Marking Scheme

Informatics College Pokhara

informatics

college pokhara

Application Development

CS6004NI

Course Work 1

Submitted By:	**Nirdesh Gurung**	**Submitted To:**	Ishwor Sapkota
London Met ID:	Enter ID Here		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	appropriate use of data types but missing some properties required or missing CRUD operation
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	Missing Component
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:	**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.

Tried to complete the feature and is able to code by himself.

LONDON
METROPOLITAN
UNIVERSITY

informatics
college • pokhara

Module Code & Module Title

CS6004NA Application Development

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2018-19 Spring

Name: Nirdesh Gurung

College ID: NP04CP4S180010

University ID: 17031945

Table of Contents

Table of Figures

Table of Tables

INTRODUCTION

This coursework is about Desktop application called Student Information System, which needs to be developed in C#. Visual Studio 2019 was installed for the development of the application. The application allows user to input the student personal detail including registration date. System can generate a weekly enrolment report of the student with the help of registration date. Details like Name, address, contact no, email, registration date is included as label and textbox to input their data. This application is developed to keep track of student's details, program enroll and registration date.

Application is secured with login username and password. So other person cannot use the application. If the login is successful user can input the data i.e. student's detail which includes registration number, name of student, address, contact number, email id, registration date, program enroll. User can click add button to add the detail in database. Report button will generate report form which includes display weekly report button to display total enrolled student in a week, another button name display chart which displays chart of the students according to their program enroll and lastly exit button is placed to exit the window. Similarly, according to need of the project student's detail button is placed which opens new window called student's information. It consists of four buttons enrolled student list which shows every student enrolled in a system. Another button named sorted by registration date which sorts the data according to registration date from newest to oldest. Similarly sorted by student name displays data in alphabetical order from A to Z. While working in this window we can jump into student's detail form by clicking enroll new student button. Lastly close button is kept to exit the window. Simple UI is made in visual studio so that user can easily use the application.

Current Scenario

This is 21st century, school, college, institute and several educational sectors are still using old traditional paper-Based system i.e. notebooks to keep the records of their student details. Keeping records in big registers is messy and time consuming as well as space consuming. Using this method, we need a lot of time to find out specific detail of student. There will be large number of notebooks for student detail and we have to go through every page to find out the specific detail. But using this application these tasks will be easy. Everything is well organised in digital form, within a few seconds after clicking few buttons we can search our desired detail of student.

Proposed System

The proposed framework is digitized framework which is exceptionally intended to conquer issue referenced previously. The framework guarantees security with the nearness of login area. Passage of information and show of information have been made simple with the nearness of simple UI.

USER MANUAL

Screenshots are placed below to illustrate how application works.

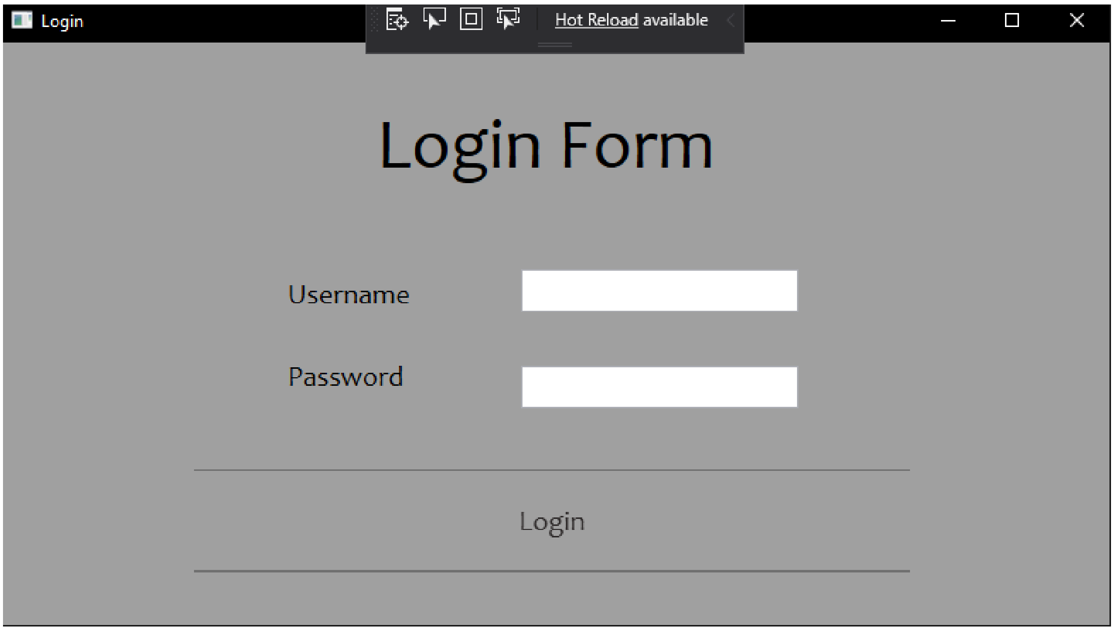


Figure 1: Login Form

First of all, login screen appears which contains label and text box for username and password. User needs to input correct username and password to get access into application. Username and password of this system is “nirdesh”.

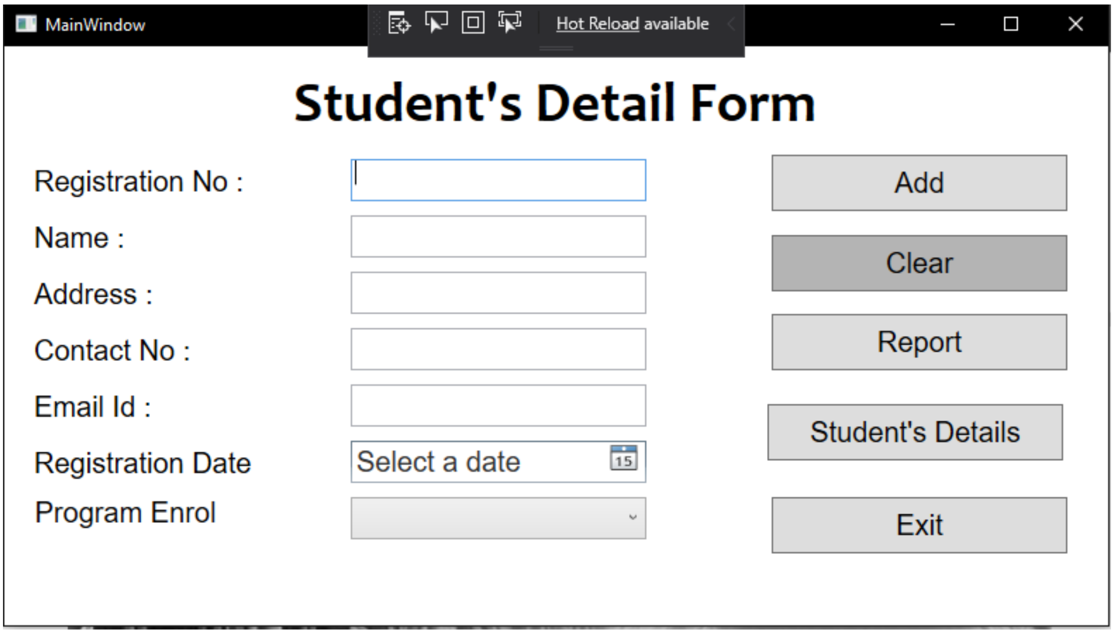
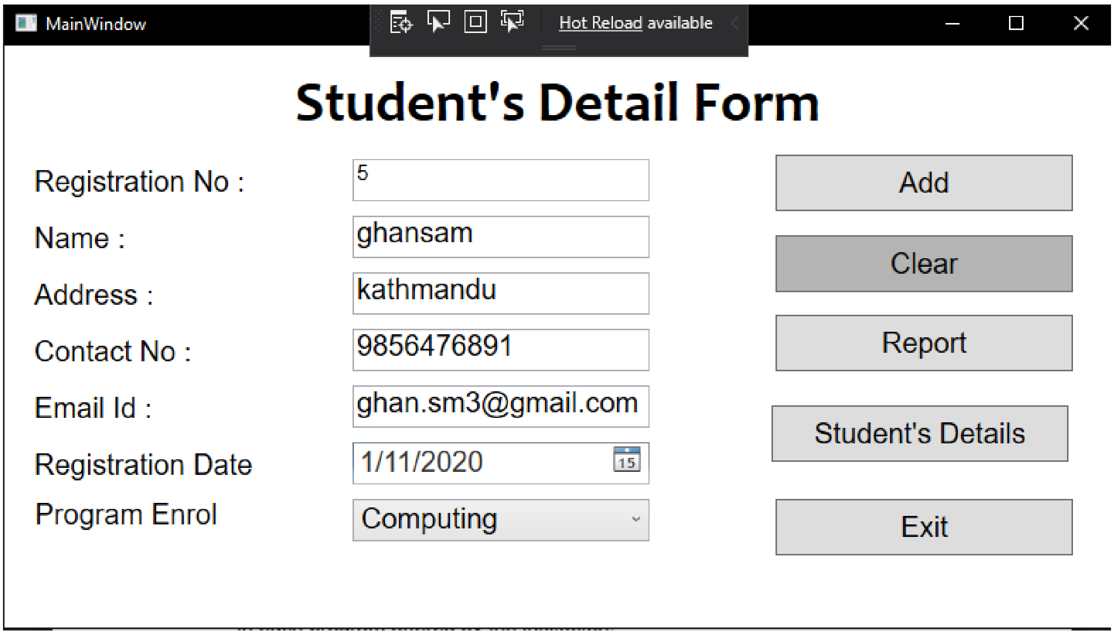


Figure 2: Student’s Detail Form

After successful login the main screen of application named “Student’s Detail Form” will appear. Here, user can input data or student’s detail. User must input registration number, name, address, contact no, email id, registration date, program enroll or a student and click add if all textbox is filled with needed data. After clicking add button a message box will appear having “Successfully added” as message.



MainWindow

Student's Detail Form

Registration No : 5

Name : ghansam

Address : kathmandu

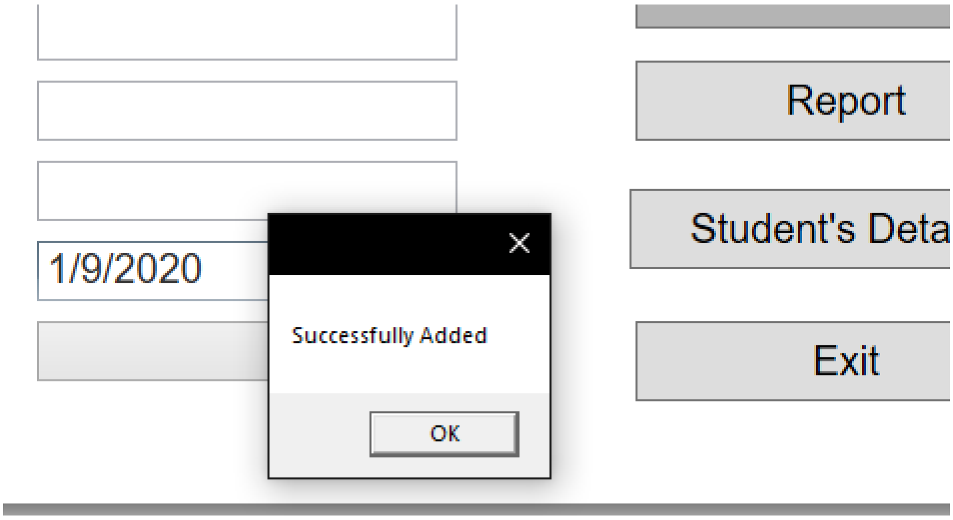
Contact No : 9856476891

Email Id : ghan.sm3@gmail.com

Registration Date 1/11/2020

Program Enrol Computing

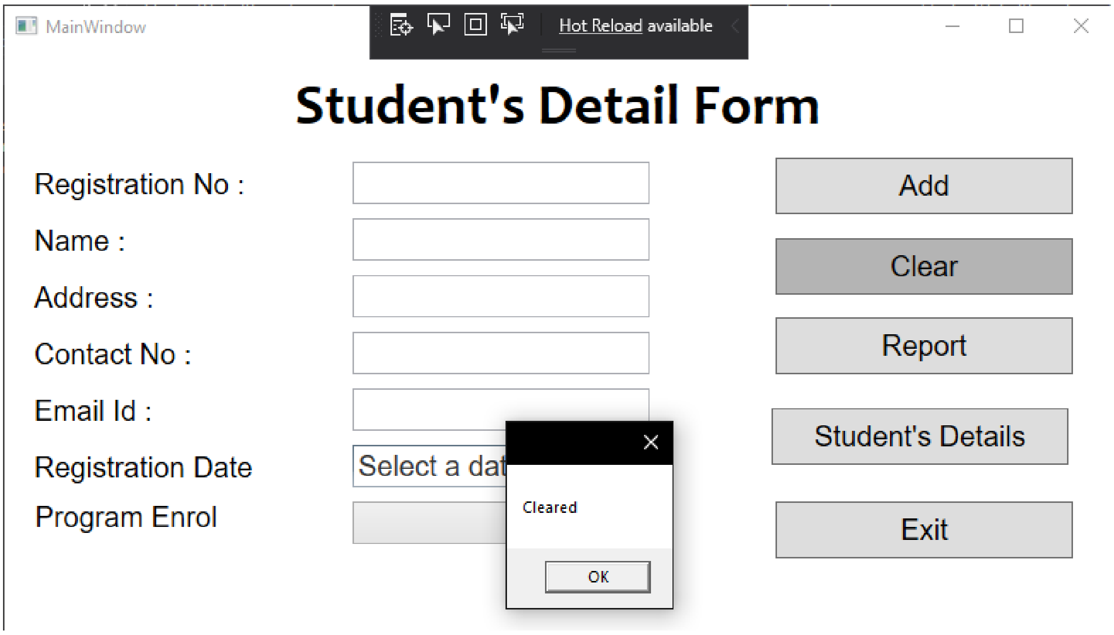
Figure 3: Inserting Data in application



1/9/2020

Successfully Added

Figure 4: Successfully added



Cleared

Figure 5: Clear Button

Clear button clears the data of textbox and message box will appear saying cleared right after click the button.

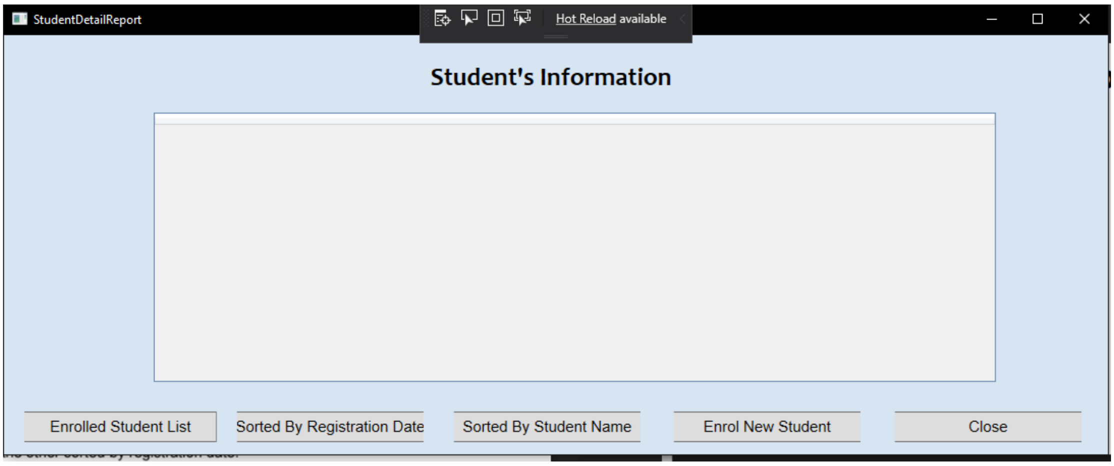
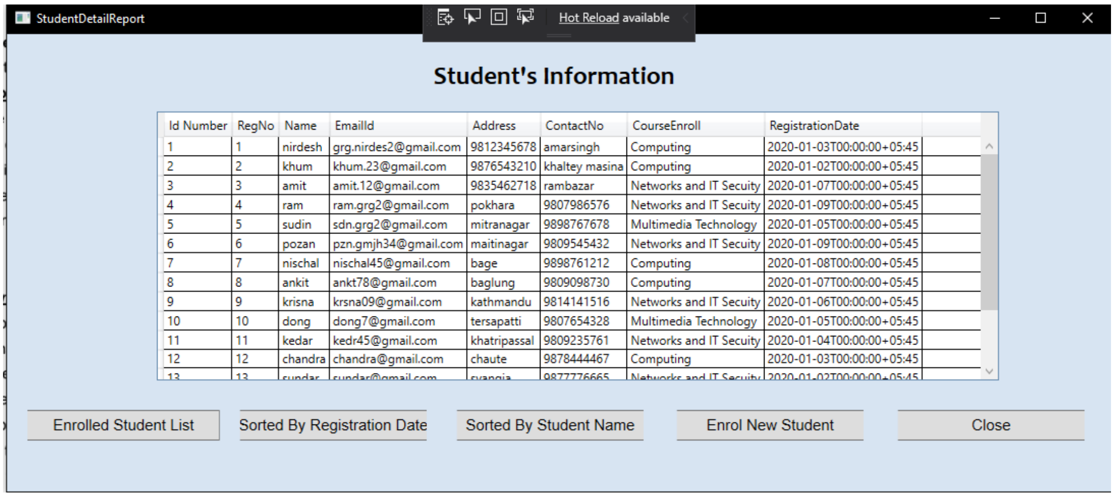


Figure 6: Student’s information

After pressing student’s detail button new window appears labeled as student’s information which consists of buttons like enrolled student list to display the total student enrolled in system.

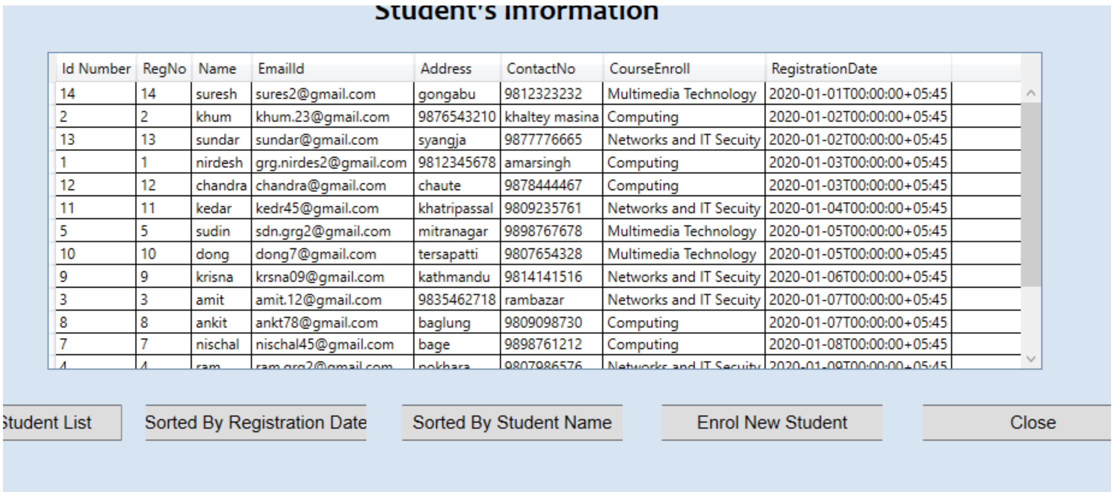


Student's Information

Id Number	RegNo	Name	EmailId	Address	ContactNo	CourseEnroll	RegistrationDate
1	1	nirdesh	grg.nirdes2@gmail.com	9812345678	amarsingh	Computing	2020-01-03T00:00:00+05:45
2	2	khum	khum.23@gmail.com	9876543210	khaltey masina	Computing	2020-01-02T00:00:00+05:45
3	3	amit	amit.12@gmail.com	9835462718	rambazar	Networks and IT Secuity	2020-01-07T00:00:00+05:45
4	4	ram	ram.grg2@gmail.com	pokhara	9807986576	Networks and IT Secuity	2020-01-09T00:00:00+05:45
5	5	sudin	sdn.grg2@gmail.com	mitranagar	9898767678	Multimedia Technology	2020-01-05T00:00:00+05:45
6	6	pozan	pzn.gmjh34@gmail.com	maitinagar	9809545432	Networks and IT Secuity	2020-01-09T00:00:00+05:45
7	7	nischal	nischal45@gmail.com	bage	9898761212	Computing	2020-01-08T00:00:00+05:45
8	8	ankit	ankt78@gmail.com	baglung	9809098730	Computing	2020-01-07T00:00:00+05:45
9	9	krisna	krsna09@gmail.com	kathmandu	9814141516	Networks and IT Secuity	2020-01-06T00:00:00+05:45
10	10	dong	dong7@gmail.com	tersapatti	9807654328	Multimedia Technology	2020-01-05T00:00:00+05:45
11	11	kedar	kedr45@gmail.com	khatripassal	9809235761	Networks and IT Secuity	2020-01-04T00:00:00+05:45
12	12	chandra	chandra@gmail.com	chaute	9878444467	Computing	2020-01-03T00:00:00+05:45
13	13	sundar	sundar@gmail.com	syangja	9877776665	Networks and IT Secuity	2020-01-02T00:00:00+05:45

Figure 7: Enrolled student list

Another button named sorted by registration date displays the student details according to registration date from oldest to newest. Date 2020-01-01 named Suresh is oldest which is in first of the table and eventually other list are displayed.

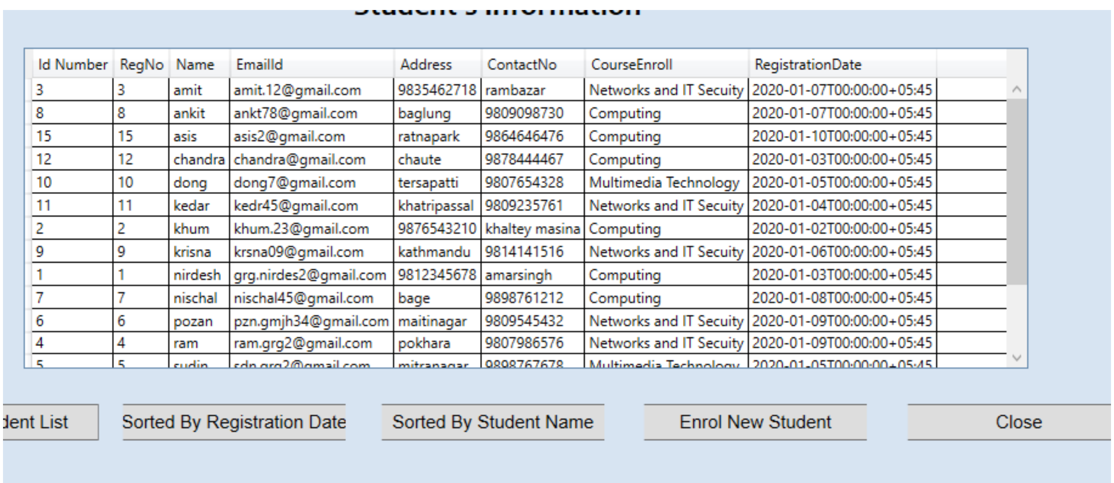


Student's Information

Id Number	RegNo	Name	EmailId	Address	ContactNo	CourseEnroll	RegistrationDate
14	14	suresh	sures2@gmail.com	gongabu	9812323232	Multimedia Technology	2020-01-01T00:00:00+05:45
2	2	khum	khum.23@gmail.com	9876543210	khaltey masina	Computing	2020-01-02T00:00:00+05:45
13	13	sundar	sundar@gmail.com	syangja	9877776665	Networks and IT Secuity	2020-01-02T00:00:00+05:45
1	1	nirdesh	grg.nirdes2@gmail.com	9812345678	amarsingh	Computing	2020-01-03T00:00:00+05:45
12	12	chandra	chandra@gmail.com	chaute	9878444467	Computing	2020-01-03T00:00:00+05:45
11	11	kedar	kedr45@gmail.com	khatripassal	9809235761	Networks and IT Secuity	2020-01-04T00:00:00+05:45
5	5	sudin	sdn.grg2@gmail.com	mitranagar	9898767678	Multimedia Technology	2020-01-05T00:00:00+05:45
10	10	dong	dong7@gmail.com	tersapatti	9807654328	Multimedia Technology	2020-01-05T00:00:00+05:45
9	9	krisna	krsna09@gmail.com	kathmandu	9814141516	Networks and IT Secuity	2020-01-06T00:00:00+05:45
3	3	amit	amit.12@gmail.com	9835462718	rambazar	Networks and IT Secuity	2020-01-07T00:00:00+05:45
8	8	ankit	ankt78@gmail.com	baglung	9809098730	Computing	2020-01-07T00:00:00+05:45
7	7	nischal	nischal45@gmail.com	bage	9898761212	Computing	2020-01-08T00:00:00+05:45
4	4	ram	ram.grg2@gmail.com	pokhara	9807986576	Networks and IT Secuity	2020-01-09T00:00:00+05:45

Figure 8: Sorted by Registration date

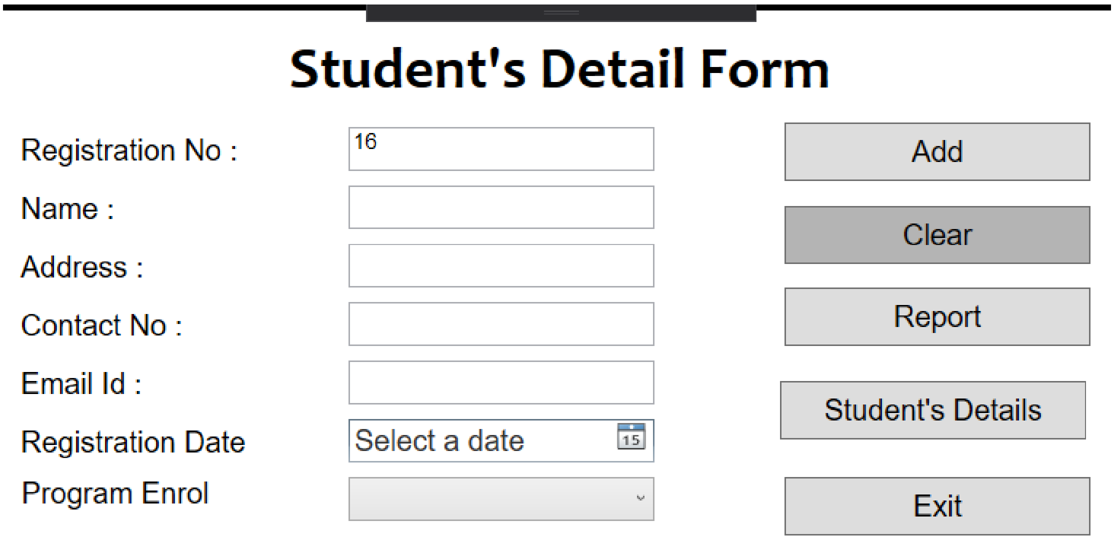
Similarly, another button sorted by student name displays the data in alphabetical order i.e. from A to Z. here name like amit, ankit which starts with A is at top of the table and eventually other names are displayed.



Id Number	RegNo	Name	EmailId	Address	ContactNo	CourseEnroll	RegistrationDate
3	3	amit	amit.12@gmail.com	9835462718	rambazar	Networks and IT Secuity	2020-01-07T00:00:00+05:45
8	8	ankit	ankt78@gmail.com	baglung	9809098730	Computing	2020-01-07T00:00:00+05:45
15	15	asis	asis2@gmail.com	ratnapark	9864646476	Computing	2020-01-10T00:00:00+05:45
12	12	chandra	chandra@gmail.com	chaute	9878444467	Computing	2020-01-03T00:00:00+05:45
10	10	dong	dong7@gmail.com	tersapatti	9807654328	Multimedia Technology	2020-01-05T00:00:00+05:45
11	11	kedar	kedr45@gmail.com	khatripassal	9809235761	Networks and IT Secuity	2020-01-04T00:00:00+05:45
2	2	khum	khum.23@gmail.com	9876543210	khaltey masina	Computing	2020-01-02T00:00:00+05:45
9	9	krisna	krsna09@gmail.com	kathmandu	9814141516	Networks and IT Secuity	2020-01-06T00:00:00+05:45
1	1	nirdesh	grg.nirdes2@gmail.com	9812345678	amarsingh	Computing	2020-01-03T00:00:00+05:45
7	7	nischal	nischal45@gmail.com	bage	9898761212	Computing	2020-01-08T00:00:00+05:45
6	6	pozan	pzn.gmjh34@gmail.com	maitinagar	9809545432	Networks and IT Secuity	2020-01-09T00:00:00+05:45
4	4	ram	ram.grg2@gmail.com	pokhara	9807986576	Networks and IT Secuity	2020-01-09T00:00:00+05:45
5	5	sudin	sdn.grg2@gmail.com	mitranagar	9898767678	Multimedia Technology	2020-01-05T00:00:00+05:45

Figure 9: Sorted by Student name

Enroll new student button takes user to student’s detail form so user can input more data.



Student's Detail Form

Registration No : 16

Name :

Address :

Contact No :

Email Id :

Registration Date Select a date

Program Enrol

Figure 10: After clicking Enroll new Student button

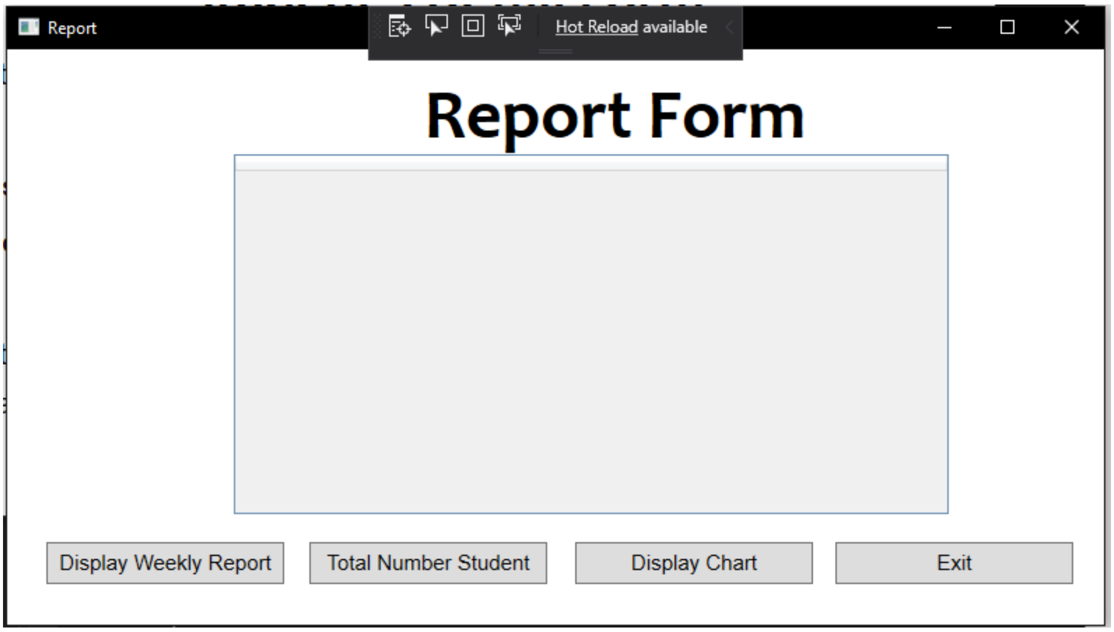
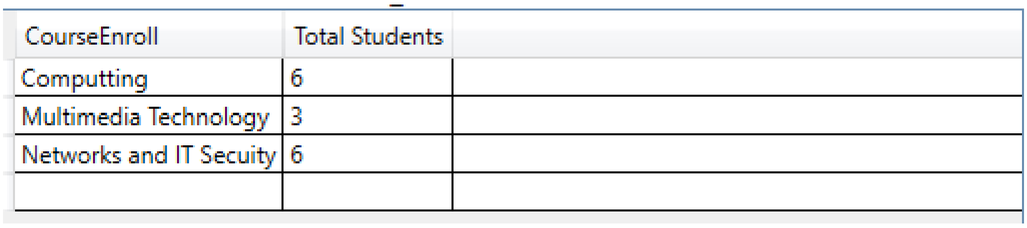


Figure 11: Report Form

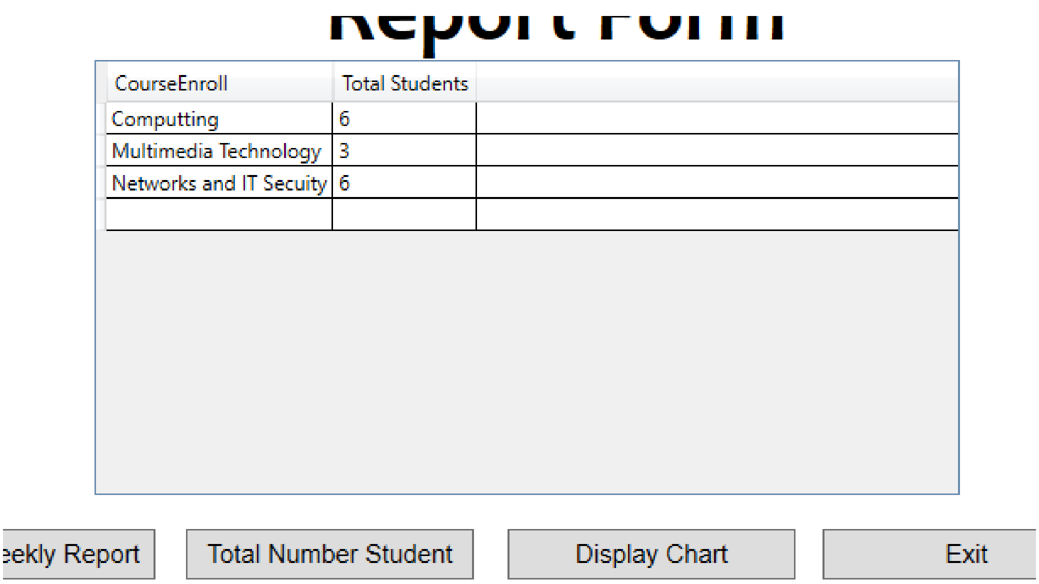
After pressing report button, window which is labeled as report form will appear which includes buttons to display weekly report, total number student, display chart and exit.



CourseEnroll	Total Students
Computting	6
Multimedia Technology	3
Networks and IT Secuity	6

Figure 12: Weekly Tabular Report

Weekly tabular report displayed by application which is mandatory. After pressing display weekly report button this table appears.



CourseEnroll	Total Students
Computting	6
Multimedia Technology	3
Networks and IT Secuity	6

Figure 13: Total student on each program

After pressing total number student button, total number of students will be displayed in each program (computing, multimedia, networking).

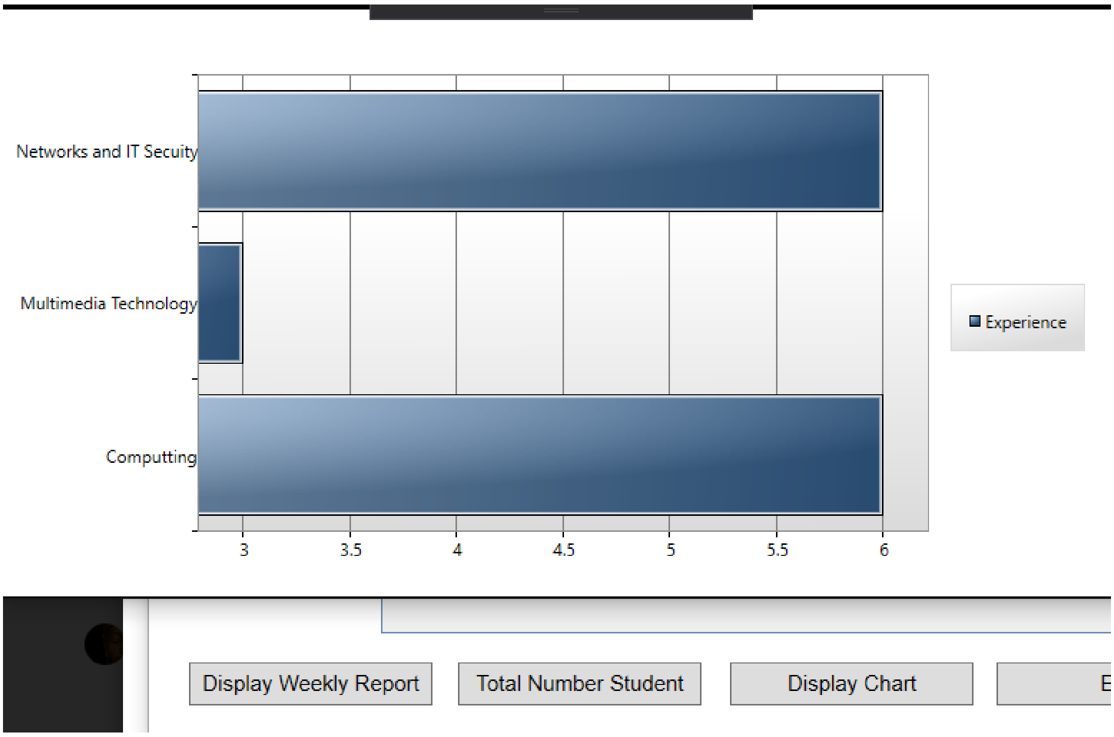


Figure 14: Chart

Chart displayed after pressing display chart button.

JOURNAL ARTICLES

1) Technology potentially provides a key component in efforts to encourage and increase class attendance. At San Diego State University, business management lecturer Kimberly King uses an integrated classroom-enhancement platform from Course Key to monitor attendance as part of her overall assessment of student performance (Walker, 2017)
2) Student records are often viewed as paperwork produced for the education bureaucracy. The most important of these is the ability to report information for decision making about individual students, schools, programs, and school districts. A second benefit, particularly with automated systems, is efficiency in processing and exchanging student records among schools. When student records are added to an overall management information system that includes information on staff, materials, and budgeting for the school or school district, more management activities can be accomplished and efficiency will be improved. Student record systems, thus, play a key role in the overall functioning of the education system; but more importantly, they increase a school's ability to meet the needs of students (nces.edu.gov, n.d.).
3) The Student Information System (SIS) would be a new way of record management and transaction processing that would achieve efficiency on processing student information. It would be a great help to the administrative personnel, academic personnel or stakeholders and students in updating, retrieving and generating student data (vijayalakshmi, 2015).
4) Today’s schools can purchase and implement administrative systems that provide easy and secure access to student records, enrolment, scheduling, and attendance; eliminate the need for duplicate data; easily integrate with other applications; and offer an array of online features for students and parents. However, this level of sophistication did not occur overnight. In making an effort to respond first to the changing needs in the K-12 student information system market, we have learned several valuable lessons (darby, 2010).

SYSTEM ARCHITECTURE

Architecture Diagram

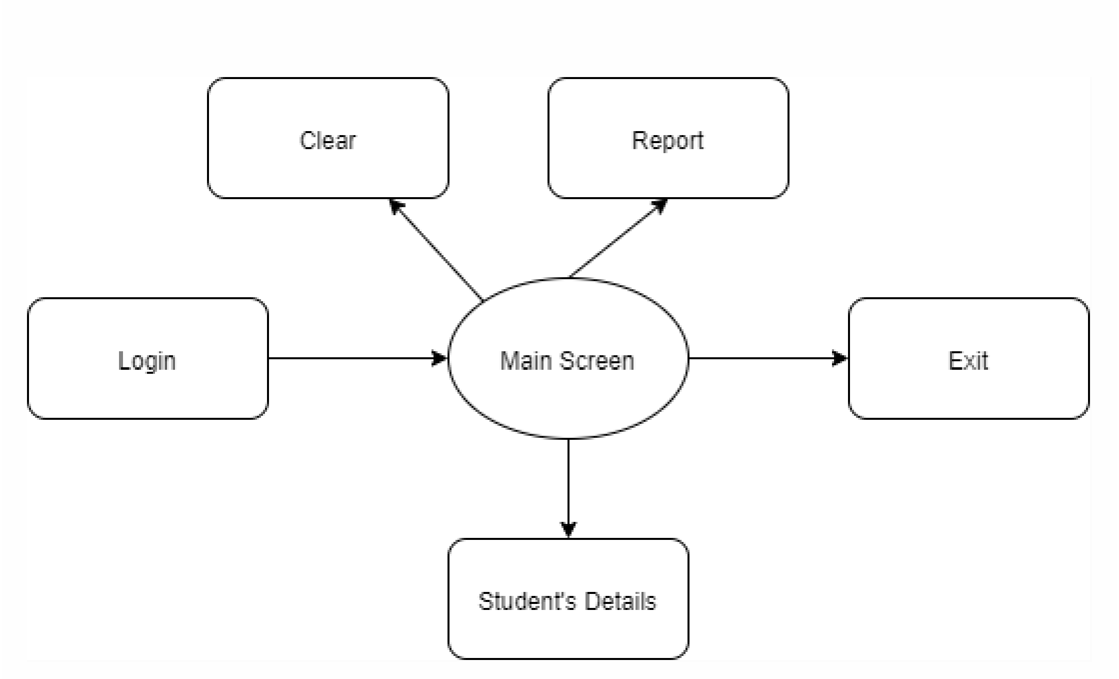


Figure 15: Architecture diagram

CLASS DIAGRAM

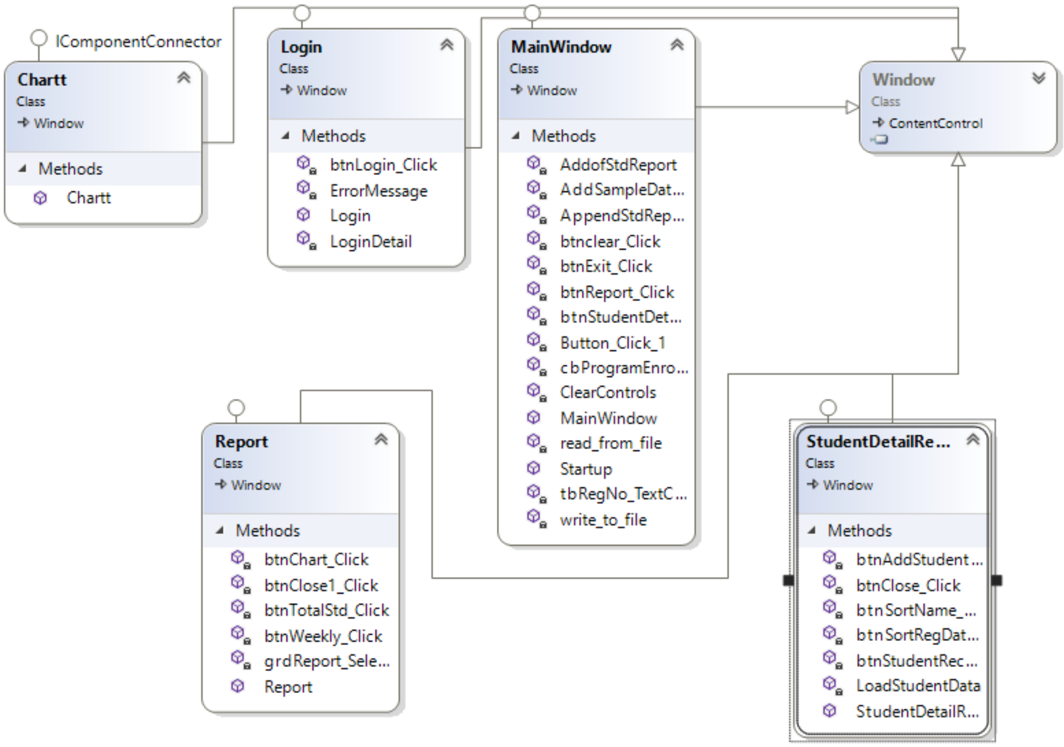


Figure 16: Class diagram

INDIVIDUAL DIAGRAM

1) Login

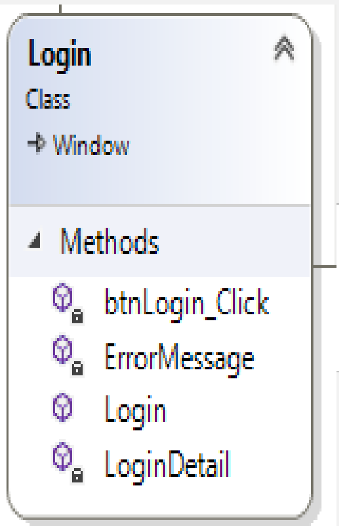
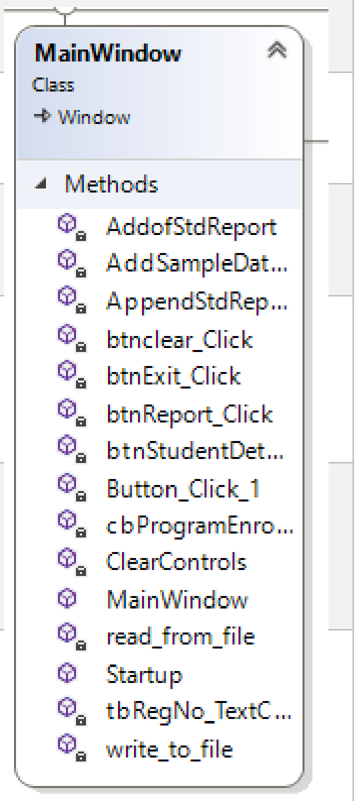
Methods	Description	Diagram
btnLogin_Click	Lets user enter in mainwindow if username and password is correct	
ErrorMessage	Displays error message if username and password is incorrect	
LoginDetail	Contains username and password of the application	

Table 1: Login Table

2) MainWindow

Methods	Description	Diagram
AddofStdReport	Contains dataset and tables	
btnclear_Click	This method clears the textbox	
btnExit_Click	This method exits the window	
btnReport_Click	This method forwards to next window called report form	
btnStudentDetails	This method forwards to student's information window	
MainWindow	This is the main window which contains labels and textbox for user input	

ClearControls	This method will clear textbox	
Read_from_file	This method will read data from file	
Write_to_file	This method will write to file	

Table 2: MainWindow Table

3) Report

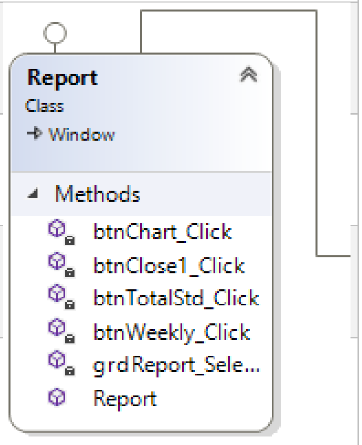
Methods	**Description**	**Diagram**
BtnChart_Click	This method generates chart	 Report Class Window Methods btnChart_Click btnClose1_Click btnTotalStd_Click btnWeekly_Click grdReport_Sele... Report
BtnClose1_Click	This method exits window	
BtnTotalStd_Click	This method displays total student	
btnWeekly_Click	This method displays Weekly enrolled student	

Table 3: Report Table

4) StudentDetailReport

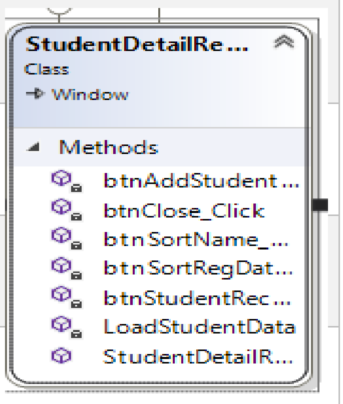
Methods	**Description**	**Diagram**
BtnClose_Click	This method exits window	 StudentDetailRe... Class Window Methods btnAddStudent... btnClose_Click btnSortName_... btnSortRegDat... btnStudentRec... LoadStudentData StudentDetailR...
BtnSortName_Click	This method arranges data in name order	
BtnSortRegDate_Click	This method arranges data in date order	
btnStudentRecord_Click	This method passes loadStudentData()	

Table 4: StudentDetailReport Table

5) Chartt

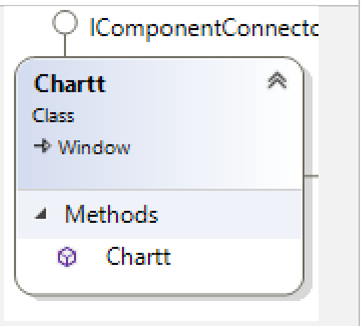
Method	Description	Diagram
Chartt	This method generates chartt	

Table 5: Chartt Table

FLOWCHART OF WEEKLY REPORT

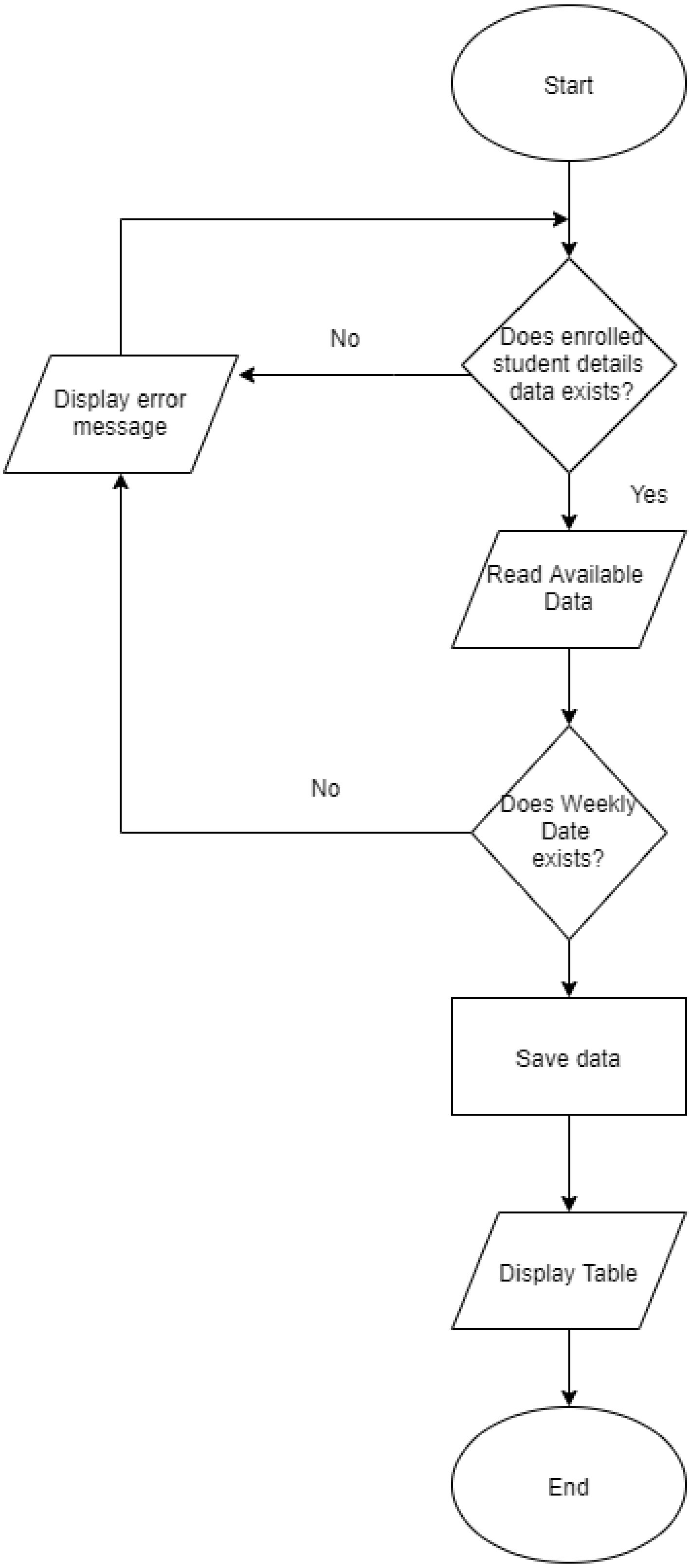


Figure 17: Flowchart of Weekly report

SORTING ALGORITHM

Bubble Sort is a simple algorithm which is used to sort a given set of n elements provided in form of an array with n number of elements. Bubble Sort compares all the element one by one and sort them based on their values.

If the given array has to be sorted in ascending order, then bubble sort will start by comparing the first element of the array with the second element, if the first element is greater than the second element, it will swap both the elements, and then move on to compare the second and the third element, and so on.

If we have total n elements, then we need to repeat this process for n-1 times. It is known as bubble sort, because with every complete iteration the largest element in the given array, bubbles up towards the last place or the highest index, just like a water bubble rises up to the water surface. Sorting takes place by stepping through all the elements one-by-one and comparing it with the adjacent element and swapping them if required.

Let's consider an array with values {5,1,6,2,4,3}

Below, we have a pictorial representation of how bubble sort will sort the given array.

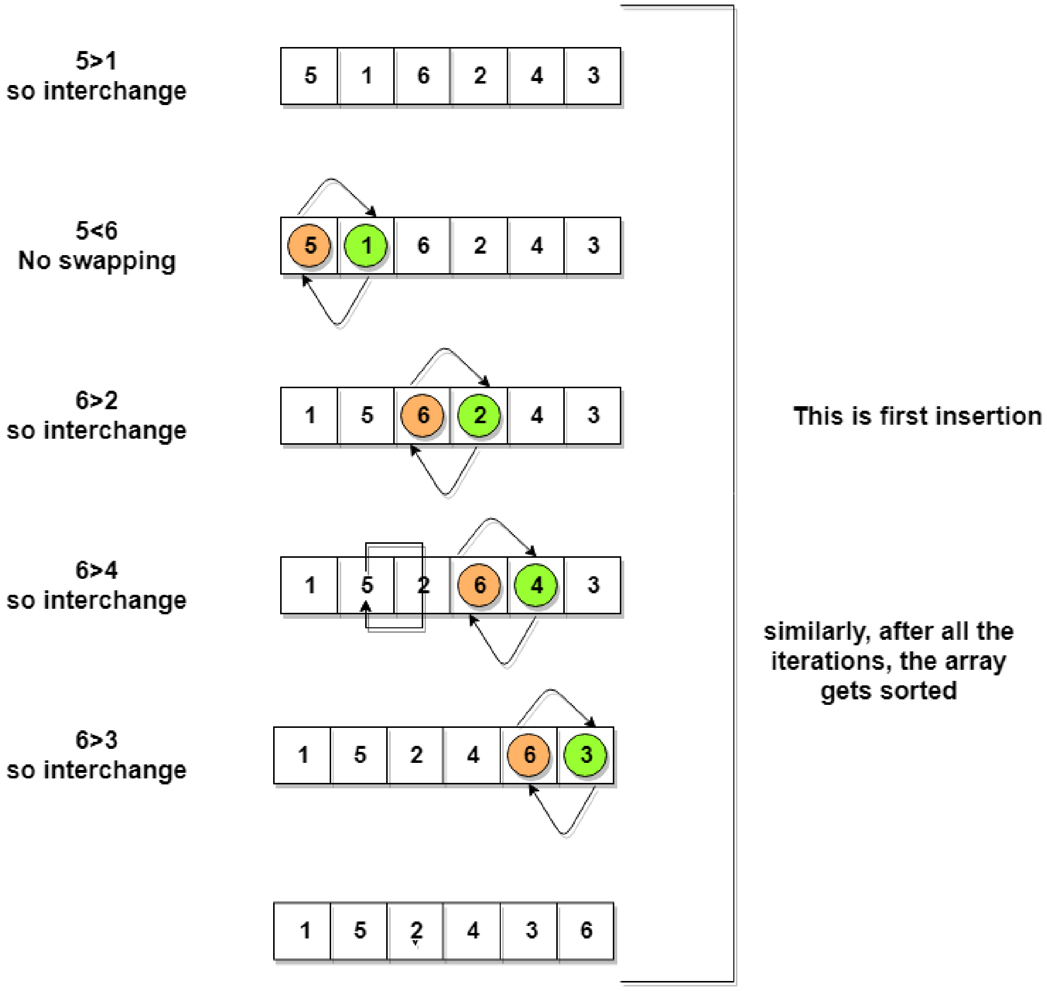


Figure 18: Bubble sort algorithm

So, as we can see in the representation above, after the first iteration, 6 is placed at the last index, which is the correct position for it. Similarly, after the second iteration, 5 will be at the second last index, and so on (studytonight, 2018).

CONCLUSION

Finally, with the proper guidance of our lecturer Mr. Ishwor Sapkota and few of my friends, coursework has come to an end. Using C# language in visual studio 2019 coursework was settled. First few weeks it was difficult to learn because it was our first time using this platform. Gradually, it became easier to perform with the helping hand of our dear sir ishwor sapkota. This coursework will polish our skill and will definitely help in near future as the system has huge demand in the market. Every schools, colleges, university and other educational institute needs this kind of application to keep record of their enrolled students. The framework has login screen to add security. After login, framework shows a screen where user can input data manually. Every functionality is placed in the main screen. UI is simple and effective so that user won't face any difficulty using the application.

BIBLIOGRAPHY

darby, r. (2010, 1 5). the evolution of student information system. *the evolution of student information system*, 2. Retrieved from https://thejournal.com/Articles/2005/10/01/The-Evolution-of-Student-Information-Systems.aspx

nces.edu.gov. (n.d.). Retrieved 1 10, 2019, from nces.edu.gov: https://nces.ed.gov/pubs2000/building/desc_system.asp

studytonight. (2018, 4 12). *studytonight*. Retrieved from studytonight: https://www.studytonight.com/data-structures/bubble-sort

vijayalakshmi. (2015). An Android Application for Student Information System. *An Android Application for Student Information System*, 4. Retrieved from http://ijarcet.org/wp-content/uploads/IJARCET-VOL-4-ISSUE-9-3615-3619.pdf

Walker, R. (2017, 10 23). *edscoop*. Retrieved from edscoop: https://edscoop.com/new-technologies-encourage-and-record-student-class-attendance-and-engagement/

APPENDIX

MainWindow.xaml.cs

```
using DataHandler;
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.Navigation;
using System.Windows.Shapes;
using System.Data;
using System.IO;

namespace StudentInformationSystems
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            Startup();

            tbRegNo.Text = read_from_file();
        }

        public void Startup()
        {

        }



        private void AddSampleDataforStd(DataSet dataSet)
        {

            var dr = dataSet.Tables["Course"].NewRow();
            dr["Name"] = "BBA";
            dr["DisplayText"] = "BBA Hons";
            dataSet.Tables["Course"].Rows.Add(dr);

            var dr1 = dataSet.Tables["Student"].NewRow();
            dr1["Name"] = tbName.Text;
            dr1["Address"] = tbAddress.Text;
            dr1["EmailId"] = tbEmail.Text;
            dr1["ContactNo"] = tbContact.Text;
            dr1["CourseEnroll"] = cbProgramEnroll.Text;
            dr1["RegistrationDate"] = dpDateTime.SelectedDate;
            dataSet.Tables["Student"].Rows.Add(dr1);
        }
        private void AddofStdReport(DataSet dataSet)
        {

            var dr1 = dataSet.Tables["StudentReport"].NewRow();

dataSet.Tables["StudentReport"].ReadXml(@"D:\StudentInformationSystem\StudentRe
port.xml");

            dr1["Name"] = tbName.Text;
            dr1["Address"] = tbAddress.Text;
            dr1["EmailId"] = tbEmail.Text;
            dr1["ContactNo"] = tbContact.Text;
            dr1["CourseEnroll"] = cbProgramEnroll.Text;
            dr1["RegistrationDate"] = dpDateTime.SelectedDate;
            dataSet.Tables["StudentReport"].Rows.Add(dr1);


        }

        private void AppendStdReport(DataSet dataSet)
        {
            if (File.Exists(@"D:\StudentInformationSystem\StudentReport.xml"))
            {
                var handler = new Handler();


dataSet.Tables["StudentReport"].ReadXml(@"D:\StudentInformationSystem\StudentRe
port.xml");
                var dr2 = dataSet.Tables["StudentReport"].NewRow();
                dr2["RegNo"] = tbRegNo.Text;
                dr2["Name"] = tbName.Text;
                dr2["Address"] = tbAddress.Text;
                dr2["EmailId"] = tbEmail.Text;
                dr2["ContactNo"] = tbContact.Text;
                dr2["CourseEnroll"] = cbProgramEnroll.Text;
                dr2["RegistrationDate"] = dpDateTime.SelectedDate;

                dataSet.Tables["StudentReport"].Rows.Add(dr2);

//dataSet.Tables["StudentReport"].WriteXml(@"C:\Informatics\Coursework\Applicat
ion Development\StudentReport.xml");
            }
            else
            {

dataSet.Tables["StudentReport"].WriteXml(@"D:\StudentInformationSystem\StudentR
eport.xml");
                AppendStdReport(dataSet);
            }



        }

        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
            var handler = new Handler();
            var dataSet = handler.CreateDataSet();
            AddSampleDataforStd(dataSet);
            AppendStdReport(dataSet);

            var regno = tbRegNo.Text;
            var name = tbName.Text;


dataSet.WriteXmlSchema(@"D:\StudentInformationSystem\StudentCWSchema1.xml");
            dataSet.Tables["Student"].WriteXml(@"D:\StudentInformationSystem\"
+ name + "CWData" + regno + ".xml");

dataSet.Tables[2].WriteXml(@"D:\StudentInformationSystem\StudentReport.xml");

            write_to_file(tbRegNo.Text);

            tbRegNo.Text = read_from_file();

            ClearControls();

            MessageBox.Show("Successfully Added");
        }

        private void write_to_file(string text)
        {



System.IO.File.WriteAllText(@"D:\StudentInformationSystem\count.txt", text);


        }
        private string read_from_file()
        {
            int i = 1;
            if (File.Exists(@"D:\StudentInformationSystem\count.txt"))
            {
                string text =
File.ReadAllText(@"D:\StudentInformationSystem\count.txt");
                i = int.Parse(text.ToString());
                i = i + 1;
            }
            else
            {
                File.WriteAllText(@"D:\StudentInformationSystem\count.txt",
"text");
            }
            return i.ToString();

        }

        private void ClearControls()
        {
            tbName.Text = "";
            tbAddress.Text = "";
            tbContact.Text = "";
            tbEmail.Text = "";
            cbProgramEnroll.Text = "";
        }

        private void tbRegNo_TextChanged(object sender, TextChangedEventArgs e)
        {

        }

        private void btnclear_Click(object sender, RoutedEventArgs e)
        {
            tbRegNo.Text = "";
            tbName.Text = "";
            tbAddress.Text = "";
            tbContact.Text = "";
            tbEmail.Text = "";
            cbProgramEnroll.Text = "";

            MessageBox.Show("Cleared");

        }

        private void btnStudentDetails_Click(object sender, RoutedEventArgs e)
        {
            StudentDetailReport studentDetailReport = new
StudentDetailReport();
            studentDetailReport.Show();
        }

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

        private void btnExit_Click(object sender, RoutedEventArgs e)
        {
            this.Close();
        }

        private void cbProgramEnroll_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {

        }
    }
}
```

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 StudentInformationSystems
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Window
    {
        public Login()
        {
            InitializeComponent();
        }
        private void LoginDetail()
        {
            string user = tbUsername.Text;
            string pass = pbPassword.Password;

            if (user == "nirdesh" && pass == "nirdeshc")
            {
                this.Hide();
                MainWindow mainWindow = new MainWindow();
                mainWindow.Show();
            }
            else
            {
                ErrorMessage();
            }
        }

        private void btnLogin_Click(object sender, RoutedEventArgs e)
        {
            LoginDetail();
        }

        private void ErrorMessage()
        {
            MessageBox.Show("Invalid username or password", "Login Error",
MessageBoxButton.OK, MessageBoxImage.Error);
            tbUsername.Text = "";
            pbPassword.Password = "";

        }
    }
}
```

Report.xaml.cs

```
using DataHandler;
using System;
using System.Collections.Generic;
using System.Data;
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 StudentInformationSystems
{
    /// <summary>
    /// Interaction logic for Report.xaml
    /// </summary>
    public partial class Report : Window
    {
        public Report()
        {
            InitializeComponent();
        }

        private void btnClose1_Click(object sender, RoutedEventArgs e)
        {
            this.Close();
        }

        private void btnWeekly_Click(object sender, RoutedEventArgs e)
        {
            var handler = new Handler();
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\StudentInformationSystem\StudentReport.xml");
            DataTable dtStdReport = dataSet.Tables[0];

            int total_Computing = 0;
            int total_MultimediaTechnology = 0;
            int total_NetworksandITSecuity = 0;

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

            for (int i = 0; i < dtStdReport.Rows.Count; i++)
            {
                string col = dtStdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Computing++;
                }
                else if (col == "Multimedia Technology")
                {
                    total_MultimediaTechnology++;
                }
                else if (col == "Networks and IT Secuity")
                {
                    total_NetworksandITSecuity++;
                }

            }
            dt.Rows.Add("Computting", total_Computing);
            dt.Rows.Add("Multimedia Technology", total_MultimediaTechnology);
            dt.Rows.Add("Networks and IT Secuity", total_NetworksandITSecuity);
            grdReport.ItemsSource = dt.DefaultView;
        }

        private void btnTotalStd_Click(object sender, RoutedEventArgs e)
        {
            var handler = new Handler();
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\StudentInformationSystem\StudentReport.xml");
            DataTable dtStdReport = dataSet.Tables[0];

            int total_Computing = 0;
            int total_MultimediaTechnology = 0;
            int total_NetworksandITSecuity = 0;

            DataTable dt = new DataTable("newTable");
            dt.Columns.Add("CourseEnroll", typeof(string));
            dt.Columns.Add("Total Students", typeof(int));
            for (int i = 0; i < dtStdReport.Rows.Count; i++)
            {
                string col = dtStdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Computing++;
                }
                else if (col == "Multimedia Technology")
                {
                    total_MultimediaTechnology++;
                }
                else if (col == "Networks and IT Secuity")
                {
                    total_NetworksandITSecuity++;
                }

            }
            dt.Rows.Add("Computting", total_Computing);
            dt.Rows.Add("Multimedia Technology", total_MultimediaTechnology);
            dt.Rows.Add("Networks and IT Secuity", total_NetworksandITSecuity);
            grdReport.ItemsSource = dt.DefaultView;
        }

        private void grdReport_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {

        }

        private void btnChart_Click(object sender, RoutedEventArgs e)
        {
            Chartt chartt = new Chartt();
            chartt.Show();
        }
    }

}
```

StudentDetailReport.xaml.cs

```
using DataHandler;
using System;
using System.Collections.Generic;
using System.Data;
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 StudentInformationSystems
{
    /// <summary>
    /// Interaction logic for StudentDetailReport.xaml
    /// </summary>
    public partial class StudentDetailReport : Window
    {
        public StudentDetailReport()
        {
            InitializeComponent();

        }
        private void LoadStudentData()
        {

            if
(System.IO.File.Exists(@"D:\StudentInformationSystem\StudentReport.xml"))
            {
                var handler = new Handler();


                var dataSet = new DataSet();


dataSet.ReadXml(@"D:\StudentInformationSystem\StudentReport.xml");

                DataTable dtStdReport = new DataTable();
                dtStdReport = dataSet.Tables[0];
                grdStudentDetails.ItemsSource = dtStdReport.DefaultView;
            }

        }


        private void btnAddStudent_Click(object sender, RoutedEventArgs e)
        {
            MainWindow mainWindow = new MainWindow();
            mainWindow.Show();
        }

        private void btnStudentRecord_Click(object sender, RoutedEventArgs e)
        {
            LoadStudentData();
        }

        private void btnSortRegDate_Click(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\StudentInformationSystem\StudentReport.xml");
            DataTable DataTable = dataSet.Tables["StudentReport"];
            DataTable.DefaultView.Sort = "RegistrationDate Asc";
            grdStudentDetails.ItemsSource = DataTable.DefaultView;
        }

        private void btnSortName_Click_1(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\StudentInformationSystem\StudentReport.xml");
            DataTable DataTable = dataSet.Tables["StudentReport"];
            DataTable.DefaultView.Sort = "Name Asc";
            grdStudentDetails.ItemsSource = DataTable.DefaultView;
        }

        private void btnClose_Click(object sender, RoutedEventArgs e)
        {
            this.Close();
        }
    }
}
```

Chartt.xaml.cs

```
using DataHandler;
using System;
using System.Collections.Generic;
using System.Data;
using System.Text;
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 StudentInformationSystems
{
    /// <summary>
    /// Interaction logic for Chartt.xaml
    /// </summary>
    public partial class Chartt : Window
    {
        public Chartt()
        {
            InitializeComponent();

            var handler = new Handler();
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\StudentInformationSystem\StudentReport.xml");
            DataTable dtStdReport = dataSet.Tables[0];

            int total_Computing = 0;
            int total_MultimediaTechnology = 0;
            int total_NetworksandITSecuity = 0;

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

            for (int i = 0; i < dtStdReport.Rows.Count; i++)
            {
                string col = dtStdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Computing++;
                }
                else if (col == "Multimedia Technology")
                {
                    total_MultimediaTechnology++;
                }
                else if (col == "Networks and IT Secuity")
                {
                    total_NetworksandITSecuity++;
                }
            }
            dt.Rows.Add("Computting", total_Computing);
            dt.Rows.Add("Multimedia Technology", total_MultimediaTechnology);
            dt.Rows.Add("Networks and IT Secuity", total_NetworksandITSecuity);

            ((BarSeries) totalChart.Series[0]).ItemsSource = new
KeyValuePair<string, int>[] {
                new KeyValuePair<string, int>("Computting", total_Computing),
            new KeyValuePair<string, int>("Multimedia Technology",
total_MultimediaTechnology),
            new KeyValuePair<string, int>("Networks and IT Secuity",
total_NetworksandITSecuity)
    };
        }
    }
}
```

Handler.cs

```
using System;
using System.Data;

namespace DataHandler
{
    public class Handler
    {
        public DataSet CreateDataSet()
        {
            var ds = new DataSet();
            ds.Tables.Add(CreateCourseTable());
            ds.Tables.Add(CreateStudentTable());
            ds.Tables.Add(CreateStudentReportTable());


            return ds;
        }

        private DataTable CreateStudentTable()
        {
            var dt = new DataTable("Student");
            DataColumn dataColumn = new DataColumn("Id Number", typeof(int));
            dataColumn.AutoIncrement = true;
            dataColumn.AutoIncrementSeed = 1;
            dataColumn.AutoIncrementStep = 1;

            dt.Columns.Add(dataColumn);

            dt.Columns.Add("Name", typeof(string));
            dt.Columns.Add("Address", typeof(string));
            dt.Columns.Add("EmailId", typeof(string));
            dt.Columns.Add("ContactNo", typeof(string));
            dt.Columns.Add("CourseEnroll", typeof(string));
            dt.Columns.Add("RegistrationDate", typeof(DateTime));
            //dt.Columns.Add("PermanentAddress", typeof(string));
            //dt.Columns.Add("ParentsName", typeof(string));
            //dt.Columns.Add("ParentsContact", typeof(string));
            //dt.Columns.Add("", typeof(string));
            //dt.Columns.Add("Address", typeof(string));
            //dt.Columns.Add("Address", typeof(string));
            //dt.Columns.Add("Address", typeof(string));

            dt.PrimaryKey = new DataColumn[] { dt.Columns["Id Number"] };
            return dt;
        }

        private DataTable CreateCourseTable()
        {
            var dt = new DataTable("Course");
            DataColumn dataColumn = new DataColumn("Id Number", typeof(int));
            dataColumn.AutoIncrement = true;
            dataColumn.AutoIncrementSeed = 1;
            dataColumn.AutoIncrementStep = 1;
            dt.Columns.Add(dataColumn);

            dt.Columns.Add("Name", typeof(string));
            dt.Columns.Add("DisplayText", typeof(string));
            // dt.Columns.Add("CourseDuration", typeof(string));

            dt.PrimaryKey = new DataColumn[] { dt.Columns["IdNumber"] };
            return dt;
        }

        private DataTable CreateStudentReportTable()
        {
            var dt = new DataTable("StudentReport");
            DataColumn dataColumn = new DataColumn("Id Number", typeof(int));
            dataColumn.AutoIncrement = true;
            dataColumn.AutoIncrementSeed = 1;
            dataColumn.AutoIncrementStep = 1;

            dt.Columns.Add(dataColumn);

            dt.Columns.Add("RegNo", typeof(string));
            dt.Columns.Add("Name", typeof(string));
            dt.Columns.Add("EmailId", typeof(string));
            dt.Columns.Add("Address", typeof(string));
            dt.Columns.Add("ContactNo", typeof(string));
            dt.Columns.Add("CourseEnroll", typeof(string));
            dt.Columns.Add("RegistrationDate", typeof(DateTime));


            //dt.PrimaryKey = new DataColumn[] { dt.Columns["ID"] };
            return dt;
        }
    }
}
```