Marking Scheme

Informatics College Pokhara

informatics
college pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: **Namuna Shrestha**
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	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 poorly written code and no comments at all
System Usability	unusable system

Overall Grade:	E+	E+

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

CU6004NP Application Development

30% Individual Coursework

2017-18 Spring

Name: Namuna Shrestha

College ID: NP04CP4A170018

University ID: 17030723

Table of Contents

Table of Figure

Table of Tables

1. Introduction

The introduced system is mainly for the student management and record keeping using C# (inside Visual studio) programming language. The main requirement related to student management is fulfilled with proper development and testing of the codes under different circumstances. In this project, registration of new student is created and stored in the database like format. The user interface thus created can handle the student and is flexible enough to enter the details of the student like email address, contact number, occupation etc. and we can also input the CSV file directly into the system. In a nutshell this project helps you visualize the data weekly and monthly as well as chart if necessary.

1.1 Current scenario

Depending upon the organization and management of student in different institution in the present world, I believe that most of them have very complex database management system. The worst scenario is that in some organization in our country still use paper-based student recording system, lacking the awareness of importance of digital DBMS using programming language.

1.2 Proposed system

This is the digitized version of oldest form of student recording system, mainly focused to remove the above-mentioned obstacles. This proposed project and its UI is going to overcome the problems and handle new comers into the college and display the student details in the systematic order. Hence understood even by the new user to the system.

2. User Manual

The screenshot below shows the initiation of the running program to carry out the record keeping of the students.

i. Login Screen

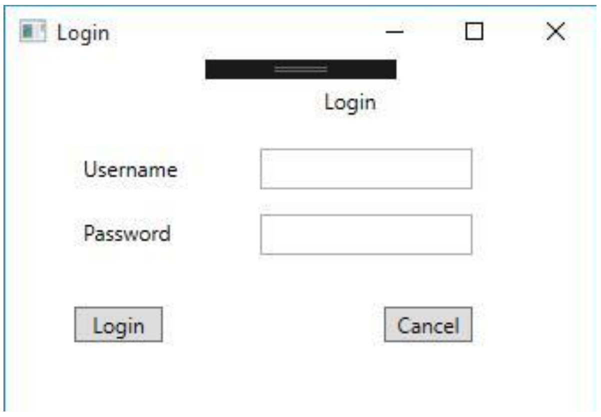
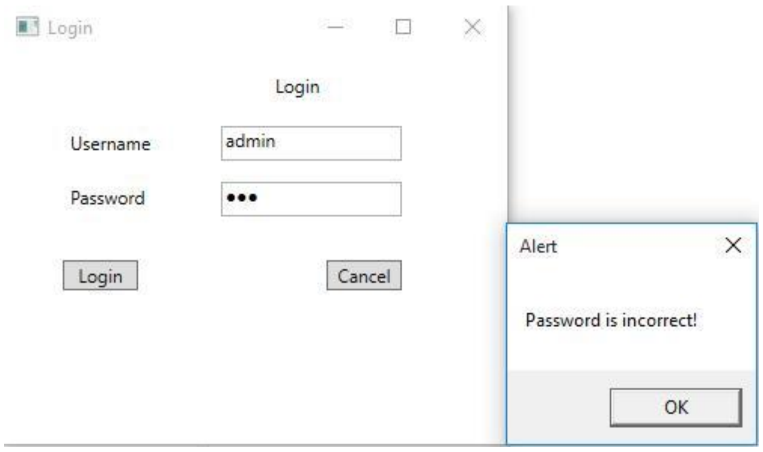


Figure 1: Login Screen

• The username and password for the system is admin and admin respectively.

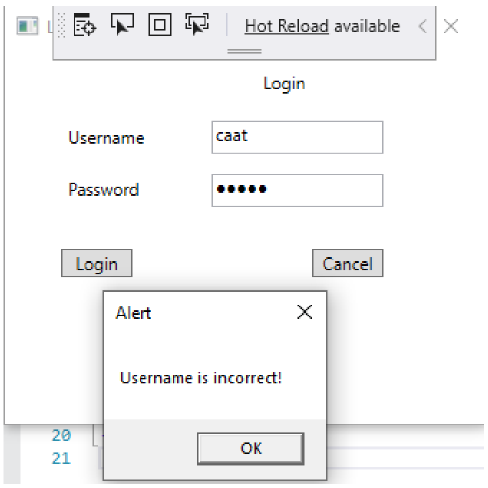
• If wrong username or password is entered it will display a message as shown in the figure below.



admin

Password is incorrect!

Figure 2: Error message displayed when password is incorrect


Hot Reload available

Login

Username caat

Password •••••

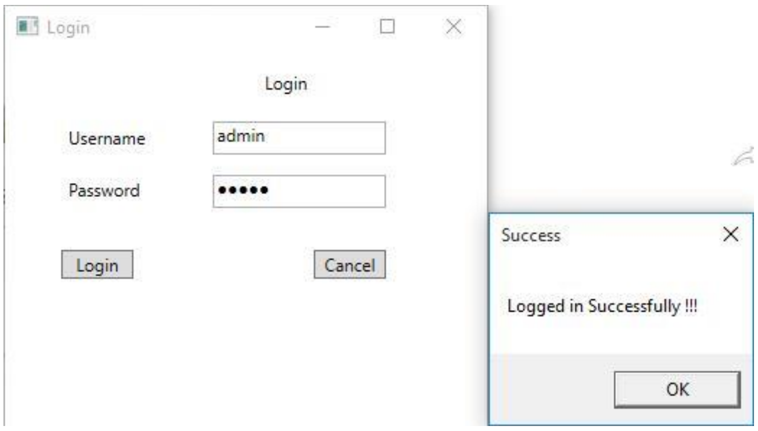
Login Cancel

Alert

Username is incorrect!

OK

Figure 3: Error messaged displayed on incorrect username


Login

Login

Username admin

Password •••••

Login Cancel

Success

Logged in Successfully !!!

OK

Figure 4: Successfully logged in

ii. Main page

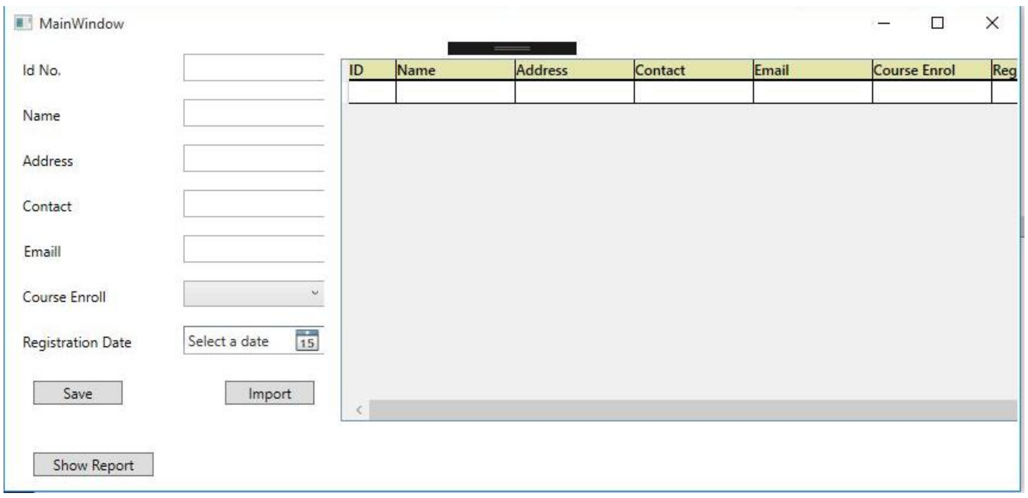
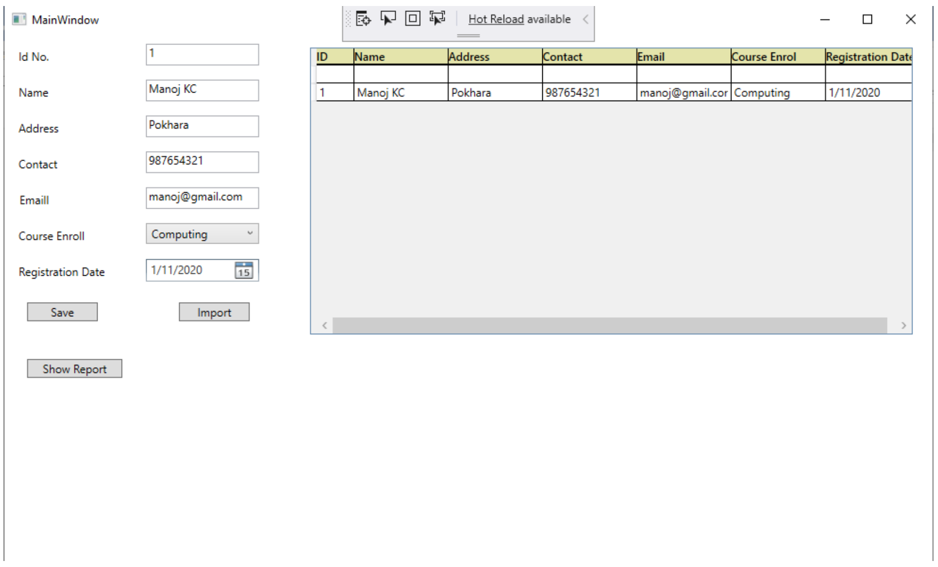


Figure 5: Main page

After logging in to the system, the main window of the system opens which is used for the registration of the students including other functions like:

- Registration Form
  The registration form is used to manually input student details from the form.
- Save
  The save button allows user to successfully save data inserted in the form in an external file
- Import
  After the user entries the file, this button allows user to see the currently inserted data in a tabular form.


MainWindow

Id No.	1
Name	Manoj KC
Address	Pokhara
Contact	987654321
Emaill	manoj@gmail.com
Course Enroll	Computing
Registration Date	1/11/2020

ID	Name	Address	Contact	Email	Course Enrol	Registration Date
1	Manoj KC	Pokhara	987654321	manoj@gmail.cor	Computing	1/11/2020

Figure 6: Import data from the form

- Data Grid

  The data grid displays the inserted data in a tabular form.

- Show Report

  This button directs user to another page named as student report.

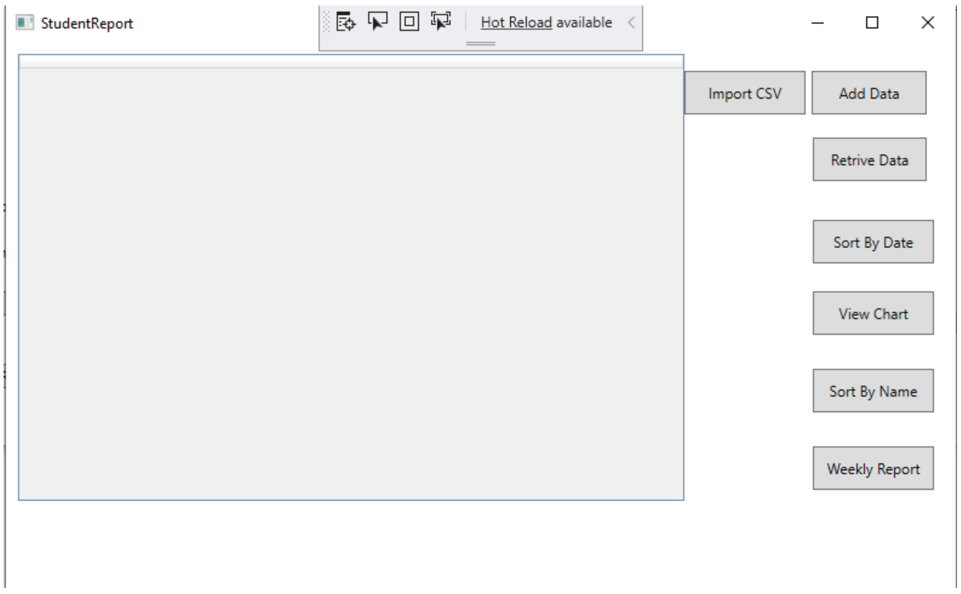


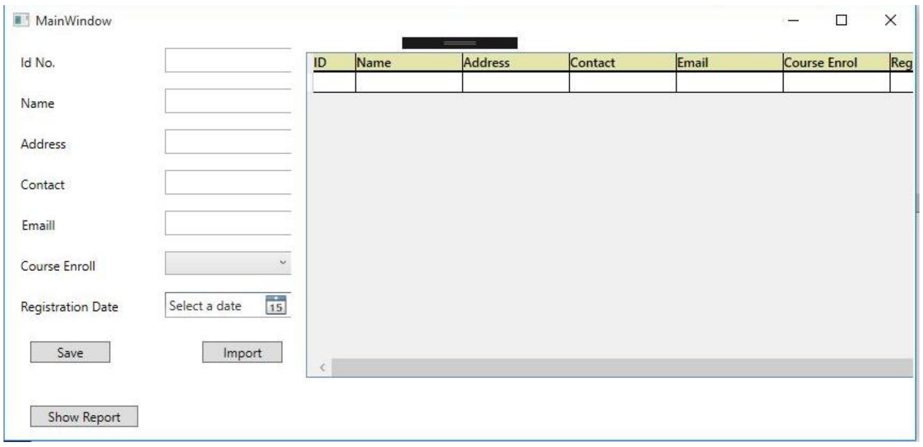
Figure 7: Show report

iii. Student Report

This page allows user to perform various functions like:

- Add Data

  This button helps user to fill up new data by directing them to the main page of the system.

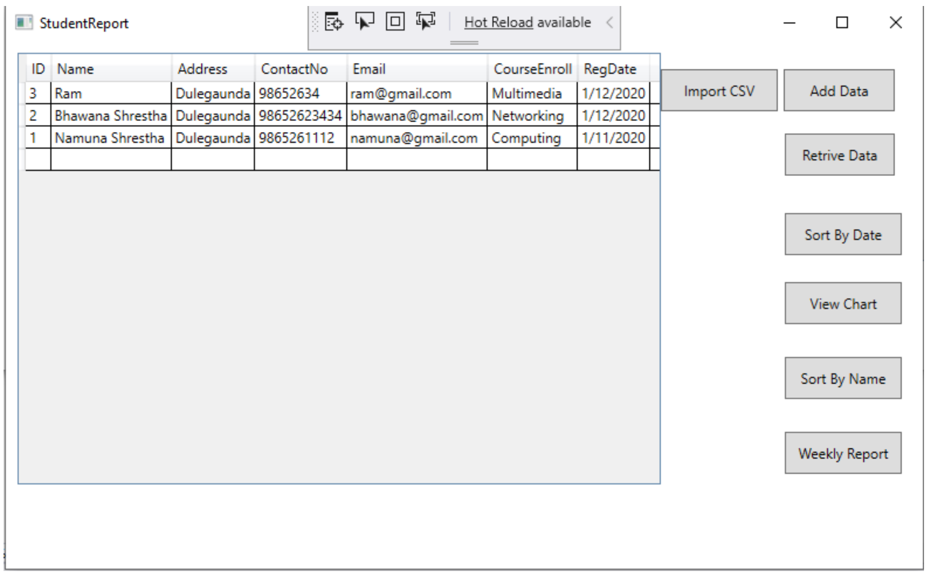


ID	Name	Address	Contact	Email	Course Enrol	Reg

Figure 8: Add data

- Retrieve Data

  This button helps user to view all the previously recorded data in a tabular form on the data grid part.

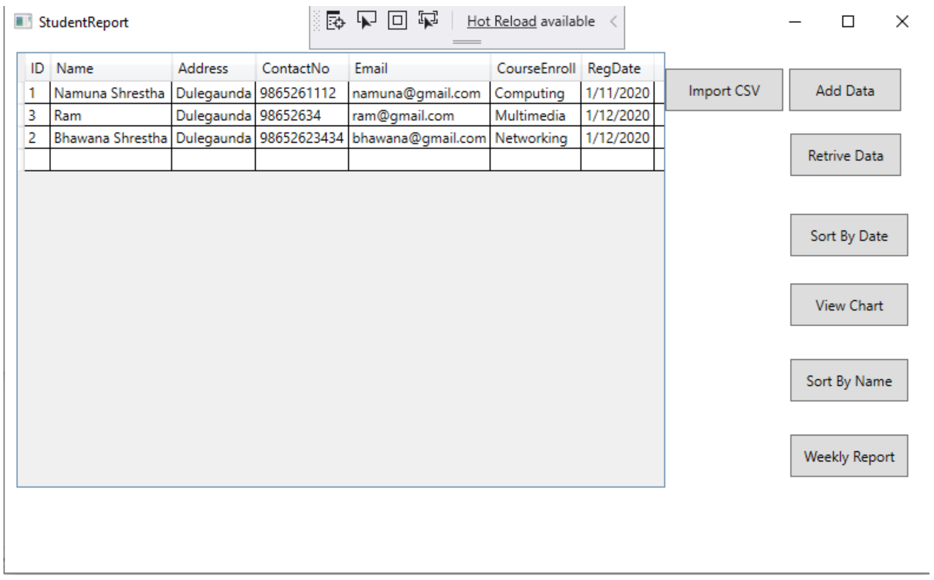


ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
3	Ram	Dulegaunda	98652634	ram@gmail.com	Multimedia	1/12/2020
2	Bhawana Shrestha	Dulegaunda	98652623434	bhawana@gmail.com	Networking	1/12/2020
1	Namuna Shrestha	Dulegaunda	9865261112	namuna@gmail.com	Computing	1/11/2020

Figure 9: Data retrieve

- Sort by Date

  This button is used to sort the student details according to the registration date in an ascending order as shown below.

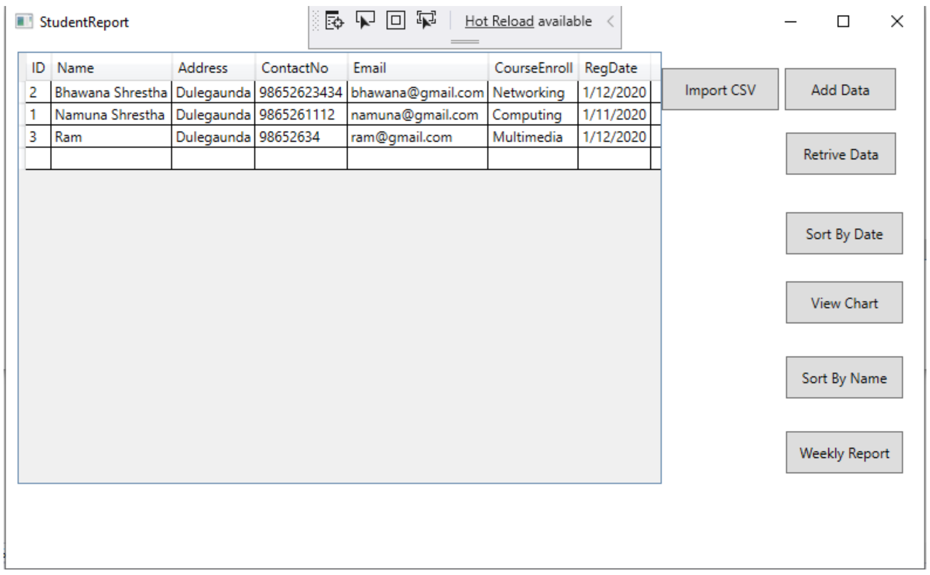


ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
1	Namuna Shrestha	Dulegaunda	9865261112	namuna@gmail.com	Computing	1/11/2020
3	Ram	Dulegaunda	98652634	ram@gmail.com	Multimedia	1/12/2020
2	Bhawana Shrestha	Dulegaunda	98652623434	bhawana@gmail.com	Networking	1/12/2020

Figure 10: Sorting by date

- Sort by Name

  This button is used to sort the student name in an alphabetical order as shown below:



ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
2	Bhawana Shrestha	Dulegaunda	98652623434	bhawana@gmail.com	Networking	1/12/2020
1	Namuna Shrestha	Dulegaunda	9865261112	namuna@gmail.com	Computing	1/11/2020
3	Ram	Dulegaunda	98652634	ram@gmail.com	Multimedia	1/12/2020

Figure 11: Sorting by name

- View Chart

  This button is used to display the chart on the basis of total number of students enrolled on programs like Computing, Networking and Multimedia. The pie chart is shown below.

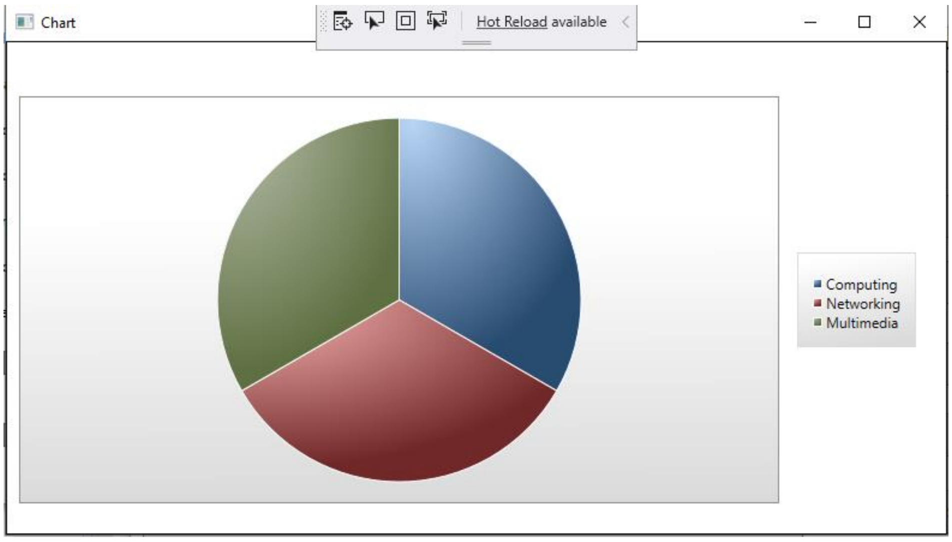
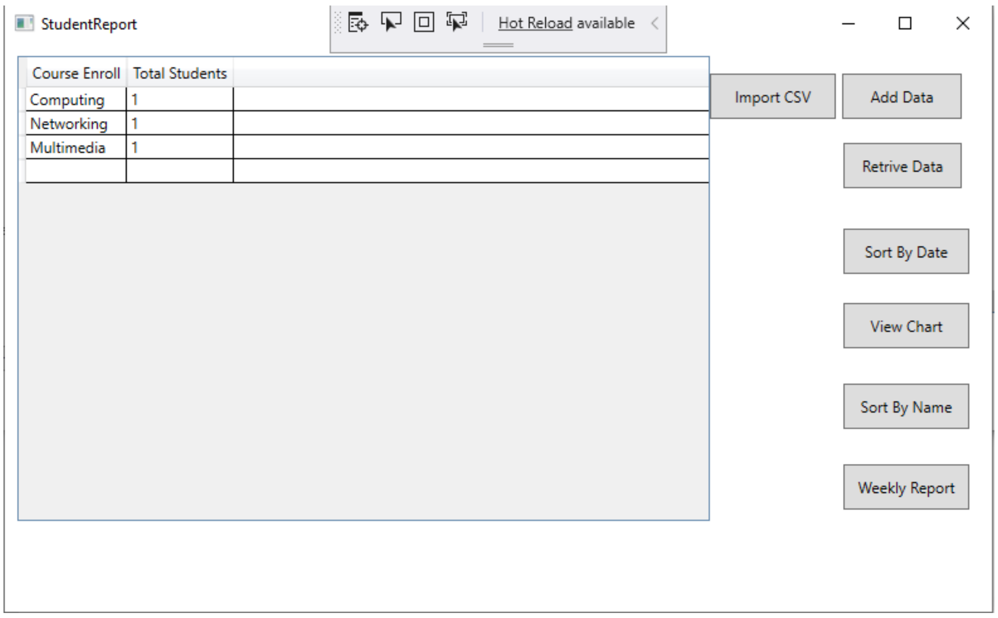


Figure 12: Weekly report in chart form

- Weekly report

  This button helps user view a report showing total number of students enrolled in each program named as Computing, Networking and Multimedia.

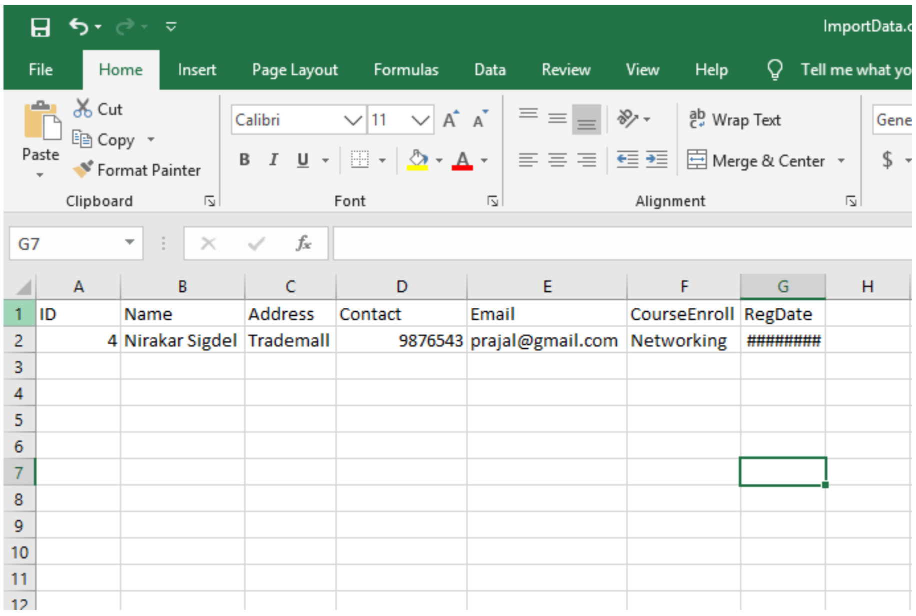


Course Enroll	Total Students
Computing	1
Networking	1
Multimedia	1

Figure 13: Weekly report in tabular form

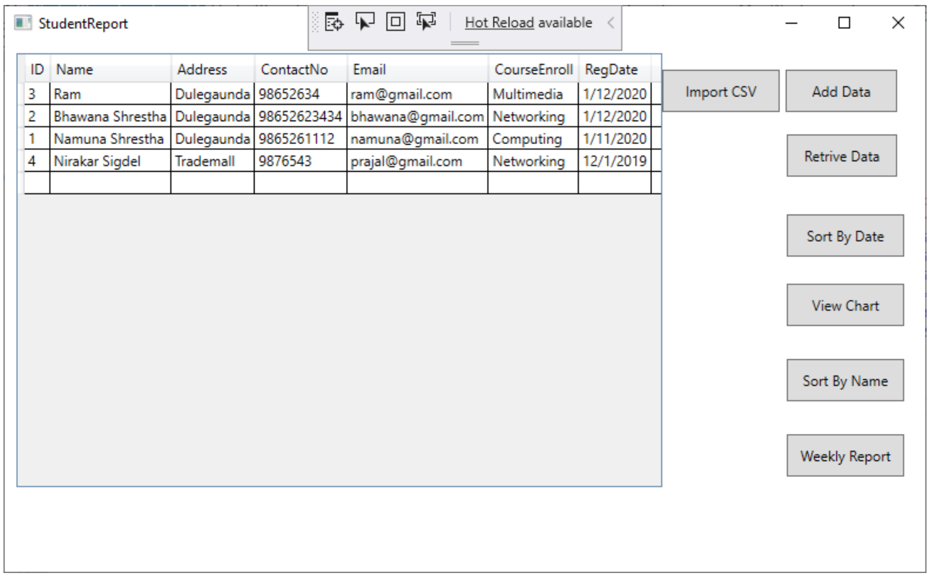
- Import CSV

  This button lets user to import student data from an external CVS file which can further be shown in a tabular form.



ID	Name	Address	Contact	Email	CourseEnroll	RegDate
4	Nirakar Sigdel	Trademall	9876543	prajal@gmail.com	Networking	########

Figure 14: Data to import from CSV



StudentReport

ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
3	Ram	Dulegaunda	98652634	ram@gmail.com	Multimedia	1/12/2020
2	Bhawana Shrestha	Dulegaunda	98652623434	bhawana@gmail.com	Networking	1/12/2020
1	Namuna Shrestha	Dulegaunda	9865261112	namuna@gmail.com	Computing	1/11/2020
4	Nirakar Sigdel	Trademall	9876543	prajal@gmail.com	Networking	12/1/2019

Figure 15: Successfully imported data of CVS file

3. System Architecture

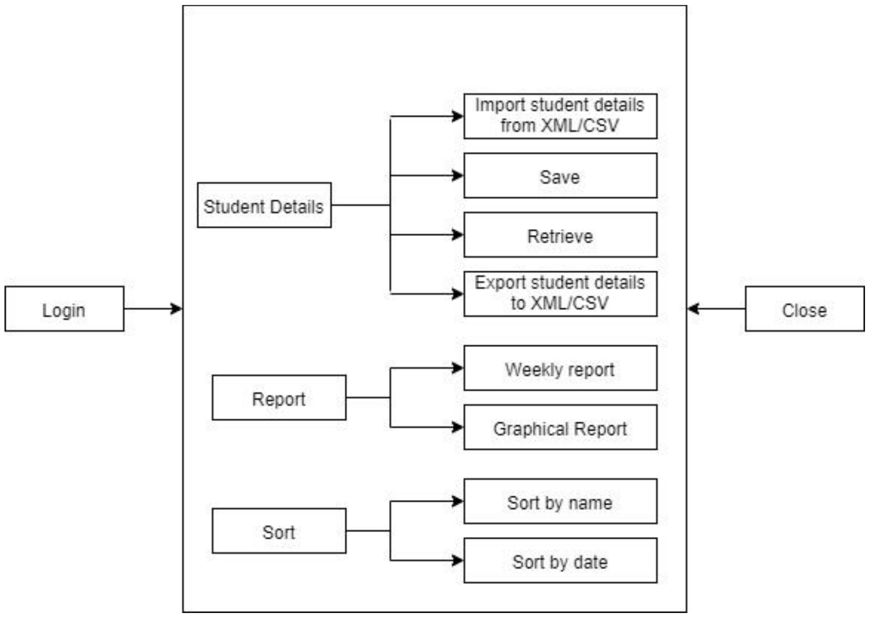


Figure 16: System architecture

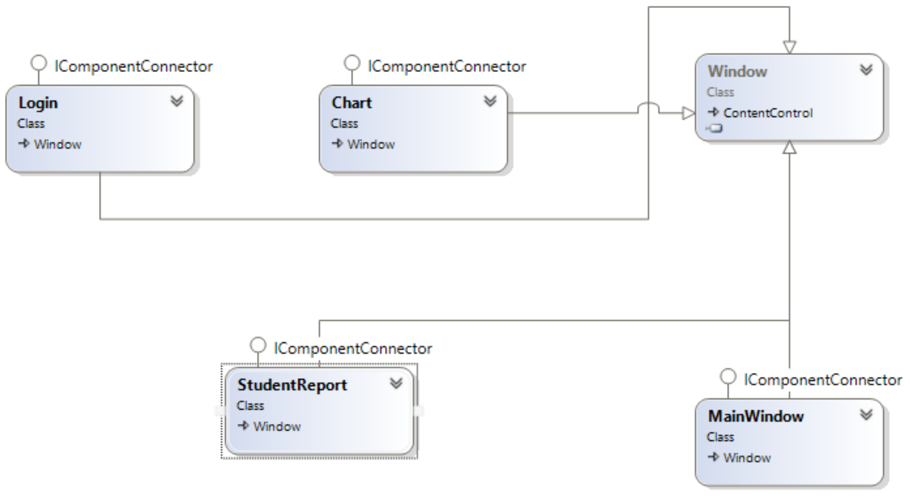


Figure 17: Class diagram

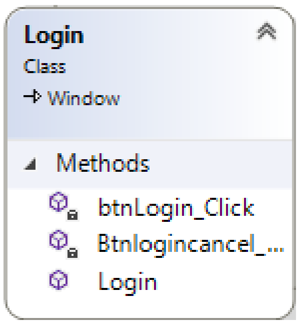


Figure 18: Individual Diagram of Login Class

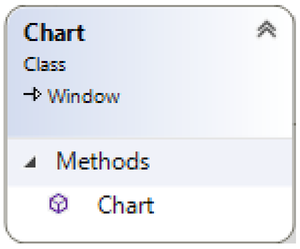


Figure 19: Individual Diagram of Chart class

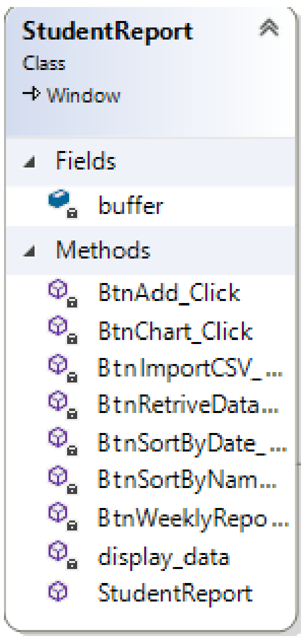


Figure 20: Individual Diagram of Student Report class

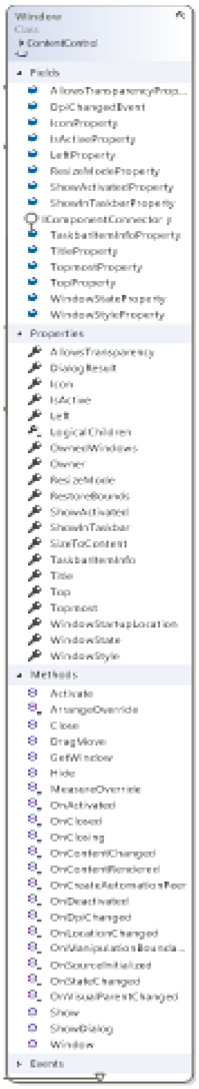


Figure 21: Individual Diagram of Main window class

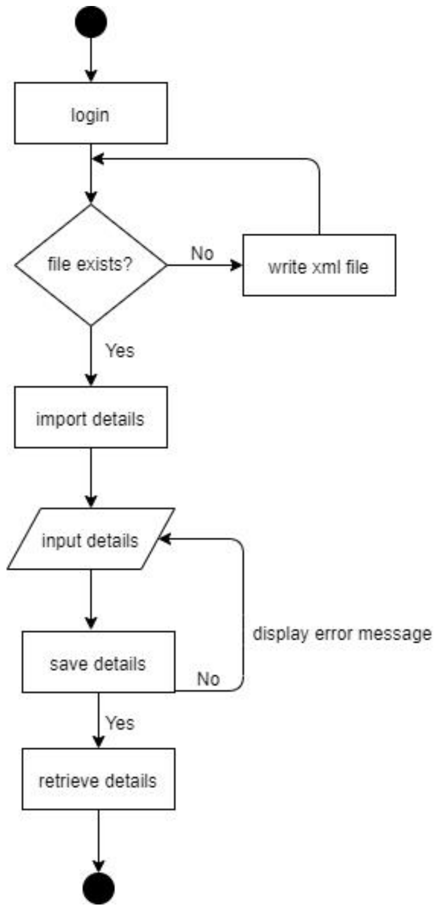


Figure 22: Flowchart for student enrolment

4. Bubble Sort 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.

Implementing Bubble Sort Algorithm

Following are the steps involved in bubble sort (for sorting a given array in ascending order):

1. Starting with the first element (index = 0), compare the current element with the next element of the array.
2. If the current element is greater than the next element of the array, swap them.
3. If the current element is less than the next element, move to the next element. Repeat Step 1.

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

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

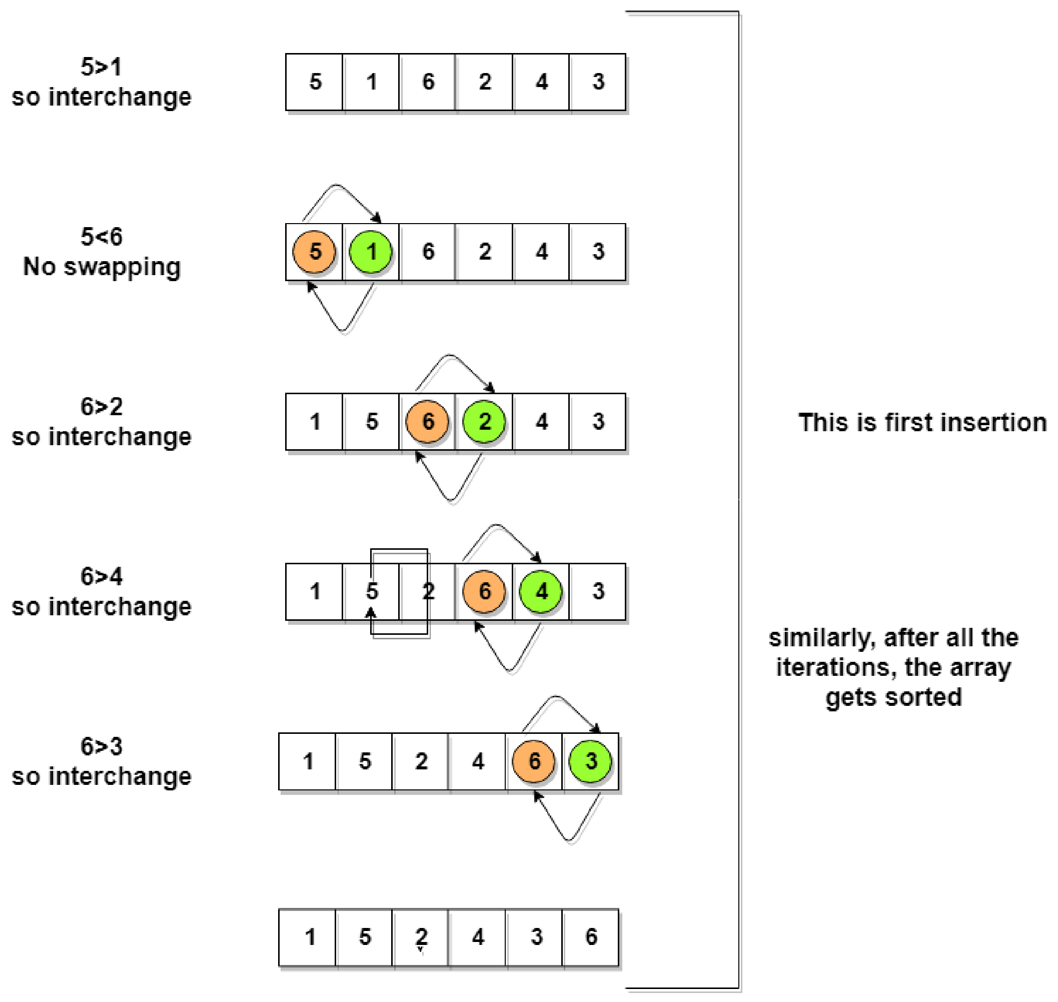


Figure 23: Bubble sort algorithm

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 (Abhijit Singh, n.d.).

5. Testing

After the successful development of the system, testing was done to make sure that there was no error while running the system.

- Test Case 1

Objective	To make sure if the user can login using wrong username and password and throws an error message
Output	Unsuccessful login and error message displayed

Table 1: Test Case1

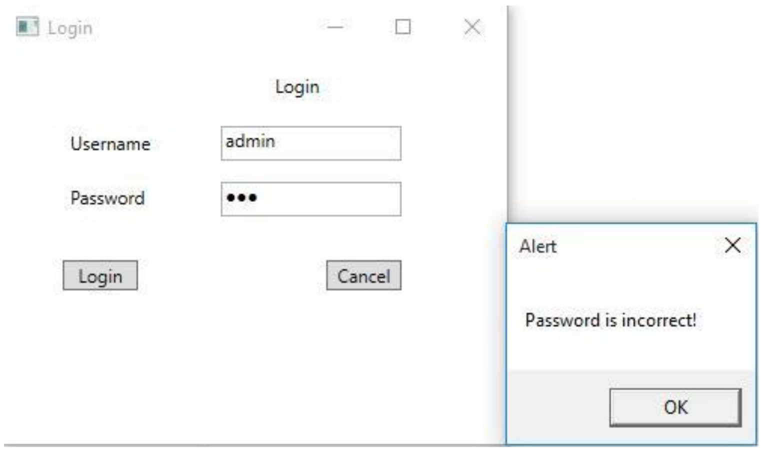


Figure 24: Logging in with incorrect password

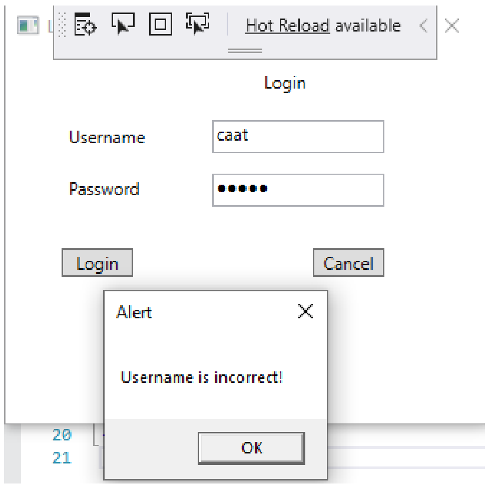
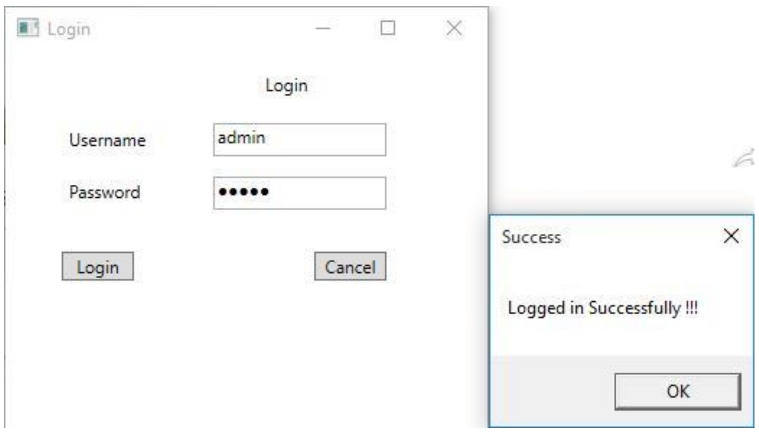


Figure 25: Logging in with incorrect username

- Test Case 2

Objective	To check the successful login to the system with correct username and password.
Output	Logged in successfully

Table 2: Test Case2



Username: admin

Password: •••••

Success

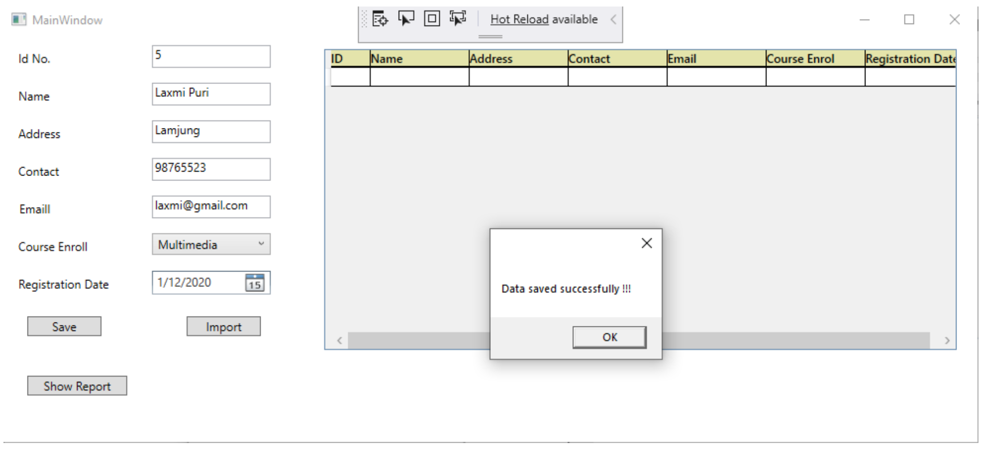
Logged in Successfully !!!

Table 3: Successfully logged in to system with correct username and password

- Test Case3

Objective	To check if the student data is inserted into the system successfully
Output	Data inserted successfully

Table 4: Test Case3



Id No.	5
Name	Laxmi Puri
Address	Lamjung
Contact	98765523
Emaill	laxmi@gmail.com
Course Enroll	Multimedia
Registration Date	1/12/2020

ID	Name	Address	Contact	Email	Course Enrol	Registration Date

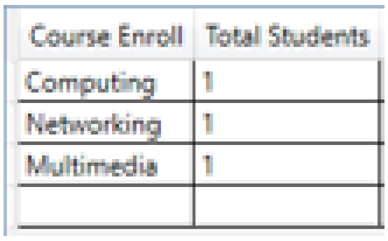
Data saved successfully !!!

Figure 26: Inserting student data in the system:

- Test Case 4

Objective	To check if the enrolled number of students can be displayed pie chart and in table using data grid.
Output	Data successfully displayed in DataGrid and in chart

Table 5: Test case4



Course Enroll	Total Students
Computing	1
Networking	1
Multimedia	1

Figure 27: Displaying weekly report on tabular form

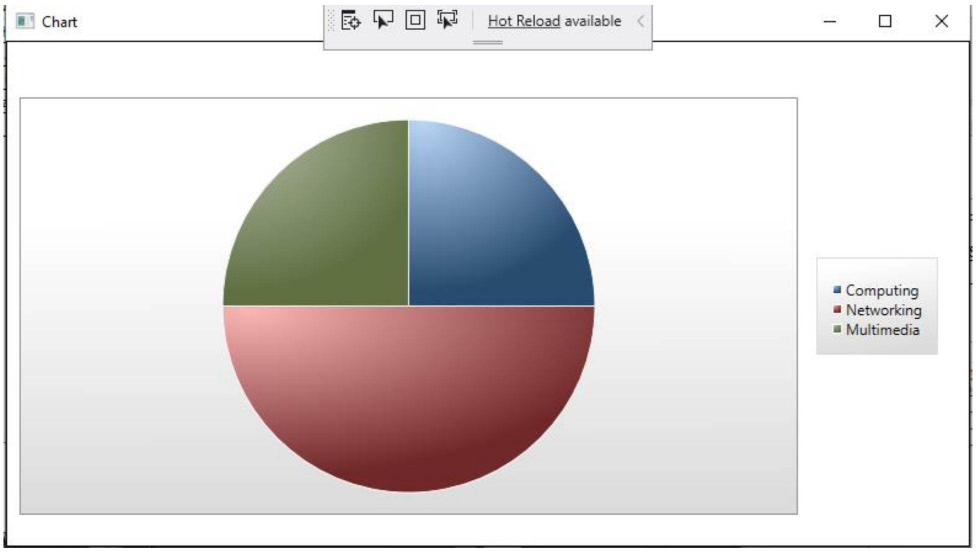
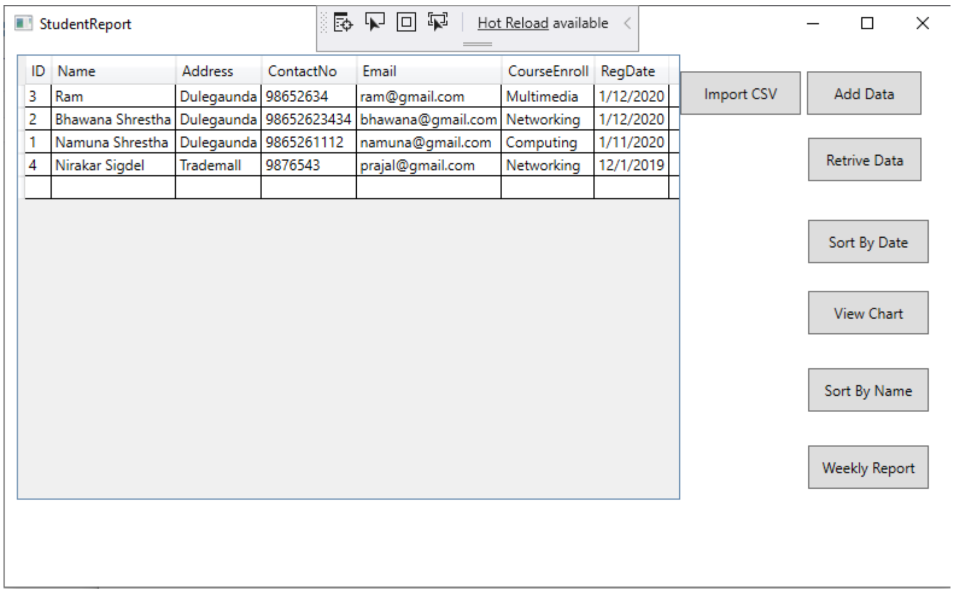


Figure 28: Displaying data in pie chart

- Test Case 5

Objective	To check if the previously inserted data can be retrieved on the data grid successfully
Output	Retrieved data successfully displayed in the data grid

Table 6: Test Case5



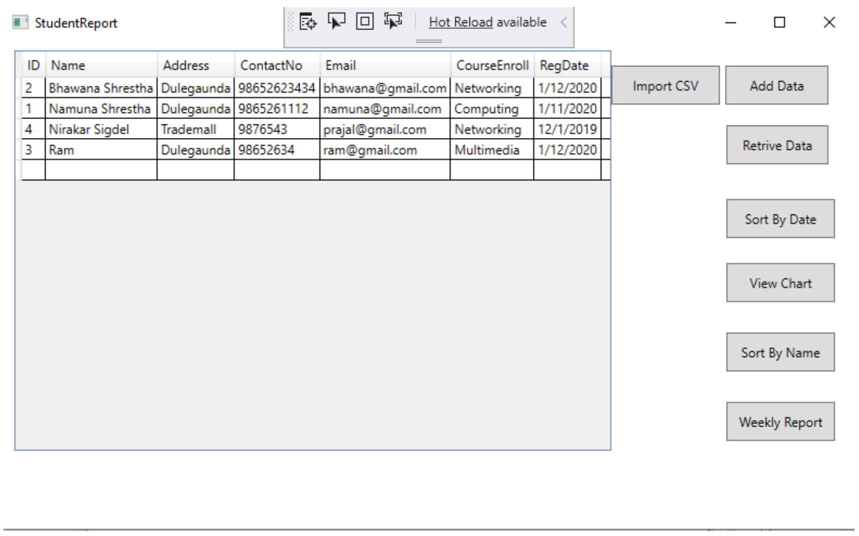
ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
3	Ram	Dulegaunda	98652634	ram@gmail.com	Multimedia	1/12/2020
2	Bhawana Shrestha	Dulegaunda	98652623434	bhawana@gmail.com	Networking	1/12/2020
1	Namuna Shrestha	Dulegaunda	9865261112	namuna@gmail.com	Computing	1/11/2020
4	Nirakar Sigdel	Trademall	9876543	prajal@gmail.com	Networking	12/1/2019

Figure 29: Displaying retrieved data

- Test Case 6

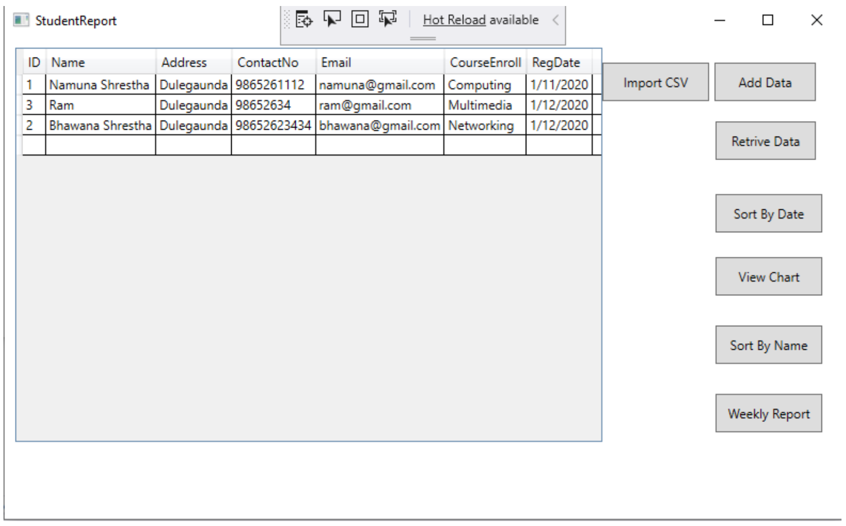
Objective	To check if the data be sorted in ascending order or alphabetical order
Output	Data successfully sorted by names in alphabetical order and registered date in ascending order

Table 7: Test Case6



ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
2	Bhawana Shrestha	Dulegaunda	98652623434	bhawana@gmail.com	Networking	1/12/2020
1	Namuna Shrestha	Dulegaunda	9865261112	namuna@gmail.com	Computing	1/11/2020
4	Nirakar Sigdel	Trademall	9876543	prajal@gmail.com	Networking	12/1/2019
3	Ram	Dulegaunda	98652634	ram@gmail.com	Multimedia	1/12/2020

Figure 30: Displaying data in alphabetical order



ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
1	Namuna Shrestha	Dulegaunda	9865261112	namuna@gmail.com	Computing	1/11/2020
3	Ram	Dulegaunda	98652634	ram@gmail.com	Multimedia	1/12/2020
2	Bhawana Shrestha	Dulegaunda	98652623434	bhawana@gmail.com	Networking	1/12/2020

Figure 31: Displaying data according to the dates in ascending order

6. Journals and research

We believe that record keeping and management is very necessary aspect while designing the system and working on it and making them secured and more reliable is the biggest challenge in the present world and using C# and java like platform can be the biggest advantage while creating the student management system and implementing them in the biggest of the institutes. In the first paragraph about this programming language suggests us that it is highly safe and security can be managed together which is all we need and this IEEE published journal was all I need for the best system development (IEEE; M.Nicol, 2019).

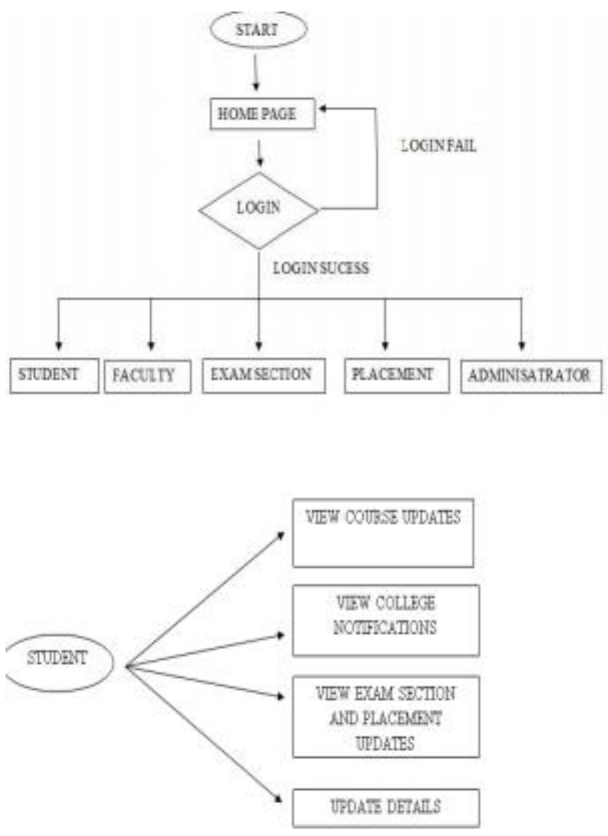


Figure 32: Research and journal

Looking at the similar project published on the other journal from communication engineering published by student of Andre Pradesh we can see the complexity of the project as well as different branches of management for simple student management in this journal looking at the flow chart and diagrams of the maintenance I can predict it is too vast to control and administer this student management multiple task has been broken inside the project but all this features are simply added in my project with simple lines of codes and less effort for the management of it so why not my project? And it won't work as the server is down and my code will still work. And this project is (RB, 2013)

7. Reflection

The designed system id the user based digital new student management system with the knowledge of general system administration can operate the system. Proper data arrangement by name and registration are the features and proper import and export of CSV data can also be done for easy management. Here the file is also recorded based on date and time of enrolment of the students.

Explaining the previous experience with the visual studio believe I choose the best platform for my database management and proper GUI display. Gathering experience and creating features made me more habitual with this language and get rid of obstructions and prevent the bugs. Shorting the data and secured data management was reality after using this software for this project. I worked with the aim and fulfilled the objective such that I can design the same project in the coming days.

8. Conclusion

Working with visual studio and C# as baseline programming language the data handling was done and successfully recorded and tested under many circumstances and different phases of data keeping methodologies. Here the framework checks the actual user via authentication the redirects the admin into the detail retrieving framework. Giving you access to the table of the students studying in the different faculties and is flexible enough to import single data or bulk data from CSV. Every components and class of this programming language was well utilized and project was completed.

And I would like to thank my respected supervisor Mr. Ishwor Sapkota for giving me this opportunity to work me in this project and guiding me throughout the project.

9. Bibliography

Abhijit Singh, A. D. (n.d.). Retrieved from Study Tonight: https://www.studytonight.com/data-structures/bubble-sort

IEEE. (n.d.). security and privacy. *8.2.1*(8013).

M.Nicol, D. (2019). IEEE Security and privacy. *6*(8013).

RB, G. (2013). *WEB based student management , 2*(6), 4.

10. Appendix

Login Class:

```
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
{
    public partial class Login : Window
    {
        public Login()
        {
            InitializeComponent();
        }

        private void btnLogin_Click(object sender, RoutedEventArgs e)
        {
            if (txtUname.Text != "admin")
            {
                MessageBox.Show("Username is incorrect!", "Alert",
MessageBoxButton.OK);
                txtUname.Clear();
            }
            else if (txtPw.Password != "admin")
            {
                MessageBox.Show("Password is incorrect!", "Alert",
MessageBoxButton.OK);
                txtPw.Clear();
            }
            else
            {
                MessageBox.Show("Logged in Successfully !!!", "Success",
MessageBoxButton.OK);
                MainWindow mainWindow = new MainWindow();
                mainWindow.Show();
            }
        }
        private void Btnlogincancel_Click(object sender, RoutedEventArgs e)
        {
            Close();
        }
    }
}
```

Main Window:

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

            Student student = new Student();
            DataGridX.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 Email { get; set; }

            public string CourseEnroll { get; set; }

            public string RegDate { get; set; }
        }
        private void btnSave_Click(object sender, RoutedEventArgs e)
        {
            var handler = new DataHandler();
            var dataSet = handler.CreateDataSet();
            AddSampleData(dataSet);
            //dataSet.WriteXmlSchema(@"D:\studentDataSchema.xml");

            MessageBox.Show("Data saved successfully !!!");
            if (File.Exists(@"D:\studentData.xml"))
            {
                dataSet.ReadXml(@"D:\studentData.xml");
                dataSet.WriteXml(@"D:\studentData.xml");
            }
            else
            {
                dataSet.WriteXml(@"D:\studentData.xml");
            }
        }

        private void AddSampleData(DataSet dataSet)
        {
            var dr1 = dataSet.Tables["Student"].NewRow();
            dr1["ID"] = txtId.Text;
            dr1["Name"] = txtName.Text;
            dr1["Address"] = txtAddress.Text;
            dr1["Contact"] = txtContact.Text;
            dr1["Email"] = txtEmail.Text;
            dr1["CourseEnroll"] = txtCourseEnrl.Text;
            dr1["RegDate"] = txtRegDate.Text;
            dataSet.Tables["Student"].Rows.Add(dr1);
        }

        private void btnImport_Click(object sender, RoutedEventArgs e)
        {
            Student dataStudent = new Student();
            dataStudent.ID = txtId.Text;
            dataStudent.Name = txtName.Text;
            dataStudent.Address = txtAddress.Text;
            dataStudent.Contact = txtContact.Text;
            dataStudent.Email = txtEmail.Text;
            dataStudent.CourseEnroll = txtCourseEnrl.Text;
            dataStudent.RegDate = txtRegDate.Text;

            DataGridX.Items.Add(dataStudent);
        }
        private void BtnReport(object sender, RoutedEventArgs e)
        {
            StudentReport studentReport = new StudentReport();
            studentReport.Show();
        }
    }
}
```

Student Report class:

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

namespace ApplicationDevelopmentCW
{

    public partial class StudentReport : Window
    {
        DataTable buffer;
        public StudentReport()
        {
            InitializeComponent();
        }
        private void display_data()
        {
            string dataXMLFile = @"D:\studentData.xml";
            DataSet dataset = new DataSet();
            dataset.ReadXml(dataXMLFile);

            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 dataView = new DataView(buffer);
            DataGridRp.ItemsSource = dataView;

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

        private void BtnRetriveData_Click(object sender, RoutedEventArgs e)
        {
            display_data();
        }
```

```
private void BtnSortByDate_Click(object sender, RoutedEventArgs e)
{
    DataView view = new DataView(buffer);
    view.Sort = "RegDate ASC";
    DataGridRp.ItemsSource = view;
}

private void BtnSortByName_Click(object sender, RoutedEventArgs e)
{
    DataView view = new DataView(buffer);
    view.Sort = "Name ASC";
    DataGridRp.ItemsSource = view;
}

private void BtnWeeklyReport_Click(object sender, RoutedEventArgs e)
{
    var dataSet = new DataSet();
    dataSet.ReadXml(@"D:\studentData.xml");
    DataTable dtStdRp = 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 < dtStdRp.Rows.Count; i++)
    {
        String col = dtStdRp.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);

    DataGridRp.DataContext = dt.DefaultView;
}

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

private void BtnImportCSV_Click(object sender, RoutedEventArgs e)
{
    try
    {
        var dataSet = new DataSet();
```

```
                dataSet.ReadXml(@"D:\studentData.xml");
                Microsoft.Win32.OpenFileDialog openFileDlg = new
Microsoft.Win32.OpenFileDialog();

                if (openFileDlg.ShowDialog() == true)
                {
                    string filePath = openFileDlg.FileName;
                    using (var scan = new StreamReader(filePath))
                    {
                        scan.ReadLine();
                        while (!scan.EndOfStream)
                        {
                            var line = scan.ReadLine();
                            var values = line.Split(',');
                            var newRow = dataSet.Tables["Student"].NewRow();
                            newRow["ID"] = values[0];
                            newRow["Name"] = values[1];
                            newRow["Address"] = values[2];
                            newRow["Contact"] = values[3];
                            newRow["Email"] = values[4];
                            newRow["CourseEnroll"] = values[5];
                            newRow["RegDate"] = values[6];
                            dataSet.Tables["Student"].Rows.Add(newRow);

                            dataSet.WriteXml(@"D:\studentData.xml");
                        }
                    }
                    MessageBox.Show("Student details imported successfully.",
"Import Sucessful.", MessageBoxButton.OK, MessageBoxImage.Asterisk);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }


        }
    }

}
```

Chart:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Windows;
using System.Windows.Controls.DataVisualization.Charting;

namespace ApplicationDevelopmentCW
{
    public partial class Chart : Window
    {
        public Chart()
        {
            InitializeComponent();
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\studentData.xml");
            DataTable dtStdRp = 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 < dtStdRp.Rows.Count; i++)
            {
                String col = dtStdRp.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);

            ((PieSeries)chartGrid).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) };
        }
    }
}
```

Data Handler class:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ApplicationDevelopmentCW
{
    class DataHandler
    {
        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");
            dt.Columns.Add("ID", typeof(string));
            dt.Columns.Add("Name", typeof(string));
            dt.Columns.Add("Address", typeof(string));
            dt.Columns.Add("Contact", typeof(string));
            dt.Columns.Add("Email", typeof(string));
            dt.Columns.Add("CourseEnroll", typeof(string));
            dt.Columns.Add("RegDate", typeof(DateTime));
            return dt;
        }

        private DataTable CreateCourseTable()
        {
            var dt = new DataTable("Course");
            DataColumn dataColumn = new DataColumn("ID", 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.PrimaryKey = new DataColumn[] { dt.Columns["ID"] };
            return dt;
        }

        private DataTable CreateStudentReportTable()
        {
            var dt = new DataTable("StudentReport");
            DataColumn dataColumn = new DataColumn("ID", 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("Contact", typeof(string));
            dt.Columns.Add("Email", typeof(string));
            dt.Columns.Add("CourseEnroll", typeof(string));
            dt.Columns.Add("RegDate", typeof(DateTime));
            return dt;
        }
    }
}
```