05
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	2020/01/03
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	2020/01/03
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
sneha	milantole	34	sneha@gmail.com	Computing	2020/1/2

Clear

Upload Record

Figure 24: Test case sort by name



ICP Student Management System

Add Record | Retrieve Record | Course Enroll | Chart

Retrieve Student Records

Sort by Name [✓ Sort by date]

Name	Address	Contact No	Email	Enrolled Program	Registration Date
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Sneha Gurung	Amarsingh	9876543210	sneha@gmail.com	Computing	2020/01/01
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	1/3/2020
Cemant Gurung	Hari Chowk	9876543210	cemant@gmail.com	Network And IT Security	2019/01/01
Riza khati	Rambazar	9819100234	riza@gmail.com	Multimedia Technology	2020/01/03

Clear

Upload Record

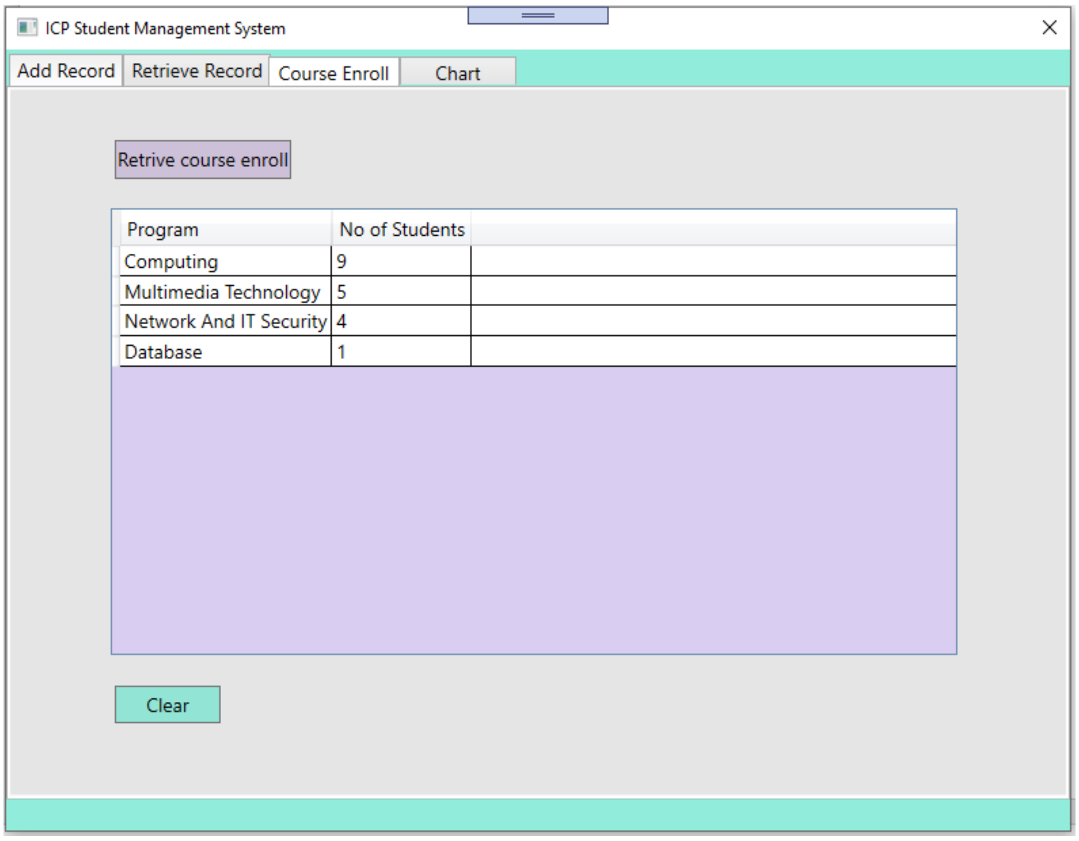
Figure 25: Test case sort by date

Result: Successfully displays the student details sorted by the first name of the student and another sorted by their registration date.

Status: Test is successful

Test case 5

Objective: To analyse whether retrieve button works or not.



Program	No of Students
Computing	9
Multimedia Technology	5
Network And IT Security	4
Database	1

Figure 26: Test case 5

Result: Successfully displays the course details and number of student enrol.

Status: Test is successful

7. Reflection

The foremost purpose of this project is to design and implement Student Information System in C# - WPF. Developing this system in Microsoft Visual Studio 2019 keeping C# as primary programming language is new experience for me. I have created a desktop system using java before but learning it on visual studio with C# was tough. Serialization and deserialization are another new thing I learn while developing the system. Though, creating new classes and methods helps to pace the development task. Importing and exporting of CSV file is also a new task and it really help me in gaining knowledge of file handling. Creating a class diagram within the visual studio helps me in documentation phase. With the growing of technology, the visual studio and its community helps newbie developer like us to pace our development speed. Overall, I get to learn Window Presentation Foundation system also abbreviated ad WPF. The WPF development platform helps me supports a broad set of application development features, including an application model, resources, controls, graphics, layout, data binding, documents, and security. The framework is part of .NET and I have never done it previously so, building an application with .NET using ASP.NET or Windows Forms is a great experience. I also get to learn Extensible Application Markup Language (XAML) to provide a declarative model for application programming.

This bring to the end of this project. The excess research, group deliberation, guide of our module leader helps me to break the fence to reach my goal of finishing this course work on time. This coursework has made me habitual with implementing codes and inspires me to start projects on new languages as well. Hence, the knowledge and experience I have gained through this project will definitely assist me in future.

Bibliography

edusys. (n.d.). Retrieved from What is Student Management System Software: https://www.edusys.co/blog/what-is-student-management-system-software

fedena. (2017). Retrieved from Advantages of Student Management System: https://fedena.com/blog/2017/07/advantages-of-student-management-system.html

fedena. (2017). Retrieved from Top 7 benefits of Student Management System: https://fedena.com/blog/2018/02/student-management-system.html

irjcs. (n.d.). Retrieved from School Management System Software: http://irjcs.com/volumes/vol4/iss05/46.MYCSSP10083.pdf

programiz. (n.d.). Retrieved from Bubble Sort Algorithm: https://www.programiz.com/dsa/bubble-sort

Appendix

Login.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Application_Development_CW1
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Window
    {
        public static bool isLoggedIn = false;
        public Login()
        {
            InitializeComponent();
        }

        private void button_login_Click_1(object sender, RoutedEventArgs e)
        {

            if (tbox_username.Text.Trim() == "" && tbox_password.Text.Trim() ==
"")
            {
                MessageBox.Show("Please enter username & password.");
            }
            else
            {
                if (tbox_username.Text == "admin" && tbox_password.Text ==
"admin")
                {
                    Login.isLoggedIn = true;

                    MessageBox.Show("Your are successfully logged in.");
                    MainWindow mainWindow = new MainWindow();
                    mainWindow.Show();
                    this.Close();

                }
                else
                {
                    MessageBox.Show("Username & Password is incorrect.");
                }

            }
        }

        private void button_clear_Click_1(object sender, RoutedEventArgs e)
        {
            tbox_username.Clear();
            tbox_password.Clear();
        }
    }
}
```

MainWindow.xaml.cs

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Application_Development_CW1
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            if (!Login.isLoggedIn)//login windows is called for
authentification purpose.
            {
                Login login = new Login();
                login.Show();
                this.Hide();
            }

        }

        public void clearField()
        {
            tbox_id.Text = "";
            tbox_name.Text = "";
            tbox_address.Text = "";
            tbox_contact.Text = "";
            tbox_email.Text = "";
            tbox_regdate.Text = "";
            datagrid_1.ItemsSource = "";
            combox_1.SelectedItem = "";
            radbutton_sortbydate.IsChecked = false;
            radbutton_sortbyname.IsChecked = false;

            datagrid_2.ItemsSource = "";

        }
        private void button_clear_Click(object sender, RoutedEventArgs e)
        {
            clearField();
        }
        private void addRecord(string id, string name, string address, string
contactno, string email, string enrolprogram, string enroldate)
        {
            StringBuilder csvcontent = new StringBuilder();
            csvcontent.AppendLine(id + "," + name + "," + address + "," +
contactno + "," + email + "," + enrolprogram + "," + enroldate);
            string csvfilepath = "Record.csv";
            File.AppendAllText(csvfilepath, csvcontent.ToString());
        }

        private void button_addrec_Click(object sender, RoutedEventArgs e)
        {

            if (tbox_id.Text.Trim() == "" || tbox_name.Text.Trim() == "" ||
tbox_address.Text.Trim() == "" || tbox_contact.Text.Trim() == "" ||
tbox_email.Text.Trim() == "" || tbox_regdate.Text.Trim() == "")
            {
                MessageBox.Show("Please enter all the details.");
            }
            else
            {
                string program = "";
                if (combox_1.Text == "Computing") {
                    program = "Computing";
                }
                else if (combox_1.Text == "Multimedia Technology") {
                    program = "Multimedia Technology";
                }
                else if (combox_1.Text == "Network And IT Security")
                {
                    program = "Network And IT Security";
                }
                else if (combox_1.Text == "Database")
                {
                    program = "Database";
                }
                addRecord(tbox_id.Text, tbox_name.Text, tbox_address.Text,
tbox_contact.Text, tbox_email.Text, program, tbox_regdate.Text);
                MessageBox.Show("The record was added succesfully");
                clearField();
            }
        }
        private void retrieveStudentRecord()
        {
            string delimiter = ",";
            string tableName = "Student Record";
            String FileName = "Record.csv";

            DataSet dataset = new DataSet();
            StreamReader sr = new StreamReader(FileName);

            dataset.Tables.Add(tableName);
            dataset.Tables[tableName].Columns.Add("ID");
            dataset.Tables[tableName].Columns.Add("Name");
            dataset.Tables[tableName].Columns.Add("Address");
            dataset.Tables[tableName].Columns.Add("Contact No");
            dataset.Tables[tableName].Columns.Add("Email");
            dataset.Tables[tableName].Columns.Add("Enrolled Program");
            dataset.Tables[tableName].Columns.Add("Registration Date");

            string allData = sr.ReadToEnd();
            string[] rows = allData.Split("\r".ToCharArray());

            foreach (string r in rows)
            {
                string[] items = r.Split(delimiter.ToCharArray());
                dataset.Tables[tableName].Rows.Add(items);

            }
            this.datagrid_1.ItemsSource = dataset.Tables[0].DefaultView;
        }

        private void button_retrec_Click(object sender, RoutedEventArgs e)
        {
            if (radbutton_sortbyname.IsChecked == true)
            {
                retrieveStudentRecord();
                Sort("Name");
            }
            else if (radbutton_sortbydate.IsChecked == true)
            {
                retrieveStudentRecord();
                Sort("Registration Date");
            }
            else
            {
                retrieveStudentRecord();
            }
        }
        private void getProgramRecord()
        {
            var temp = File.ReadAllLines("Record.csv");
            int com = 0, mul = 0, net = 0, dat =0;
            foreach (string line in temp)
            {
                var delimitedLine = line.Split(',');
                if (delimitedLine[5] == "Computing")
                {
                    com++;
                }
                else if (delimitedLine[5] == "Multimedia Technology")
                {
                    mul++;
                }
                else if (delimitedLine[5] == "Network And IT Security")
                {
                    net++;
                }
                else if (delimitedLine[5] == "Database")
                {
                    dat++;
                }
            }
            DataTable dt = new DataTable();
            dt.Columns.AddRange(new DataColumn[2] {new DataColumn("Program",
typeof(String)),
            new DataColumn("No of Students",typeof(int))});
            dt.Rows.Add("Computing", com);
            dt.Rows.Add("Multimedia Technology", mul);
            dt.Rows.Add("Network And IT Security", net);
            dt.Rows.Add("Database", dat);
            this.datagrid_2.ItemsSource = dt.DefaultView;
            int total = com + mul + net+ dat;
            drawPie((com * 360 / total), Brushes.Red, com);
            drawPie((mul * 360 / total), Brushes.Green, mul);
            drawPie((net * 360 / total), Brushes.Blue, net);
            drawPie((dat * 360 / total), Brushes.Gray, dat);

        }

        Point lastArcPoint = new Point(9999, 9999);
        int lastAngle = 0;

        private void drawPie(int angle, SolidColorBrush color,int prog)
        {
            int curentSliceAngle = angle;
            lastAngle += angle;
            angle = lastAngle;
            PieCanvas2.Width = 200;
            PieCanvas2.Height = 200;
            Double midPoint = 100;

            System.Windows.Shapes.Path path = new System.Windows.Shapes.Path();
            path.Fill = color;
            path.Stroke = color;
            PathGeometry pathGeometry = new PathGeometry();
            PathFigure pathFigure = new PathFigure();
            pathFigure.StartPoint = new Point(midPoint, midPoint);
            pathFigure.IsClosed = true;
            Double radius = midPoint;

            LineSegment lineSegment = new LineSegment(new Point(radius +
midPoint, midPoint), true);
            if (lastArcPoint.X != 9999 && lastArcPoint.Y != 9999)
            {
                lineSegment = new LineSegment(new Point(lastArcPoint.X,
lastArcPoint.Y), true);
            }
            pathFigure.Segments.Add(lineSegment);
            ArcSegment arcSegment = new ArcSegment();
            arcSegment.Point = new Point(midPoint + Math.Cos(angle * Math.PI /
180) * radius, midPoint + Math.Sin(angle * Math.PI / 180) * radius);
            lastArcPoint = new Point(arcSegment.Point.X, arcSegment.Point.Y);
            arcSegment.Size = new Size(radius, radius);
            arcSegment.SweepDirection = SweepDirection.Clockwise;
            pathFigure.Segments.Add(arcSegment);
            pathGeometry.Figures.Add(pathFigure);
            path.Data = pathGeometry;
            PieCanvas2.Children.Add(path);
            Point labelPoint = new Point(midPoint + Math.Cos((angle -
curentSliceAngle / 2) * Math.PI / 180) * radius * 0.8, midPoint +
Math.Sin((angle - curentSliceAngle / 2) * Math.PI / 180) * radius * 0.8);
            drawText(labelPoint.X - 7, labelPoint.Y - 7, prog.ToString());

        }

        private void drawText(double x, double y, string text)
        {

            TextBlock textBlock = new TextBlock();
            textBlock.Text = text;

            textBlock.Foreground = Brushes.White;

            Canvas.SetLeft(textBlock, x);
            Canvas.SetTop(textBlock, y);

            PieCanvas2.Children.Add(textBlock);

        }

        private void button_progrec_Click(object sender, RoutedEventArgs e)
        {
            getProgramRecord();
            tblox_piechart2.Text = "Pie Chart based on Program Record.";
        }

        private void Sort(string sortBy)
        {
            ICollectionView dataView =
              CollectionViewSource.GetDefaultView(datagrid_1.ItemsSource);

            dataView.SortDescriptions.Clear();
            SortDescription sd = new SortDescription(sortBy,
ListSortDirection.Ascending);
            dataView.SortDescriptions.Add(sd);
            dataView.Refresh();
        }


        private void button_clear1_Click(object sender, RoutedEventArgs e)
        {
            clearField();
        }

        private void button_uploadcsv_Click(object sender, RoutedEventArgs e)
        {
            uploadRecord();
        }
        public void uploadRecord()
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            if (openFileDialog.ShowDialog() == true)
            {
                string[] raw_text =
System.IO.File.ReadAllLines(openFileDialog.FileName);
                string[] data_col = null;
                int x = 0;
                DataTable dataTable = new DataTable();
                foreach (string text_line in raw_text)
                {
                    data_col = text_line.Split(",");
                    if (x == 0)

                        for (int i = 0; i <= data_col.Count() - 1; i++)
                        {
                            dataTable.Columns.Add(data_col[i]);
                            //MessageBox.Show(data_col[i]);
                        }
                    else
                    {
                        dataTable.Rows.Add(data_col);
                        addRecord(data_col[0], data_col[1], data_col[2],
data_col[3], data_col[4], data_col[5], data_col[6]);
                    }

                    x++;

                }
                this.datagrid_1.ItemsSource = dataTable.DefaultView;
                MessageBox.Show("The record is upadated to database.");

            }
        }

        private void CourseEnrollBtn_Click(object sender, RoutedEventArgs e)
        {
            getProgramRecord();

        }

        private void CourseEnrollClearBtn_Click(object sender, RoutedEventArgs
e)
        {
            clearField();
        }

        private void combox_1_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {

        }

        private void pieTb_TextChanged(object sender, TextChangedEventArgs e)
        {

        }

        private void TabControl_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {

        }
    }
}
```