Marking Scheme

Informatics College Pokhara

Application Development
CS6004NI
Course Work 1

Submitted By: **Deepesh K.C.**
London Met ID: Enter ID Here

Submitted To: Ishwor Sapkota
Module Leader

Component Grade and Comments	
A. Implementation of Application	
User Interface and proper controls used for designing	User Interface is complete but not separated and have proper use of controls
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.

Marking Scheme

Application architecture & description of the classes ad methods sued	very poorly explained.
Flow chart, algoriathms and data sctructures used	very poorly explained and no diagramatic representation
Reflective essay	Very poorly written
C. Programming Style	
Clarity of code,Popper Naming convention & comments	very poorly written code and no comments at all
System Usability	very poorly developed application

Overall Grade:	**F+**	**F+**

Overall Comment:

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

In overall the code is working and all the functionality seems working and system can be used

Module Code & Module Title

CS6004NA – Application Development

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2019-20 Autumn

Name: Dipesh K.C.

College ID: NP04CP4S180006

University ID: 17031941

Abstract

This coursework is of developing the desktop application and implementing a C# desktop application that helps education institute to manage the student information enrol record. This application will help to keep track of each new student detail with date. With the help of this application, education institute staff can also check weekly report. The chart for the report of the student will also be generated. Thus, with the help of user-friendly GUI, the institution staff can easily implement this application.

Table of Contents

Tables of Figures

1. Introduction

The coursework is about developing and implementing a C# desktop application that works in Student Information System. This application helps the student management system to input the new enrolled students information manually and system automatically import the detail to a CSV file. In this system there are features of student ID number, Name, Address, email address, phone number along with registration date and specific faculty. The system generates daily and weekly report and sort them by date and name. The information about daily and weekly report is shown in pie chart.

1.1 Current Scenario

In the era of advancement and digitalization some of the educational institute is still using traditional method to record the data. Filing data record system is used as a traditional method keeping the record in a paper file. There are some advanced and electronic digital system but somehow the system is lacking with the features which are essential for recording the Student Information System.

1.2 Proposed System

The system is designed to overcome the mentioned problem and which is fully digitalized. The system has a security system having a login username and password. The presence of different data mode display has made the way more systematic and easy to access and user friendly.

2. User Manual

Different functions screenshot has been mentioned below with the illustrate where a user can know how to operate the system.

The initial screen will be a login form where after login with specific username and password can logged in and enter to the system. This login method will play as a security role. The login username is “dipesh” and password is “dipesh”.

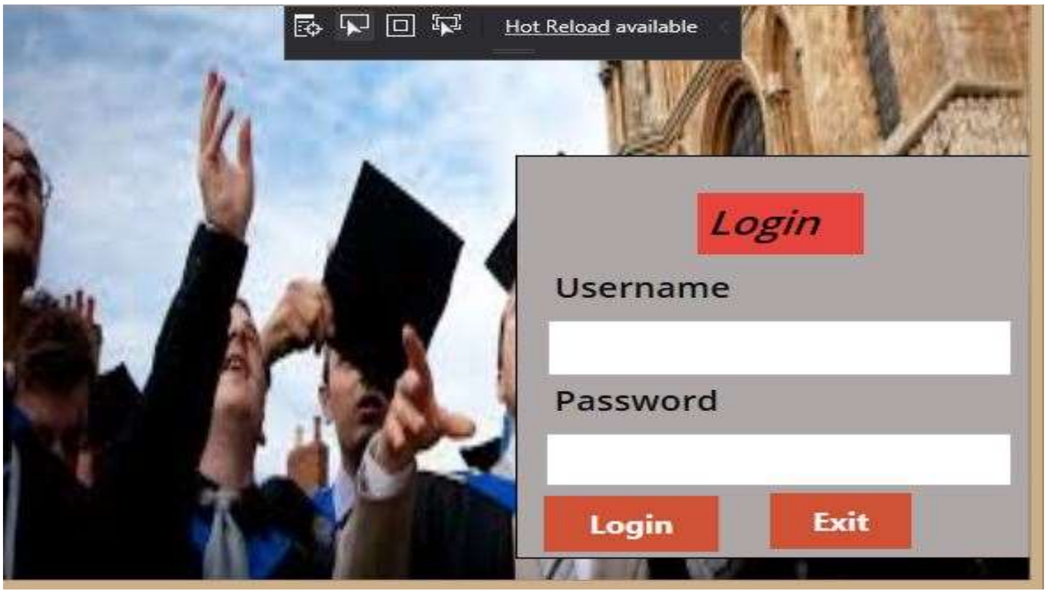
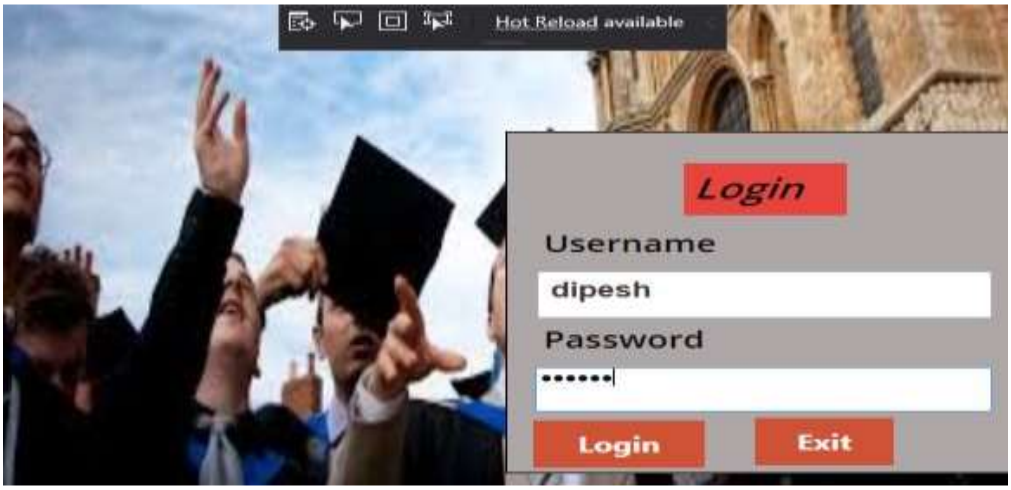


Figure 1: Login Page

This is the form of Login Page.



Username

dipesh

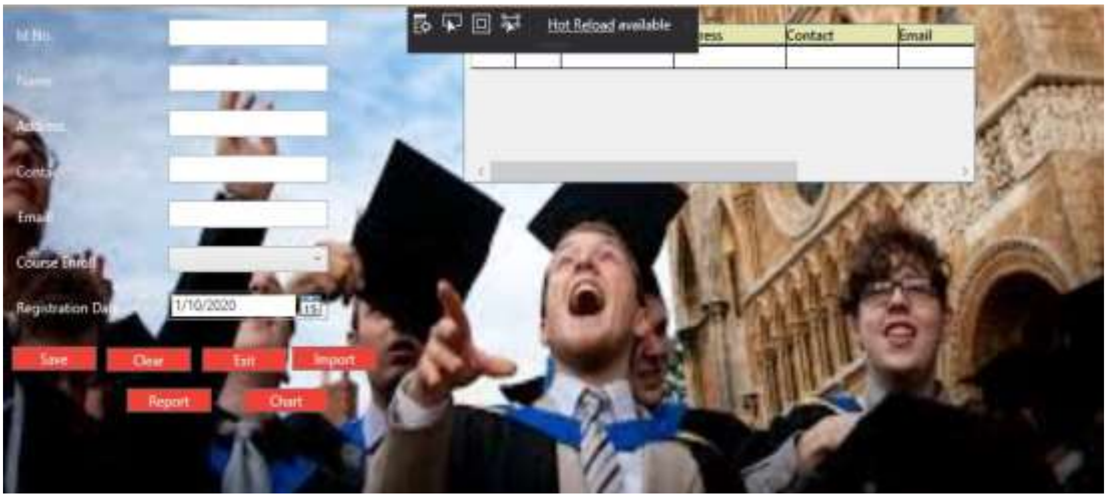
Password

•••••••

Figure 2: Login Page with form fill

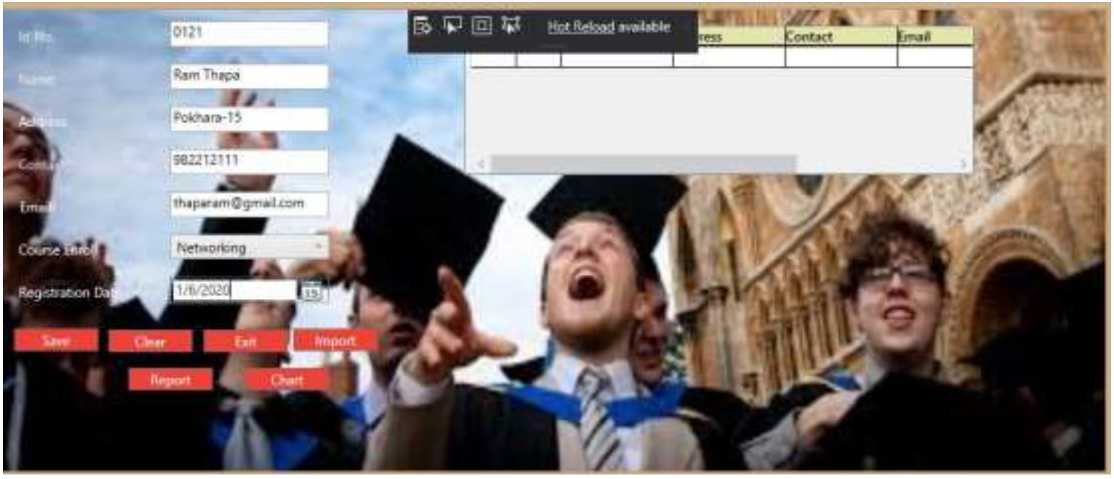
Login page after form filling up and without username and password.

After success in a login page the main screen of the system will appears.



1/10/2020

Figure 3: Main Home Page



0121

Ram Thapa

Pokhara-15

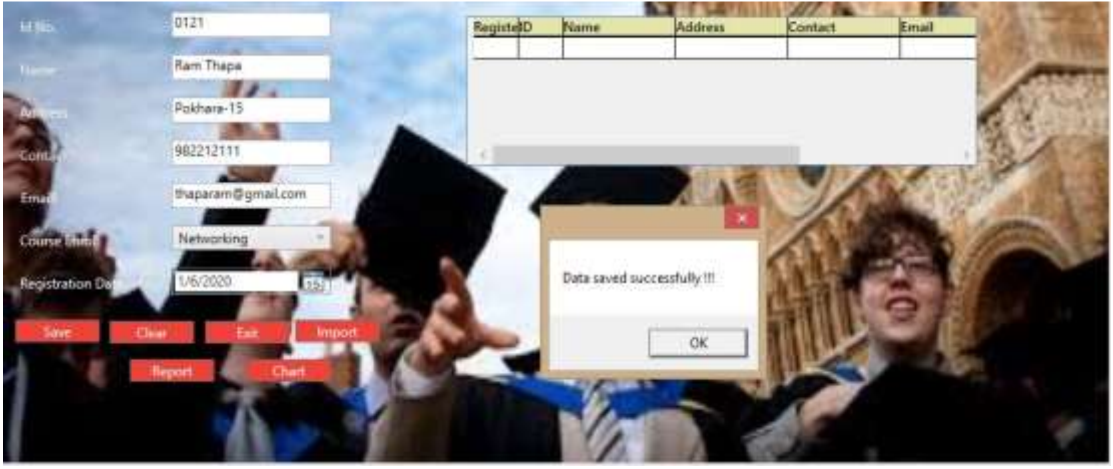
982212111

thaparam@gmail.com

Networking

1/6/2020

Figure 4: Main Page fill with data



0121

Ram Thapa

Pokhara-15

982212111

thaparam@gmail.com

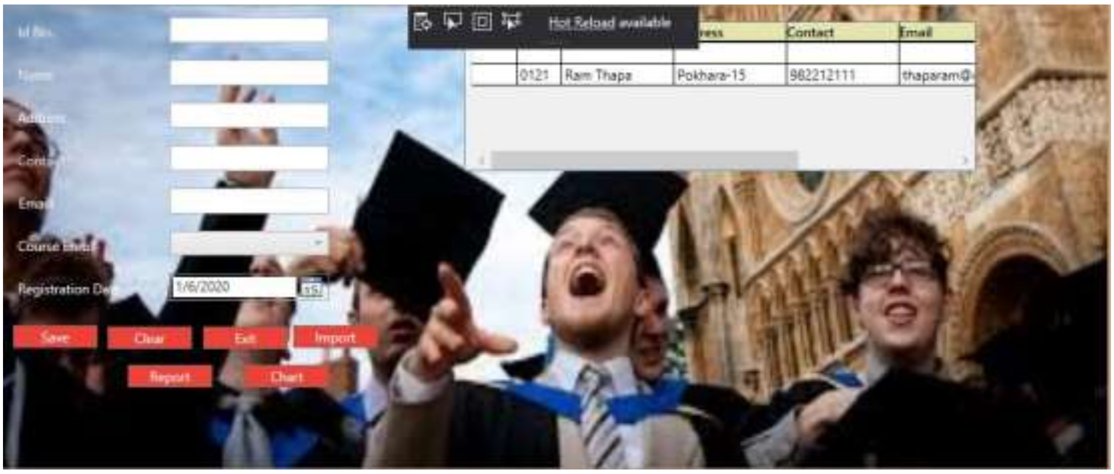
Networking

1/6/2020

Data saved successfully !!!

Figure 5: Data Save Successfully

Main Page without data and after filling the data. The data was saved successfully and shown on the display accordingly.


Id No.

Name

Address

Contact

Email

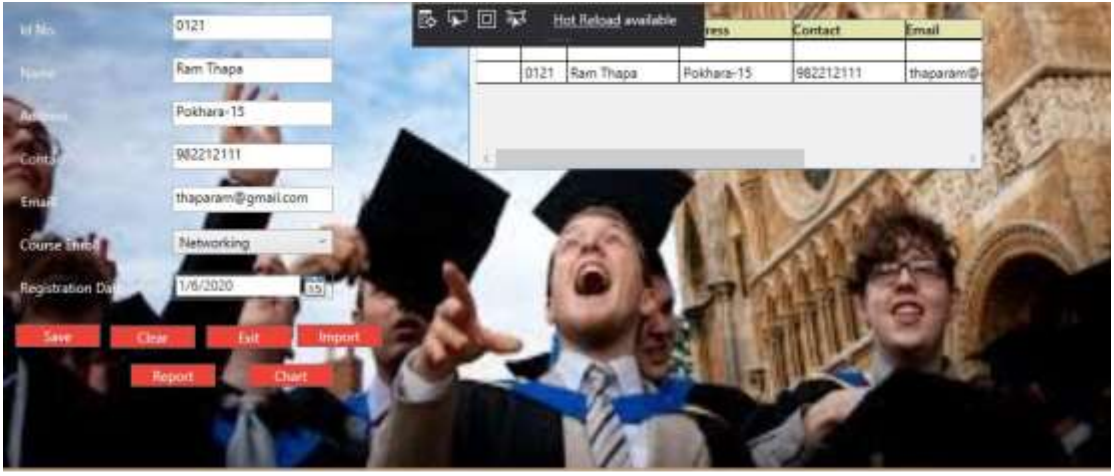
Course Enrol

Registration Da

1/6/2020

			ress	Contact	Email
	0121	Ram Thapa	Pokhara-15	982212111	thaparam@

Figure 6: Clearing Data


Id No. 0121

Name Ram Thapa

Address Pokhara-15

Contact 982212111

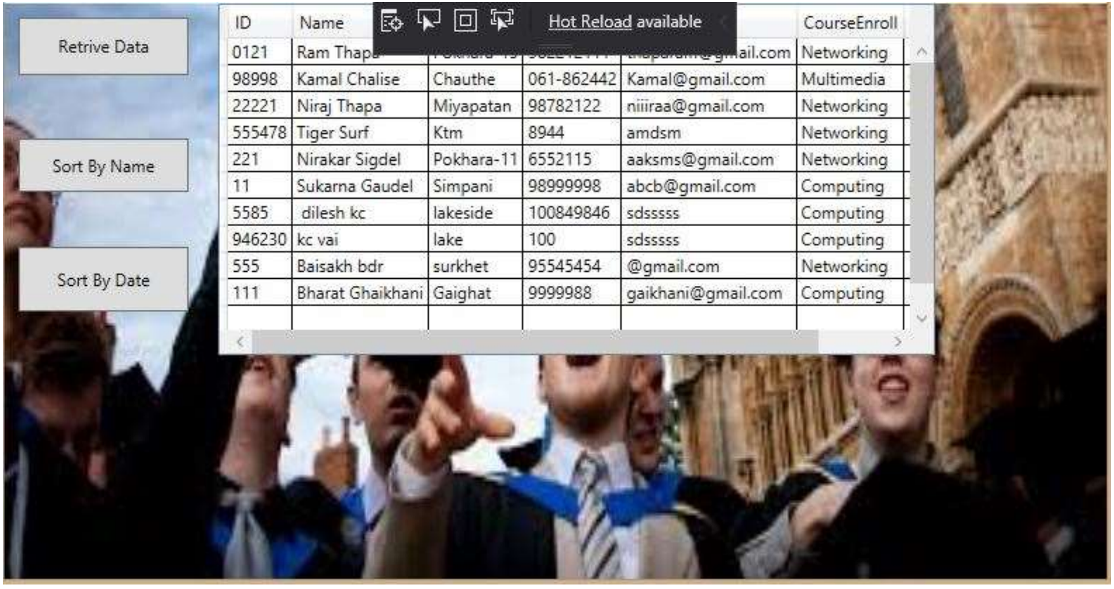
Email thaparam@gmail.com

Course Enrol Networking

Registration Da 1/6/2020

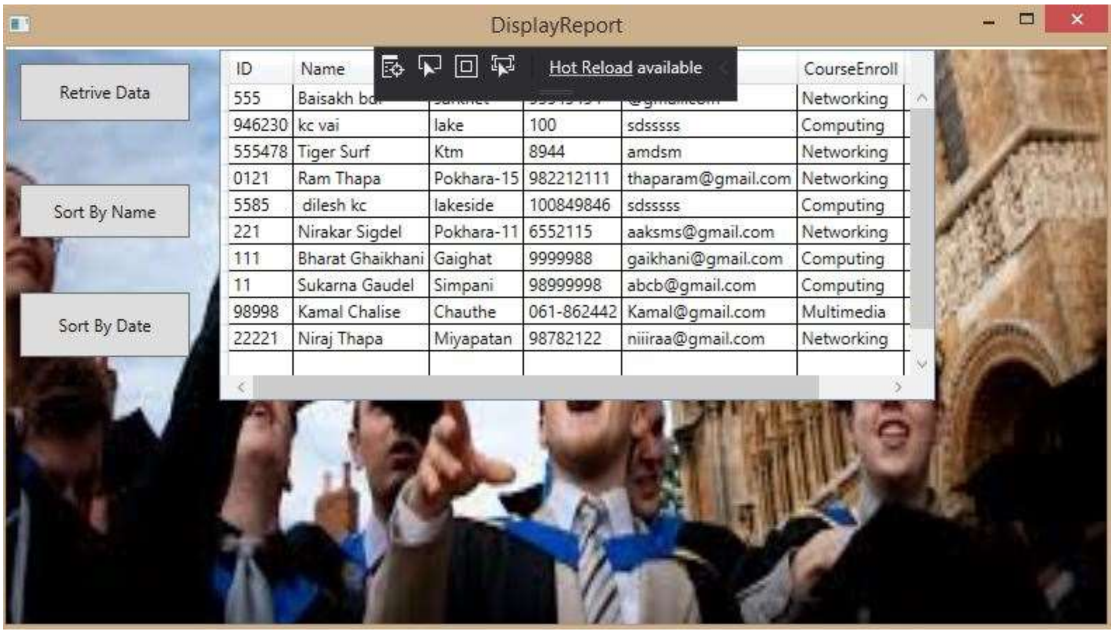
			ress	Contact	Email
	0121	Ram Thapa	Pokhara-15	982212111	thaparam@

Figure 7: Importing Data



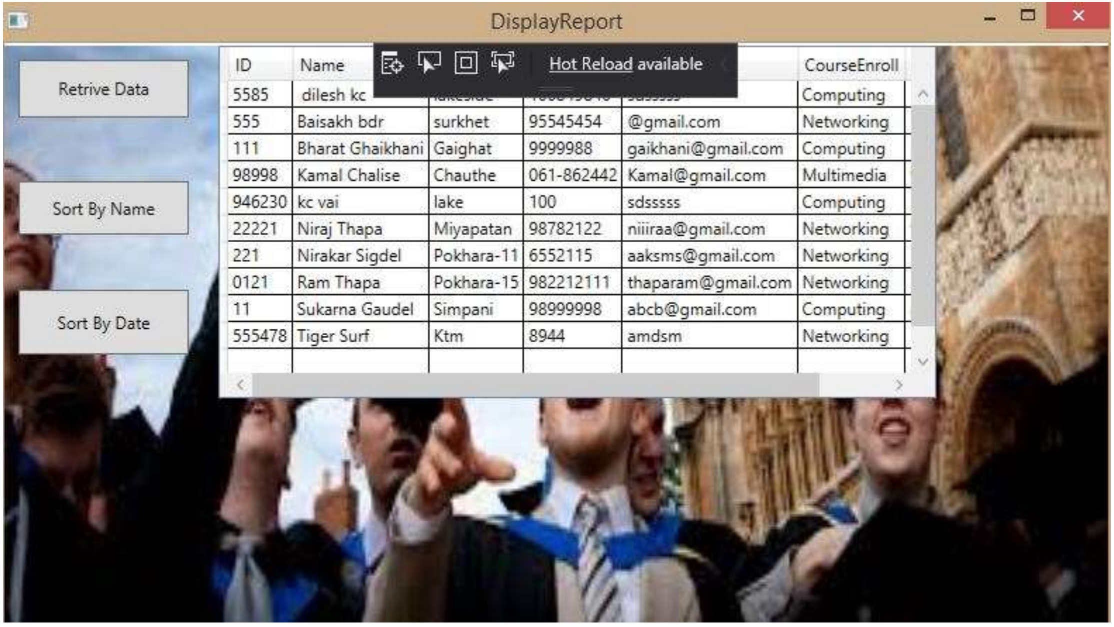
ID	Name				CourseEnroll
0121	Ram Thapa			ail.com	Networking
98998	Kamal Chalise	Chauthe	061-862442	Kamal@gmail.com	Multimedia
22221	Niraj Thapa	Miyapatan	98782122	niiiraa@gmail.com	Networking
555478	Tiger Surf	Ktm	8944	amdsm	Networking
221	Nirakar Sigdel	Pokhara-11	6552115	aaksms@gmail.com	Networking
11	Sukarna Gaudel	Simpani	98999998	abcb@gmail.com	Computing
5585	dilesh kc	lakeside	100849846	sdsssss	Computing
946230	kc vai	lake	100	sdsssss	Computing
555	Baisakh bdr	surkhet	95545454	@gmail.com	Networking
111	Bharat Ghaikhani	Gaighat	9999988	gaikhani@gmail.com	Computing

Figure 8:Retrieve Data



ID	Name				CourseEnroll
555	Baisakh bd	[illegible]	[illegible]	[illegible]	Networking
946230	kc vai	lake	100	sdsssss	Computing
555478	Tiger Surf	Ktm	8944	amdsm	Networking
0121	Ram Thapa	Pokhara-15	982212111	thaparam@gmail.com	Networking
5585	dilesh kc	lakeside	100849846	sdsssss	Computing
221	Nirakar Sigdel	Pokhara-11	6552115	aaksms@gmail.com	Networking
111	Bharat Ghaikhani	Gaighat	9999988	gaikhani@gmail.com	Computing
11	Sukarna Gaudel	Simpani	98999998	abcb@gmail.com	Computing
98998	Kamal Chalise	Chauthe	061-862442	Kamal@gmail.com	Multimedia
22221	Niraj Thapa	Miyapatan	98782122	niiiraa@gmail.com	Networking

Figure 9: Data sort by date



ID	Name				CourseEnroll
5585	dilesh kc	[illegible]	[illegible]	[illegible]	Computing
555	Baisakh bdr	surkhet	95545454	@gmail.com	Networking
111	Bharat Ghaikhani	Gaighat	9999988	gaikhani@gmail.com	Computing
98998	Kamal Chalise	Chauthe	061-862442	Kamal@gmail.com	Multimedia
946230	kc vai	lake	100	sdsssss	Computing
22221	Niraj Thapa	Miyapatan	98782122	niiiraa@gmail.com	Networking
221	Nirakar Sigdel	Pokhara-11	6552115	aaksms@gmail.com	Networking
0121	Ram Thapa	Pokhara-15	982212111	thaparam@gmail.com	Networking
11	Sukarna Gaudel	Simpani	98999998	abcb@gmail.com	Computing
555478	Tiger Surf	Ktm	8944	amdsm	Networking

Figure 10:Data sort by date

After clearing the form we can see empty there and importing the data by clicking the button import. Then after going in the report button we can see the total report displayed. Clicking report button we have a form consisting with Retrieve Data button, Sort By Name button, Sort By Date button and the data is shown respectively.

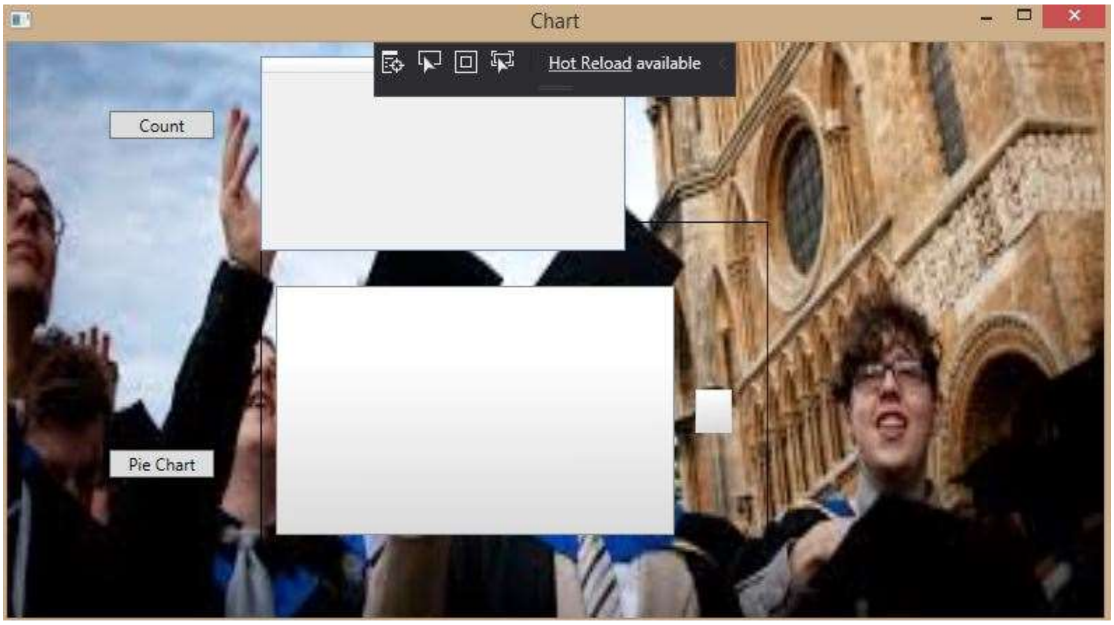
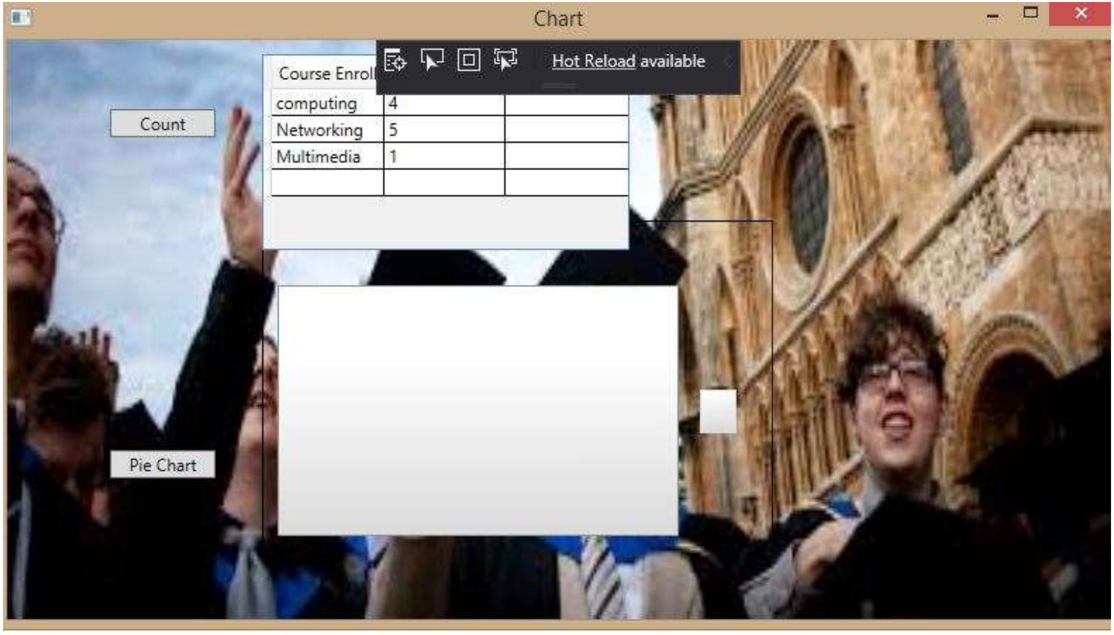
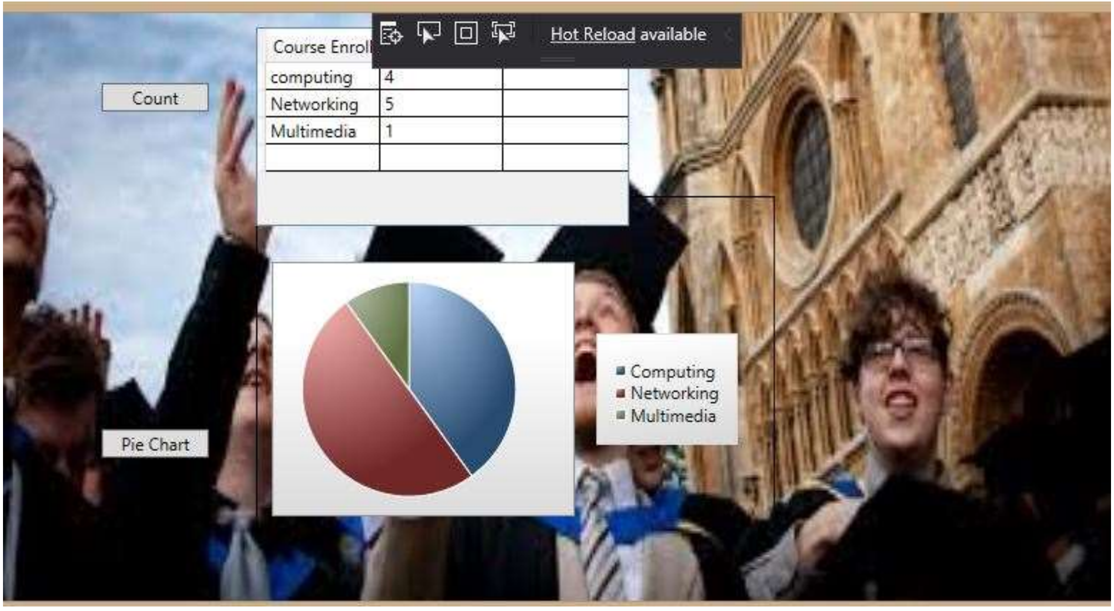


Figure 11: Chart Display



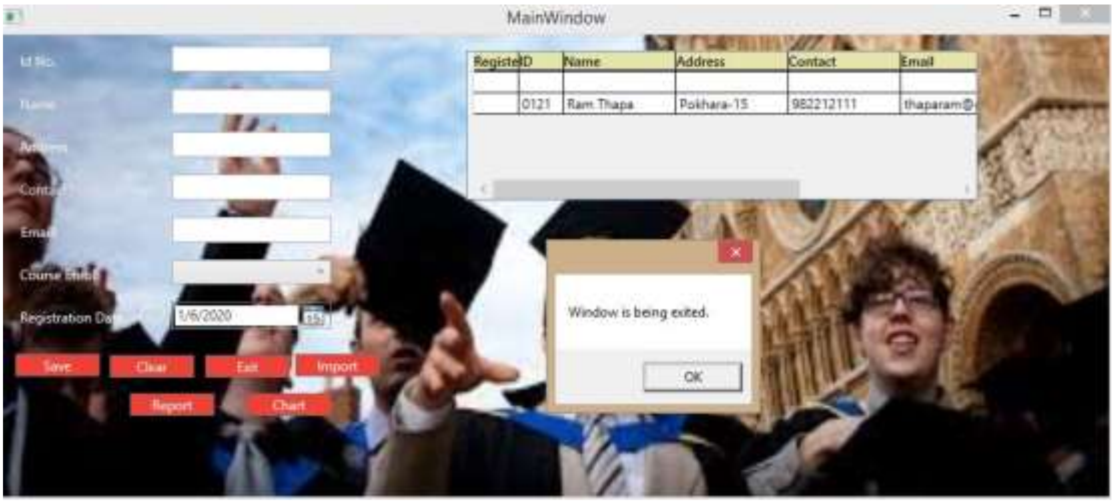
Course Enroll		
computing	4	
Networking	5	
Multimedia	1	

Figure 12: Chart on count



Course Enroll		
computing	4	
Networking	5	
Multimedia	1	

Figure 13: Pie Chart Display



MainWindow

RegisterID	Name	Address	Contact	Email
0121	Ram Thapa	Pokhara-15	982212111	thaparam@

Registration Date: 1/6/2020

Window is being exited.

Figure 14: Exit Successfully

We can see the Chart button in the Main page after clicking in button there will be display of pie chart with the recorded data shading. And at last the programme is exited after clicking in Exit button.

3. Journal Articles

I. Developments in information technologies have been impacting upon educational organizations. Principals have been using management information systems to improve the efficiency of administrative services. The aim of this research is to explore principals' perceptions about management information systems and how school management information systems are used in primary schools. The respondents of this study were 98 elementary school principals in Edirne. Data were gathered using a five-part questionnaire. The first part collected demographic information about respondents. The others had statements about school management information systems. The data were analysed using frequency, percentage, mean and standard deviation. Results indicated that although technologic infrastructures of elementary schools are insufficient, school management information systems have an important contribution to school management (DEMIR, n.d.).

II. The mission of the Student Information Management system is to create an integrated information technology environment for students, HOD, faculty, staff and administration. Our goal is to focus on services and integration for end users. It is a web based self-service environment for students, prospective students, and employees; an administrative transaction processing environment for yearly admissions; an informative environment for all levels of faculty and staff to do reporting, data extraction and information analysis. It is mainly useful for educational establishments to manage student data which also facilitates all individual associated information for easier navigation on daily basis. It provides capabilities for entering student test and other assessment scores, building student schedules, tracking student attendance and managing many other student-related data needs in a college. Our easy-to-use, integrated college administration application would be used to reduce time spent on administrative tasks, as to concentrate on other skilful practical activities other than book worming. It can accept, process and generate reports at any given point of time accurately. (BUDHRANI, n.d.)

III. Student Management System is essential for an institution or to a college or to a university, which utilizes computer, also which reduces manpower. Student Management System manages several student details like USN, student attendance, internal assessment marks, parent name, phone number, email-id, date-of-birth, class, sex etc. The goal of evolving this application is to induce the report regarding attendance at the completion of the conclave or at the middle of the conclave. Also it is possible to get the average of internal assessments and it is easy to get the report at the end. Student’s and faculty’s details uploaded by the admin. He will give username and passwords to the respective. Faculty will update the student status by putting present or absent. Suppose if particular student is absent, the message will be sent to the respective parent and email will be sent to parent. Finally student can only view his details, he can take the report. Student Management System has four modules. Initially admin will login, login module. Later he is going to upload the details of student, called student data module which has the functionalities like searching, inserting, updating and deleting the student data. At the end of the session report will be generated, called report module which is generated in the pdf format. If particular student is absent his status will be sent to the parent by a SMS, called SMS module, and email will be sent to the respective parent and it is known as email module. (RAMAKRISHNA, 2017)

IV. With advanced technology, it seems, comes more and more paperwork. Ironically, this is exactly the opposite of the "paperless society" predicted for years. Fortunately for educators, there are some products available right now that help ease paperwork-insensitive tasks. Keeping detailed records of each and every student's grades, class schedules and personal information - including medical, parental and disciplinary information - has long been the bane of educators everywhere. Now add the monumental task of keeping all such records updated district-wide. And every year seems to bring even more regulations and laws that, in turn bring even more paperwork. To keep track of this plethora of data, software firms have responded with a wide range of Student Information System (SIS) programs, gradebook software and miscellaneous administrative packages. (WILLIAM, 2019)

V. This paper presents a practitioner's view on student system implementations in the Australian university sector. A student information system is a core system of any university and integral to its operations and services to students. These systems are constantly on the list of major projects and at any point in time, a university is either implementing a new system or upgrading an existing one, or planning for either or both. These projects are costly, time-consuming and share common challenges that can be attributed to a combination of factors including software implementations, peculiarities of the individual institution, the sector, the software supplier and the environment in which it operates. Themes underpinning these challenges are explored and discussed with a view to creating greater understanding of the many facets that come into play. Questions are posed on future needs and directions given the challenges ahead, particularly the major sector reforms. (MUKERJEE, n.d.)

4. System Architecture

Algorithms of Report

Weekly Report

Steps:

1. Start

2. Check the student details.

3. If it doesn’t exists, display error message and restart

4. If exists, read the file

5. Find the data count

6. Display report

7. Retrieve the data

8. Show Report

9. Display the data in chart

10. Stop

Flowchart Diagram

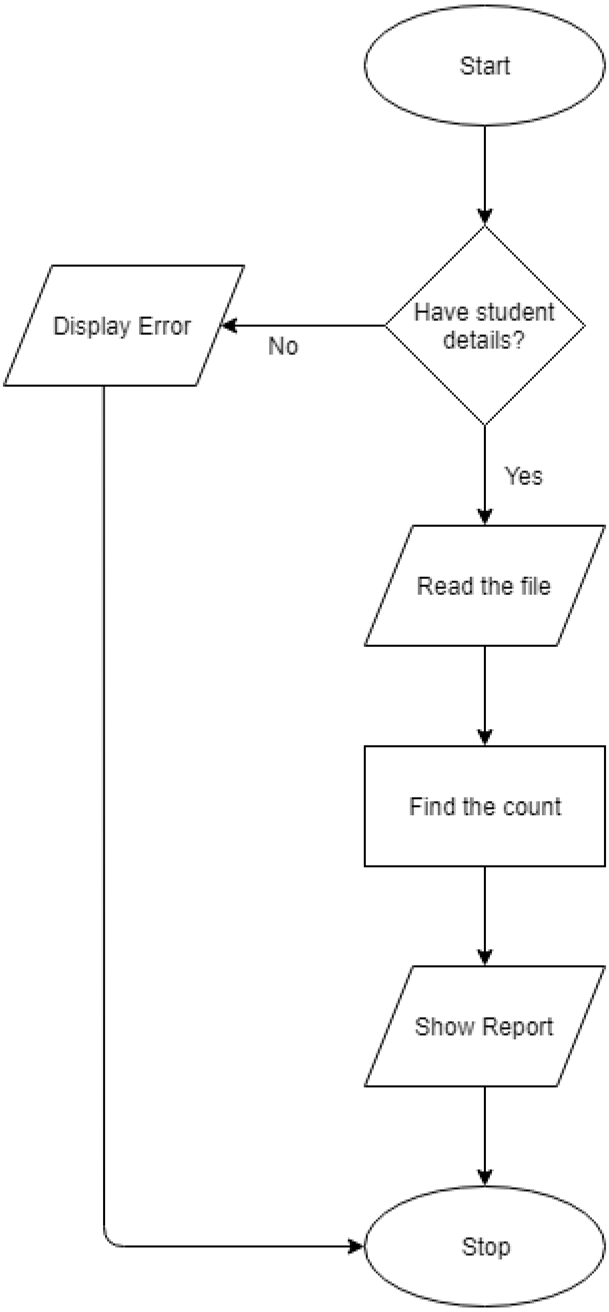


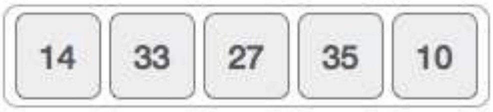
Figure 15: Diagram of FlowChart

5. Sorting Algorithm

Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of O(n^2) where **n** is the number of items.

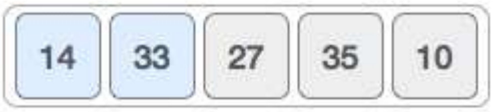
Working Mechanism:

We take an unsorted array for our example. Bubble sort takes O(n^2) time so we're keeping it short and precise.



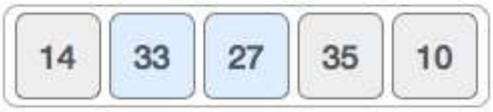
14	33	27	35	10

Bubble sort starts with very first two elements, comparing them to check which one is greater.



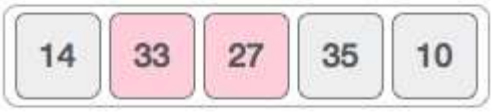
14	33	27	35	10

In this case, value 33 is greater than 14, so it is already in sorted locations. Next, we compare 33 with 27.



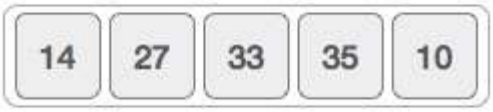
14	33	27	35	10

We find that 27 is smaller than 33 and these two values must be swapped.



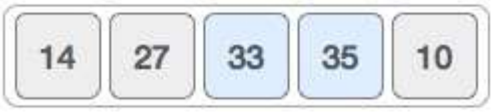
14	33	27	35	10

The new array should look like this −

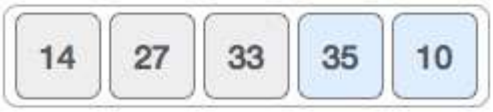


14	27	33	35	10

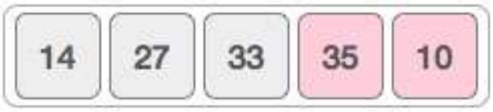
Next we compare 33 and 35. We find that both are in already sorted positions.



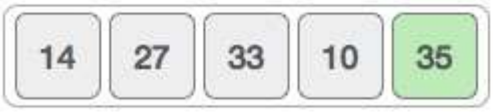
Then we move to the next two values, 35 and 10.



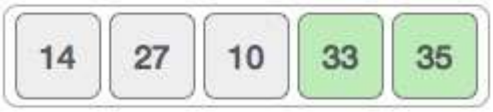
We know then that 10 is smaller 35. Hence they are not sorted.



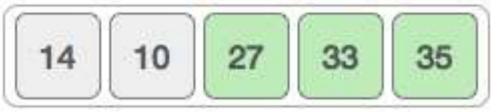
We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this −



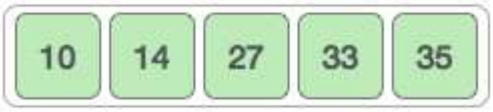
To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this −



Notice that after each iteration, at least one value moves at the end.



And when there's no swap required, bubble sorts learns that an array is completely sorted.



Now we should look into some practical aspects of bubble sort for the process (tutorialspoint.com, n.d.).

6. Reflection

The created system of is Digitalized Student Information System. It is created utilizing Visual Studio 2017 with the C# language variant 7.3. The business rational is used in the system that reflects the working environment of Student Information System. The GUI structured is profoundly UI interface and user with essential system administration can operate the system.

An end user will have the facilities of add the In Time and Out Time of the traveller together with their name. The total period spend via the visitor is calculated automatically by using the system. The info of the entered visitor can be brought such as: Last Name, telephone number, email deal with and occupation can be delivered manually by using the user. In addition to that, a user can test out the weekly chart at the side of the list.

I had a few previous experience with Visual Studio. With this revel in I had got some plus point whilst doing this coursework. I came to get more operating and enjoy with the language. Features like creating chart generating listing In addition to that, sorting of records form the grid was a new thing for me. Furthermore, import and exporting to CSV document changed into new aspect for me. Overall assessment with the great help I had a terrific revel in with the Application Development of the Student Information System.

7. Conclusion

As per the scenario, a desktop application for Student Information System was developed in C# programming language. It takes a deep research and time to build up the task in Visual Studio utilizing C# programming dialect. The framework has login screen which will give privacy and security to user. After login, the desktop application has a primary screen where user can save the data of enrolled student. I have been more familiar to the C# programming by doing this coursework.

I am grateful to my module leader Mr. Ishwor Sapkota for introducing the programming and helping us with great gratitude and would like to thank Mr. Sachin Subedi sir for guiding us throughout the project.

8. Bibliography

BUDHRANI, D. (n.d.). Retrieved from https://www.ijedr.org/papers/IJEDR1801002.pdf

DEMIR, k. (n.d.). Retrieved from https://files.eric.ed.gov/fulltext/EJ1102477.pdf

MUKERJEE, S. (n.d.). Retrieved from https://www.researchgate.net/publication/233211813_Student_information_systems_-_implementation_challenges_and_the_road_ahead

RAMAKRISHNA, D. M. (2017). Retrieved from http://irjcs.com/volumes/vol4/iss05/46.MYCSSP10083.pdf

tutorialspoint.com. (n.d.). Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm

WILLIAM, W. (2019). Retrieved from https://www.questia.com/library/journal/1G1-18213264/student-information-systems-are-integrating-more-functions

9. Appendix

```
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
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 ApplicationDevelopmentCW
{

    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            Student student = new Student();

            DataGridXAML.Items.Add(student);
        }
        public class Student
        {
            public string ID
            {
                get; set;
            }

            public string Name { get; set; }

            public string Address { get; set; }

            public string Contact { get; set; }

            public string CourseEnroll { get; set; }

            public string RegDate { get; set; }

            public string Email { get; internal set; }
        }
        private void btnSave_Click(object sender, RoutedEventArgs e)
        {

            //empty input validation
            if (textId.Text == "")
            {
                MessageBox.Show("Empty Id!");

            }
            else if (textName.Text == "")
            {
                MessageBox.Show("Name is required");
            }


            else if (textAddress.Text == "")
            {
                MessageBox.Show("Empty Address!");

            }
            else if (textContact.Text == "")
            {
                MessageBox.Show("Empty Contact Number!");

            }
            else if (textEmail.Text == "")
            {
                MessageBox.Show("Empty email id!");

            }

            else if (textCourse.Text == "")
            {
                MessageBox.Show("Invalid Level!");

            }



            else
            {
                var handler = new DataHandler();
                var dataSet = handler.CreateDataSet();
                AddSampleData(dataSet);
                MessageBox.Show("Data saved successfully !!!");
                if (File.Exists(@"D:\student.xml"))
                {
                    dataSet.ReadXml(@"D:\student.xml");
                    dataSet.WriteXml(@"D:\student.xml");
                }
                else
                {
                    dataSet.WriteXml(@"D:\student.xml");
                }
            }



        }
        private void AddSampleData(DataSet dataSet)
        {
            var dr1 = dataSet.Tables["Student"].NewRow();
            dr1["ID"] = textId.Text;
            dr1["Name"] = textName.Text;
            dr1["Address"] = textAddress.Text;
            dr1["Contact"] = textContact.Text;
            dr1["Email"] = textEmail.Text;
            dr1["CourseEnroll"] = textCourse.Text;
            string text = textDate.Text;
            dr1["RegDate"] = text;
            dataSet.Tables["Student"].Rows.Add(dr1);
        }

        private void btnImport_Click(object sender, RoutedEventArgs e)
        {
            if (textId.Text == "")
            {
                MessageBox.Show("Empty ID!");

            }
            else if (textName.Text == "")
            {
                MessageBox.Show("Name is required");
            }


            else if (textAddress.Text == "")
            {
                MessageBox.Show("Empty Address!");

            }
            else if (textContact.Text == "")
            {
                MessageBox.Show("Empty Contact Number!");

            }
            else if (textEmail.Text == "")
            {
                MessageBox.Show("Empty email id!");

            }

            else if (textCourse.Text == "")
            {
                MessageBox.Show("Invalid Course!");

            }

            else
            {
                Student dataStudent = new Student();
                dataStudent.ID = textId.Text;
                dataStudent.Name = textName.Text;
                dataStudent.Address = textAddress.Text;
                dataStudent.Contact = textContact.Text;
                dataStudent.Email = textEmail.Text;
                dataStudent.CourseEnroll = textCourse.Text;
                dataStudent.RegDate = textDate.Text;

                DataGridXAML.Items.Add(dataStudent);
            }
        }

        private void btnClear_Click(object sender, RoutedEventArgs e)
        {
            textId.Clear();
            textName.Clear();
            textAddress.Clear();
            textContact.Clear();
            textCourse.SelectedIndex = -1;
            textEmail.Clear();

        }
        private void btnExit_Click(object sender, RoutedEventArgs e)
        {
            MessageBox.Show("Window is being exited.");
            this.Close();
        }

        private void btnReport_Click(object sender, RoutedEventArgs e)
        {
            DisplayReport displayReport = new DisplayReport();
            displayReport.Show();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            Chart chart = new Chart();
            chart.Show();

        }

    }



}



using System;
using System.Collections.Generic;
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.Shapes;

namespace ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Window
    {
        public Login()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            string username = textBox1.Text;
            string password = textBox2.Password;

            if (username == "")
            {
                MessageBox.Show("Username cannot be empty!");
            }
            else if (password == "")
            {
                MessageBox.Show("Password cannot be empty!");
            }
            else if (password == "dipesh" && username == "dipesh")
            {
                this.Hide();
                MainWindow mainWindow = new MainWindow();
                mainWindow.Show();
            }

            else
            {
                MessageBox.Show("Incorrect username or password!");
            }

        }

        private void Exit_Click(object sender, RoutedEventArgs e)
        {
            MessageBox.Show("Window is being exited.");
            this.Close();
        }
    }
}
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Controls.DataVisualization.Charting;
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 ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for Chart.xaml
    /// </summary>
    public partial class Chart : Window
    {

        public KeyValuePair<string, int>[] ItemsSource { get; private set; }

        public Chart()
        {
            InitializeComponent();

        }
        private void btnChart_click(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\student.xml");
            DataTable dtStudentReport = dataSet.Tables[0];
            int total_Computing = 0;
            int total_Networking = 0;
            int total_Multimedia = 0;

            DataTable dt = new DataTable("newTable");
            dt.Columns.Add("Course Enroll", typeof(string));
            dt.Columns.Add("Total Students", typeof(int));

            for (int i = 0; i < dtStudentReport.Rows.Count; i++)
            {
                string col = 
dtStudentReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Computing++;
                }
                else if (col == "Networking")
                {
                    total_Networking++;
                }
                else if (col == "Multimedia")
                {
                    total_Multimedia++;
                }

            }
            dt.Rows.Add("computing", total_Computing);
            dt.Rows.Add("Networking", total_Networking);
            dt.Rows.Add("Multimedia", total_Multimedia);
            DataChartGrid.DataContext = dt.DefaultView;

        }

        private void btnPieChart_Click(object sender, RoutedEventArgs e)
        {
            InitializeComponent();
            LoadPieChartData();
        }


        public void LoadPieChartData()
        {
            var dataSet = new DataSet();
            if (System.IO.File.Exists(@"D:\student.xml"))
            {
                dataSet.ReadXml(@"D:\student.xml");


                DataTable dtStdReport = dataSet.Tables[0];

                int Total_Computing = 0;
                int Total_Networking = 0;
                int Total_Multimedia = 0;

                DataTable Week = new DataTable("WeekTable1");
                Week.Columns.Add("Courses Enrolled", typeof(String));
                Week.Columns.Add("Overall Student", typeof(int));


                for (int i = 0; i < dtStdReport.Rows.Count; i++)
                {
                    String column = 
dtStdReport.Rows[i]["CourseEnroll"].ToString();

                    if (column == "Computing")
                    {
                        Total_Computing++;

                    }
                    else if (column == "Networking")
                    {
                        Total_Networking++;

                    }
                    else if (column == "Multimedia")
                    {
                        Total_Multimedia++;

                    }
                }
                Week.Rows.Add("BBA", Total_Computing);
                Week.Rows.Add("BE_IT", Total_Networking);
                Week.Rows.Add("BBS", Total_Multimedia);

                ((PieSeries) chartEnroll).ItemsSource = new 
KeyValuePair<string, int>[]{
                        new KeyValuePair<string,int>("Computing", 
Total_Computing),
                        new KeyValuePair<string,int>("Networking", 
Total_Networking),
                        new KeyValuePair<string,int>("Multimedia", 
Total_Multimedia)};

                             }
            else
            {
                MessageBox.Show("No data to show!");
            }

        }


    }
    }

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;
using System.Windows;

namespace ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Data;
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.Shapes;

namespace ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for DisplayReport.xaml
    /// </summary>
    public partial class DisplayReport : Window
    {
        DataTable buffer;

        public DisplayReport()
        {
            InitializeComponent();
        }

        private void show_data()
        {
            String dataXML = @"D:\student.xml";
            DataSet dataset = new DataSet();
            dataset.ReadXml(dataXML);

            buffer = new DataTable("dt");
            buffer.Columns.Add("ID", typeof(String));
            buffer.Columns.Add("Name", typeof(String));
            buffer.Columns.Add("Address", typeof(String));
            buffer.Columns.Add("ContactNo", typeof(String));
            buffer.Columns.Add("Email", typeof(String));
            buffer.Columns.Add("CourseEnroll", typeof(String));
            buffer.Columns.Add("RegDate", typeof(String));

            for (int i = 0; i < dataset.Tables[0].Rows.Count; i++)
            {
                string s = dataset.Tables[0].Rows[i][6].ToString();
                DateTime dtime = DateTime.Parse(s);
                buffer.Rows.Add(
                    dataset.Tables[0].Rows[i][0].ToString(),
                    dataset.Tables[0].Rows[i][1].ToString(),
                    dataset.Tables[0].Rows[i][2].ToString(),
                    dataset.Tables[0].Rows[i][3].ToString(),
                    dataset.Tables[0].Rows[i][4].ToString(),
                    dataset.Tables[0].Rows[i][5].ToString(),
                    dtime.ToShortDateString());
            }
            DataView dataV = new DataView(buffer);
            DataGridReport.ItemsSource = dataV;
        }

        private void buttonRetrive_Click(object sender, RoutedEventArgs e)
        {
            show_data();
        }

        private void buttonSName_Click(object sender, RoutedEventArgs e)
        {
            DataView dataV = new DataView(buffer);
            dataV.Sort = "Name ASC";
            DataGridReport.ItemsSource = dataV;
        }

        private void buttonSD_Click(object sender, RoutedEventArgs e)
        {
            DataView dataV = new DataView(buffer);
            dataV.Sort = "RegDate ASC";
            DataGridReport.ItemsSource = dataV;
        }


        private void buttonChart_Click(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\student.xml");
            DataTable dtStudentReport = dataSet.Tables[0];

            int total_Computing = 0;
            int total_Networking = 0;
            int total_Multimedia = 0;

            DataTable dt = new DataTable("newTable");
            dt.Columns.Add("Course Enroll", typeof(string));
            dt.Columns.Add("Total Students", typeof(int));

            for (int i = 0; i < dtStudentReport.Rows.Count; i++)
            {
                string col =
dtStudentReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Computing++;
                }
                else if (col == "Networking")
                {
                    total_Networking++;
                }
                else if (col == "Multimedia")
                {
                    total_Multimedia++;
                }

            }
            dt.Rows.Add("computing", total_Computing);
            dt.Rows.Add("Networking", total_Networking);
            dt.Rows.Add("Multimedia", total_Multimedia);
            DataGridReport.DataContext = dt.DefaultView;


        }
    }
}
```