Marking Scheme

# Informatics College Pokhara

informatics
college pokhara

## Application Development

## CS6004NI

## Course Work 1


**Submitted By:** **Abhisekh Lamichhane**
**London Met ID:** 17030683

**Submitted To:** Ishwor Sapkota
Module Leader

Marking Scheme

| 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** | appropriate use of data types but missing some properties required or missing CRUD operation |
| **Data Validation** | No validation at all |
| **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** | Sorting is implemented for not function properly |
| **B. Documentation** | |
| **User Manual for running the application** | User Manual is good. Contains all varieties of forms. |

Marking Scheme

| | |
|---|---|
| **Application architecture & description of the classes ad methods sued** | architecture is included and satisactory descriptoin of class and methods used. |
| **Flow chart, algoriathms and data sctructures used** | average work with very limited explanation and missing diagramatic representation. |
| **Reflective essay** | satisfactorily written about experience and learnings |

**C. Programming Style**

| | |
|---|---|
| **Clarity of code,Popper Naming convention & comments** | Code is poorly written and lacks comments |
| **System Usability** | System can be used but is in efficient and in attracative |

| | | |
|---|---|---|
| **Overall Grade:** | **B+** | **B+** |

**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.

In overall the code is working and all the functionality seems working and system can be used. All feature implemented with few minor bugs



# Application Development
# CS6004NP
# Coursework 1

**Submitted By:**

Student Name: Abhishek Lamichane

London Met ID: 17030683

Group: L3C2

Date: 10th January 2020

**Submitted To:**

Mr. Ishwor Sapkota

# Table of Contents


**1. Introduction** ..... 1

1.1 Current Scenario ..... 1

1.2 Proposed System ..... 1

**2. User Manual** ..... 2

**3. Journal Articles** ..... 12

**4. System Architecture** ..... 15

4.1 Architecture Diagram ..... 15

4.2 Class Diagram ..... 16

4.3 Individual Diagram ..... 17

4.4 Flow-Chart of Enrol Student ..... 19

4.5 Flow-Chart of Import Data from CSV Flow Chart ..... 20

**5. Sorting Algorithm** ..... 21

**6. Reflection** ..... 24

**7. Conclusion** ..... 25

**8. Bibliography** ..... 26

**9. Appendix** ..... 27

## List of Figure

Figure 1: Login Form....2
Figure 2: Login Successfully....3
Figure 3: Invalid Username....3
Figure 4: Invalid Password....4
Figure 5: Student Enrol Screen....5
Figure 6: Require All Fields to Press Submit Button....5
Figure 7: Invalid Name....6
Figure 8: Invalid Contact Number....6
Figure 9: Invalid Email Address....7
Figure 10: Adding Student Details....7
Figure 11: Adding Student Details Successfully....8
Figure 12: Student Details Submitted Successfully....8
Figure 13: Display Student Details Window....9
Figure 14: Displaying Pie-Chart According Weekly Report....9
Figure 15: Successfully Sorted by Name....10
Figure 16: Successfully Sorted by Registration Date....10
Figure 17: Weekly Report Showing Total Number of Students....11
Figure 18: Successfully Import CSV File and Merge with XML File....11
Figure 19: Architecture Diagram....15
Figure 20: Class Diagram....16
Figure 21: Flow-Chart of Enrol Students....19
Figure 22: Flow-Chart of Import Data from CSV Flow Chart....20

List of Table

Table 1: Individual Diagram of Login Form .....17
Table 2: Individual diagram of Main window (Add Students Data) .....17
Table 3: Individual diagram of Students Details .....18
Table 4: Individual diagram of Chart .....18

# 1. Introduction

In the generation of technology change, the record keeping system should be maintained and should be kept safe. So, the classical traditional way of keeping record should now be changed into digitalized form which more secure and reliable than classical tradition that denotes paper-based system. As per in case of school, the students record keeping should into changed to digital form.

The project is all about School Management System which keep records of student. The system is highly designed developed and test under various circumstance. The features and functions that are required by School are almost fulfilled by the developed system. It consists of functions like adding student personal detail including registration date so that system can generate a weekly enrolment report of the student. In addition to that, system includes all details like name, address, contact no, email, program enrol, registration date and daily wage amount of the employee. Furthermore, there is a function to view daily and weekly table and chart. This system is used to keep track of the student's details, program enrol and registration date. Other several functions are well explained briefly in other sections of the report.

## 1.1 Current Scenario

There are several schools who keep records of their data in classical traditional system which is paper-based system result it can be lost. In additional to that, there are some school with digital system but are well lacking and occurs many problems while inserting the data or function which are needed for school.

## 1.2 Proposed System

The proposed system is digitized system or change modern tradition system which is specially designed to overcome problem mentioned above and it is safe and secure where admin can also track the student details and other related information from any other computer and anywhere in the world. The system ensures security with the presence of login section. Entry of data and display of data have been made easy with presence of easy user-interface.

## 2. User Manual

The user manual for using the Student Management system is described below:

- **Login Screen**

  At first, system will appear login screen which will be the security screen. If username and pass word will be invalid then it pop-pop error message and could not access next page. The username and password of the system is “admin” and “admin” respectively. Only a valid username and password can provide to access to the system or next page.


*Figure 1: Login Form*


*Figure 2: Login Successfully*


*Figure 3: Invalid Username*

- The username and password for this system is admin and admin respectively.
- If wrong username and password is entered it will display a message as below:


*Figure 4: Invalid Password*

- **Add Students Details**

  When the user login Successfully, the following screen will be displayed.

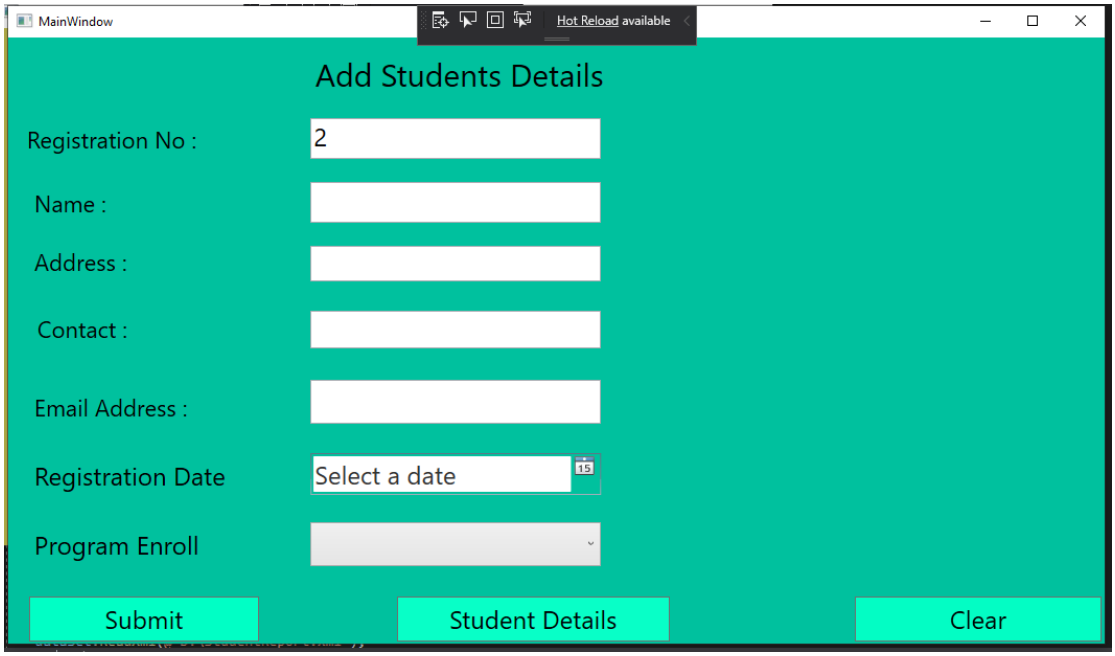



*Figure 5: Student Enrol Screen*

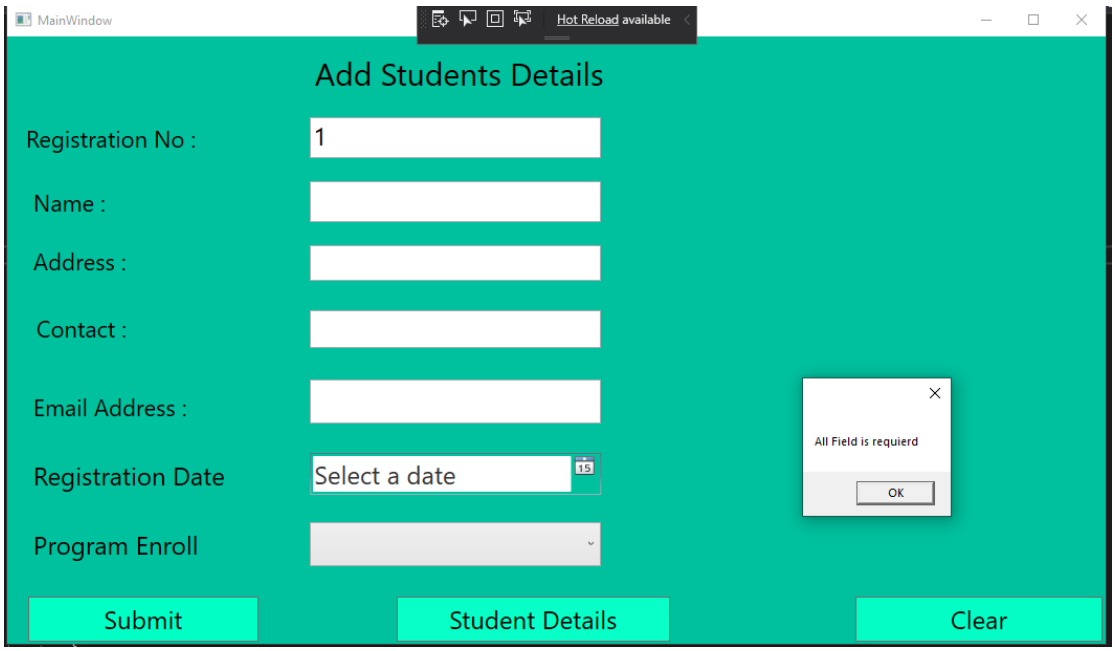



*Figure 6: Require All Fields to Press Submit Button*

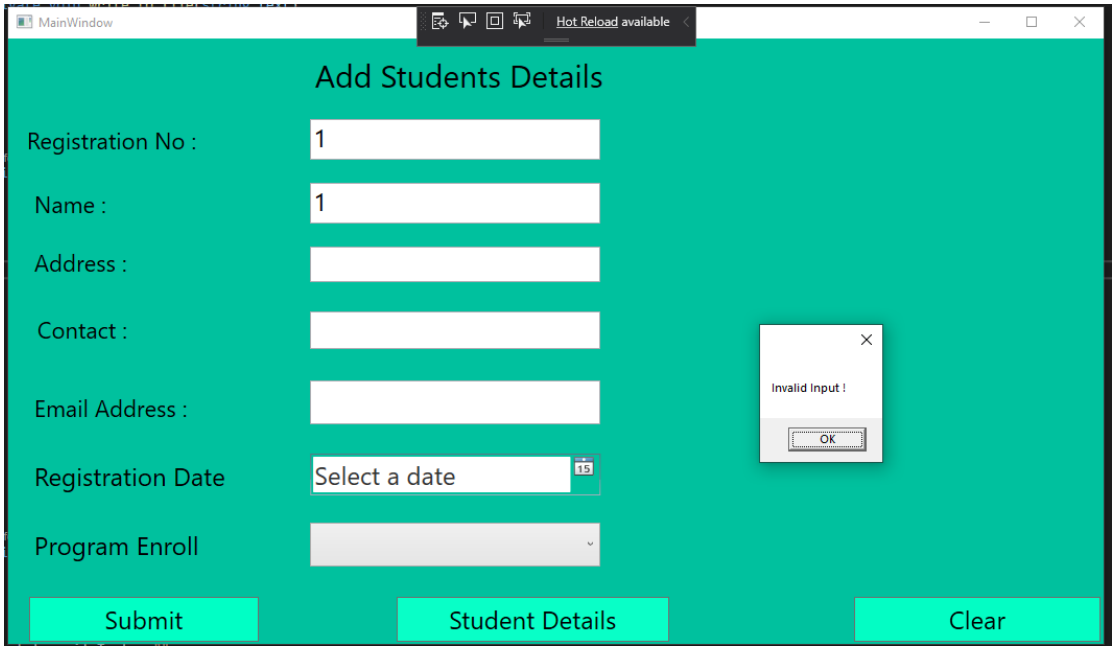



*Figure 7: Invalid Name*

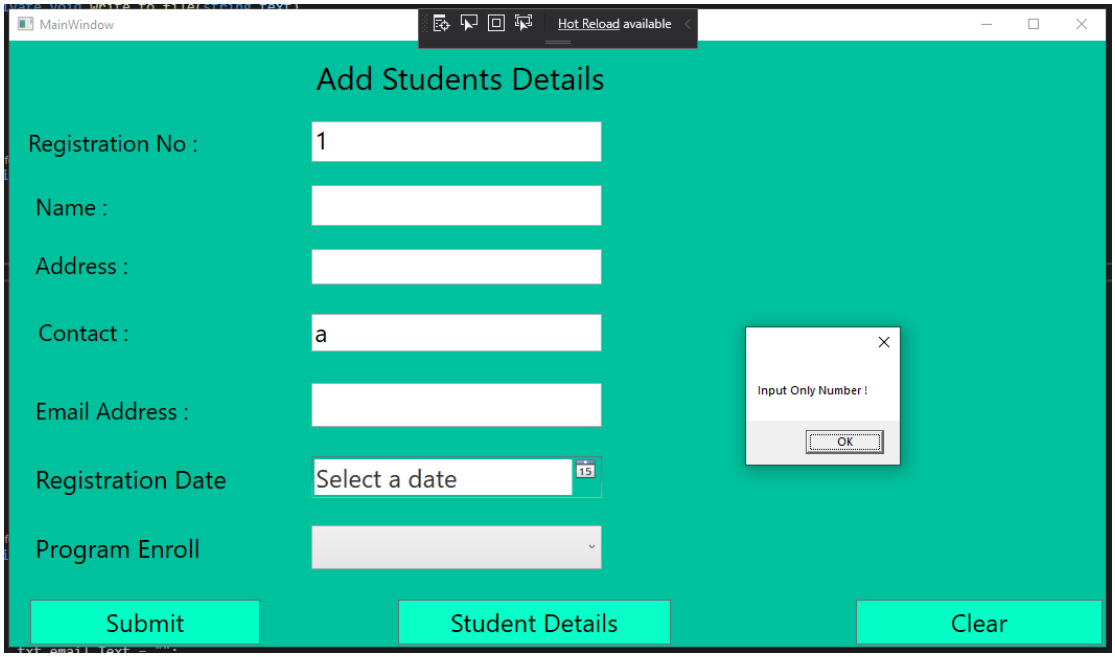



*Figure 8: Invalid Contact Number*

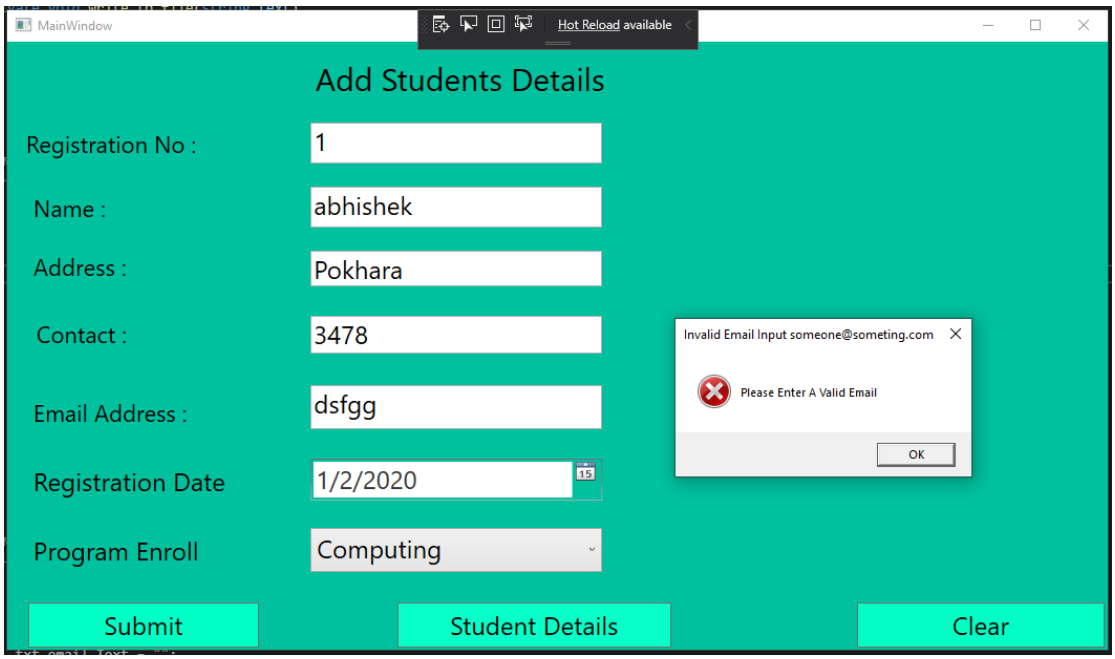



*Figure 9: Invalid Email Address*

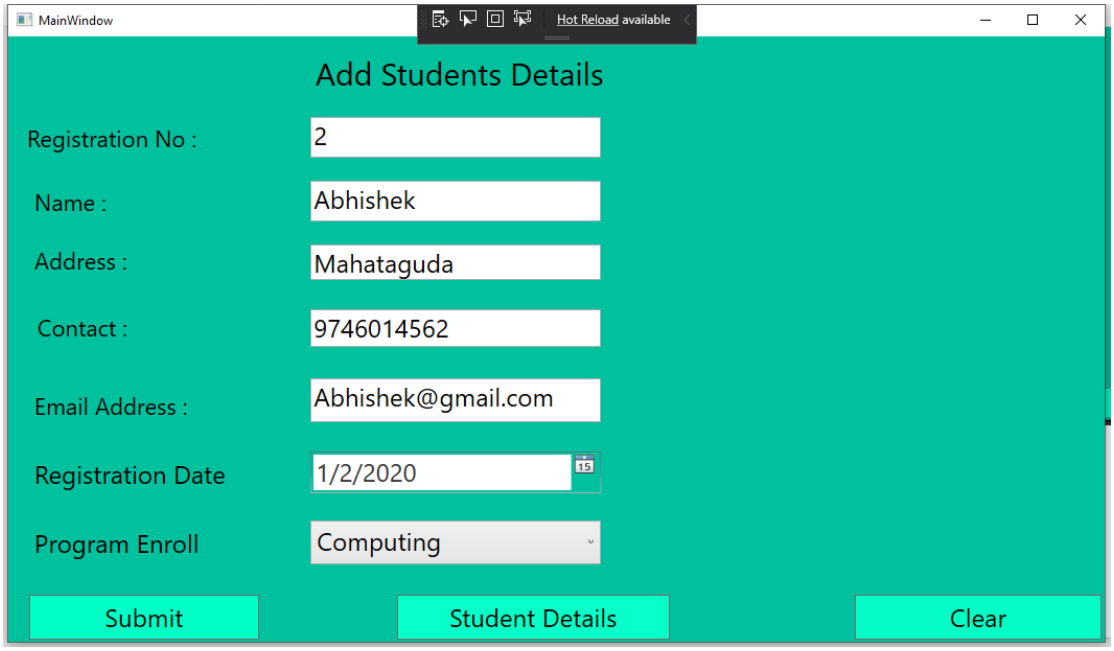



*Figure 10: Adding Student Details*

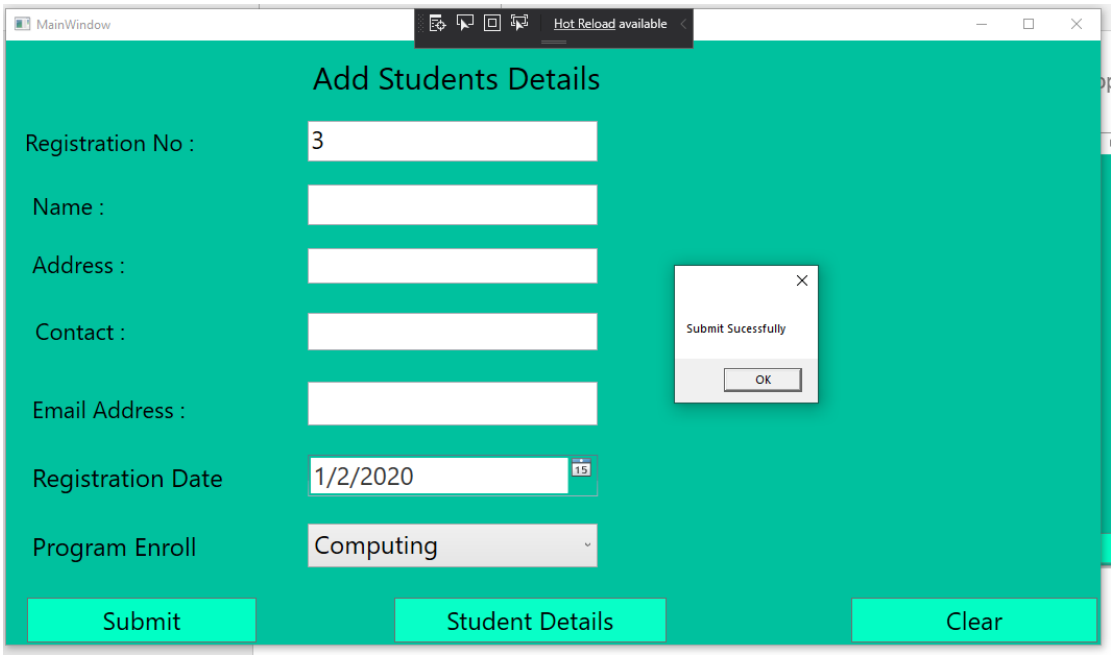



*Figure 11: Adding Student Details Successfully*

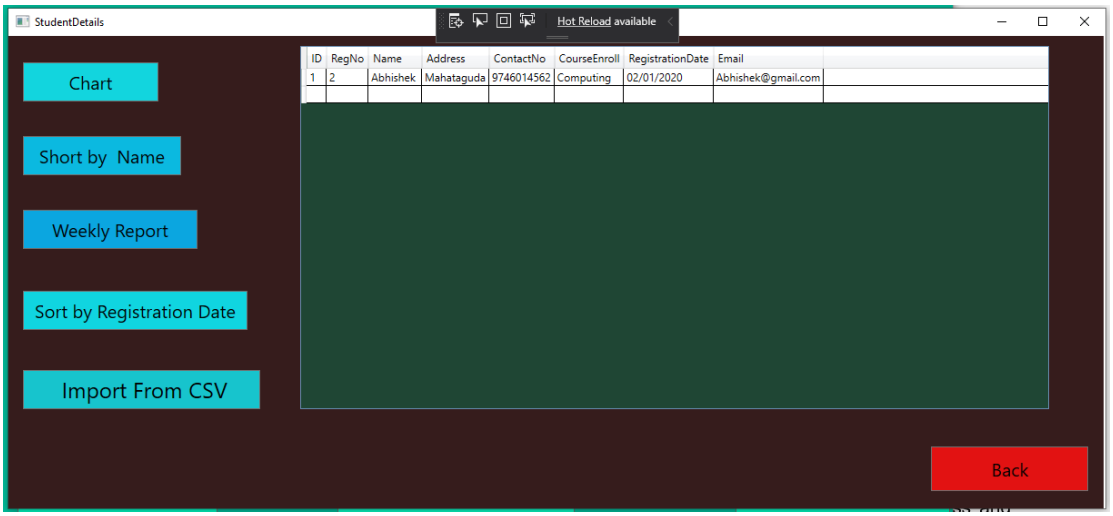



*Figure 12: Student Details Submitted Successfully*

- Submit: The Submit button on student detail form adds the detail of Student Details. i.e. First name, email, contact number, address and enrol course and it also include validation.
- Student Details: The Student details button is as a retrieve button which will display all details in grid.
- Clear: The clear button is used for clear all filed area and input next details.

- **Student Details**

  When the user input the details of the student in the fields and click the submit button to enrol the student in the system, then following screen will be displayed.

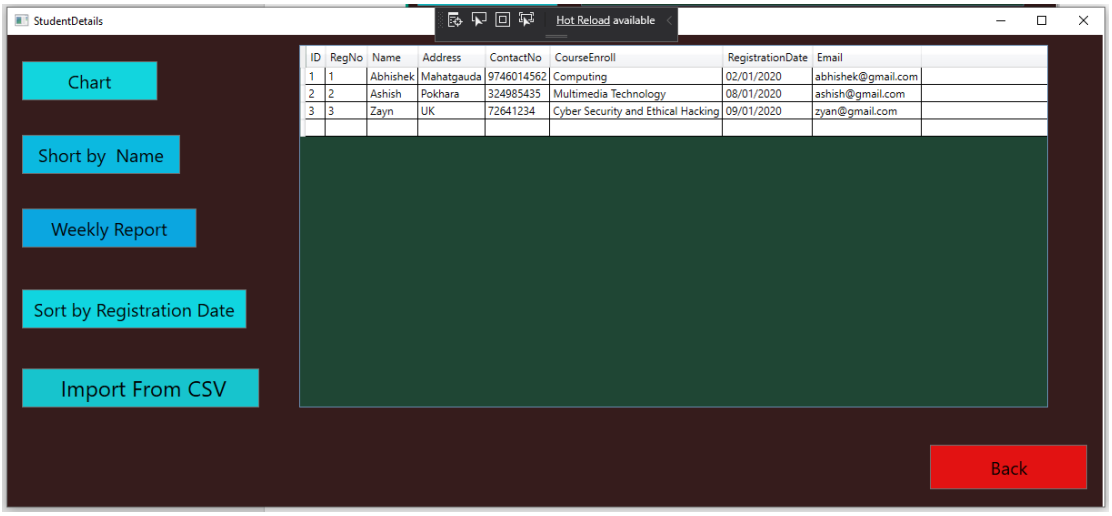



*Figure 13: Display Student Details Window*

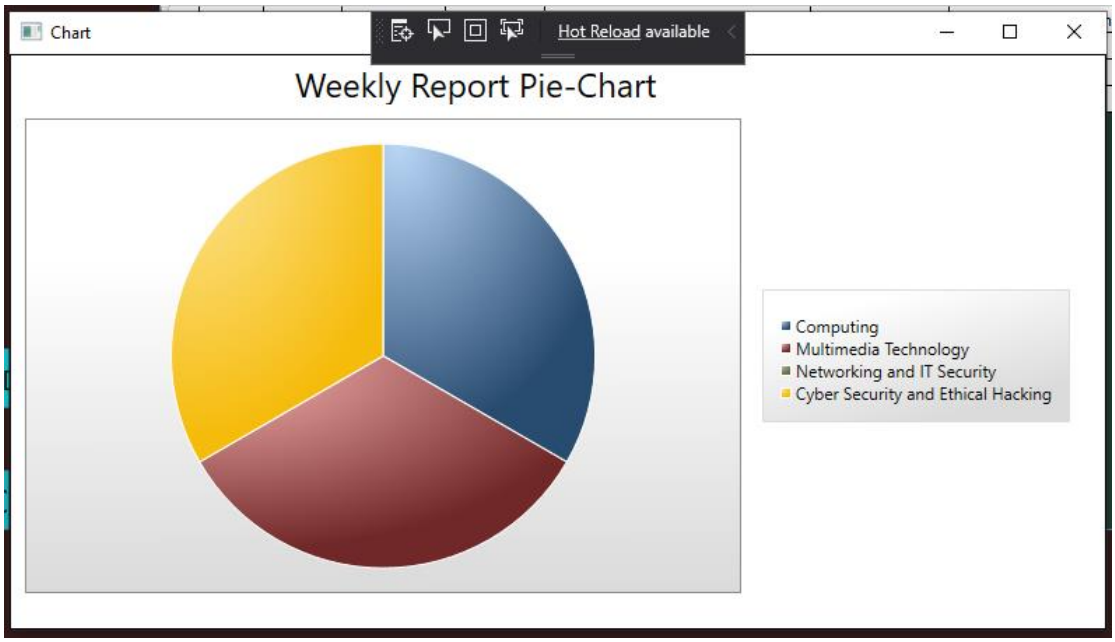



*Figure 14: Displaying Pie-Chart According Weekly Report*

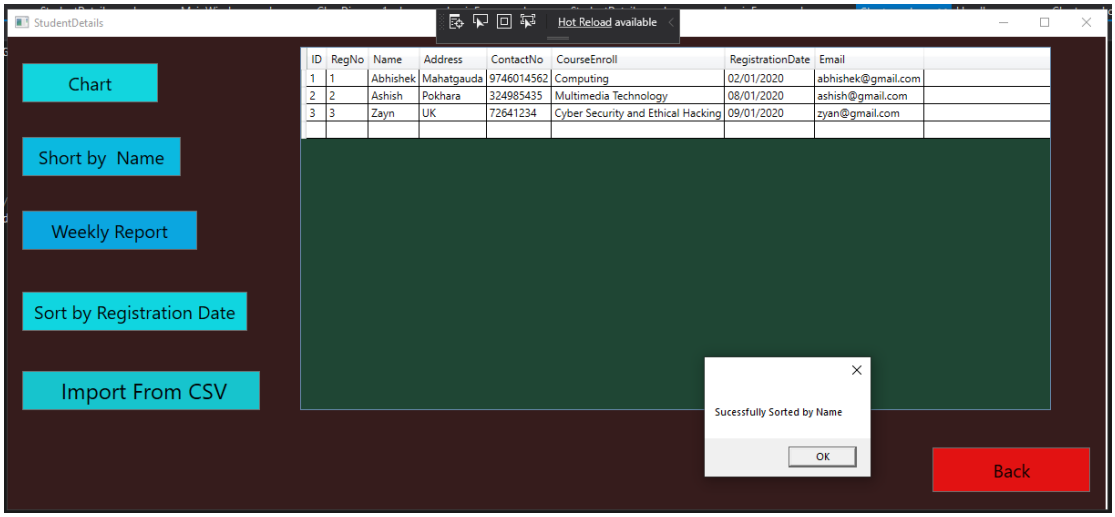



*Figure 15: Successfully Sorted by Name*

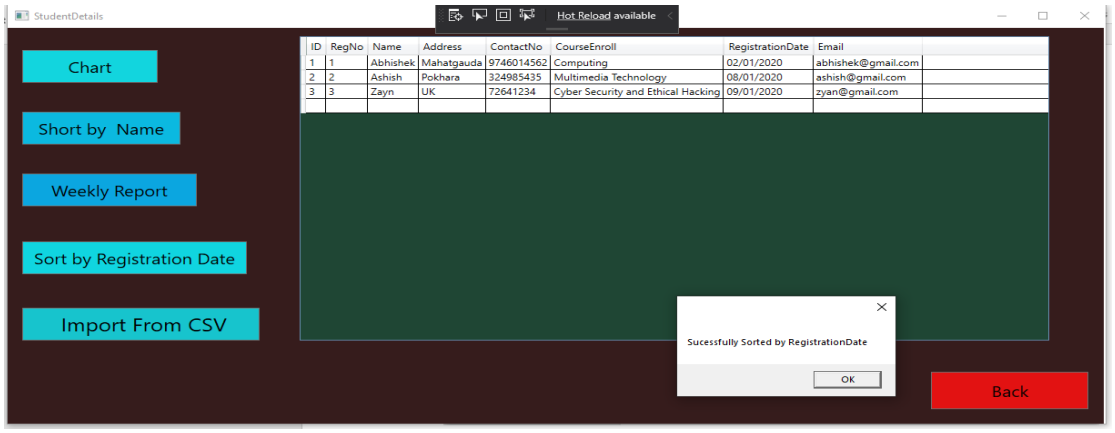



*Figure 16: Successfully Sorted by Registration Date*

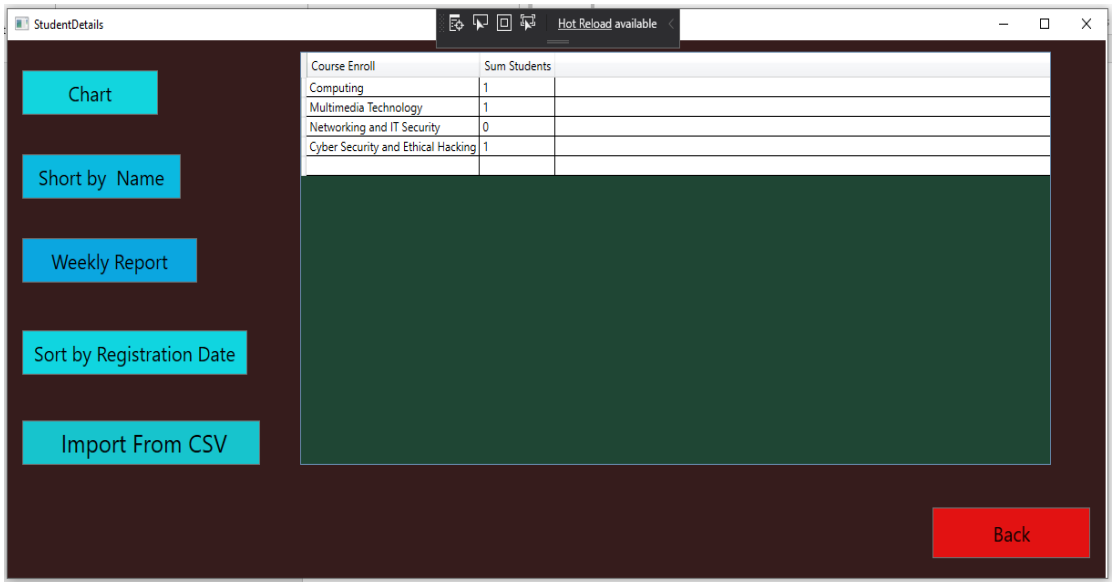



*Figure 17: Weekly Report Showing Total Number of Students*

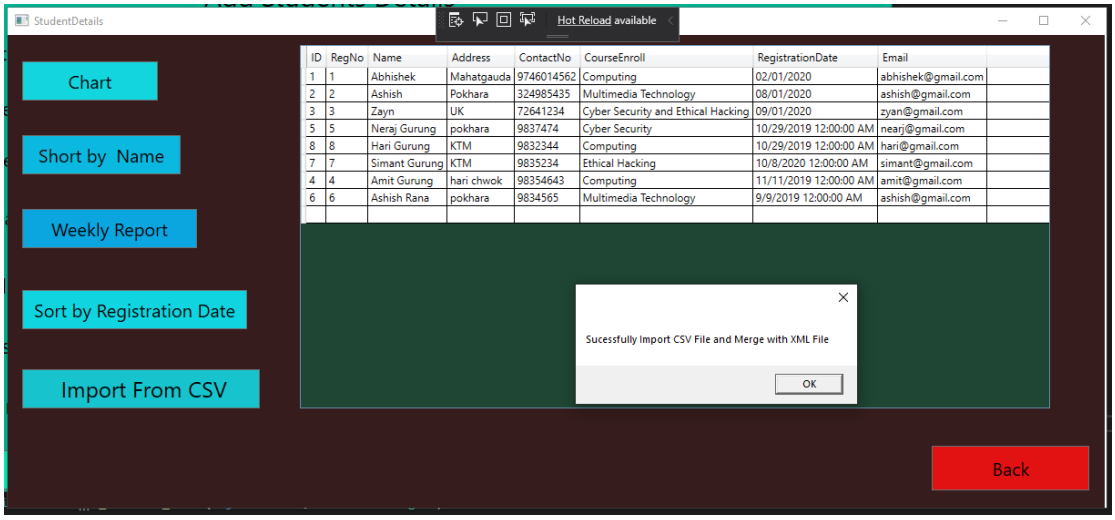



*Figure 18: Successfully Import CSV File and Merge with XML File*

- Chart: The Chart button lets allow to view the pie chart which is generate by weekly report.
- Short by Name: The short by name button lets user to sort the value according to the name.
- Weekly Report: The weekly report button lets allow to view total number of students enrol in different courses in the system.
- Short by Registration Date: The short by name button lets user to sort the value according to the name.
- Import From CSV: The import from CSV button lets user to import the CSV file from external file and merge with XML file.

## 3. Journal Articles

- The chapter focuses on a cross-national comparison of mediatized schools in Germany and England. Based on the assumption that both school systems follow the same goal of providing good school education, the question arises as to why the mediatized equipment is so different. Our empirical results show that English schools are far more mediatized, exhibiting a higher number of computers, notebooks and tablets in schools as well as digital systems and services. Non-mediatized communication forms dominate in German schools with a high usage of pen and paper or pigeon holes. The different mediatized practices also affect communication with pupils and parents, following the same characteristics as inter-teacher communication. On the other hand, teachers in both countries emphasize the importance of face-to-face contact and direct personal communication. One reason for the differences may be founded in the different educational governance of both countries (Breiter, 2018).

- 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, 2006).

- ACM Transactions on Management Information Systems (TMIS) is a scholarly quarterly journal that focuses on publishing high quality information systems research. TMIS welcomes innovative work on the design, development, assessment, and management of information technology and systems within organizations, businesses, and societies. TMIS welcomes submissions on a full range of MIS and information technology related areas and strongly encourages submissions with technical and technological ingredients, such as algorithmic, analytical modelling, design science, and system-oriented research, as well as submissions in emerging multidisciplinary MIS research topics that may span several traditional academic disciplines (Zeng, 2010).

- In nowadays increasingly ICT-related investments in an overgrowing competitive school setting, the deployment, maintenance, and particularly the effective use of ICT is in many ways a complex multifaceted managerial task involving several stakeholder groups. In a pilot study of a single school district in a Swedish municipality, we have interviewed representatives from the municipal board of education, the municipal IT-support for schools, and two principals. We have used Technology Acceptance Model (TAM3) as our analytical lens to explore how school principals' and municipal IT-managers perceive ICT adoption, usefulness, and the potential role of ICT. We conclude that the barriers for a successful integration of ICT into school-related activities requires a holistic managerial thinking in order to overcome the lack of coordinated ICT investment strategy and tracking of ICT (Babaheidari, 2014).

- Quality becomes milestone for enhancing learning and teaching practices through contemporary standards. In this respect, considering the outcomes of transformation is essential. School management is an umbrella of facilitating learning and teaching. Therefore, role of digital transformation in school management is an intensified need to be underlined. In establishing school culture, school management plays an essential role that digital transformation effects the school management and culture for the quality in education. The nature of this research is qualitative. Reflective reports of headmasters are gathered as data to understand the readability and awareness of school management to the digital transformation in the case of North Cyprus. In addition, content analysis is done to realize the upcoming issues in school management and digital transformation. As this focus is demanding for developing countries especially North Cyprus, this research revealed that headmasters have awareness on the use of digital and technological facilities. They are aware on digital transformation although budget and application restrictions are revealed to apply this transformation (Altınay, 2016).

# 4. System Architecture

## 4.1 Architecture Diagram

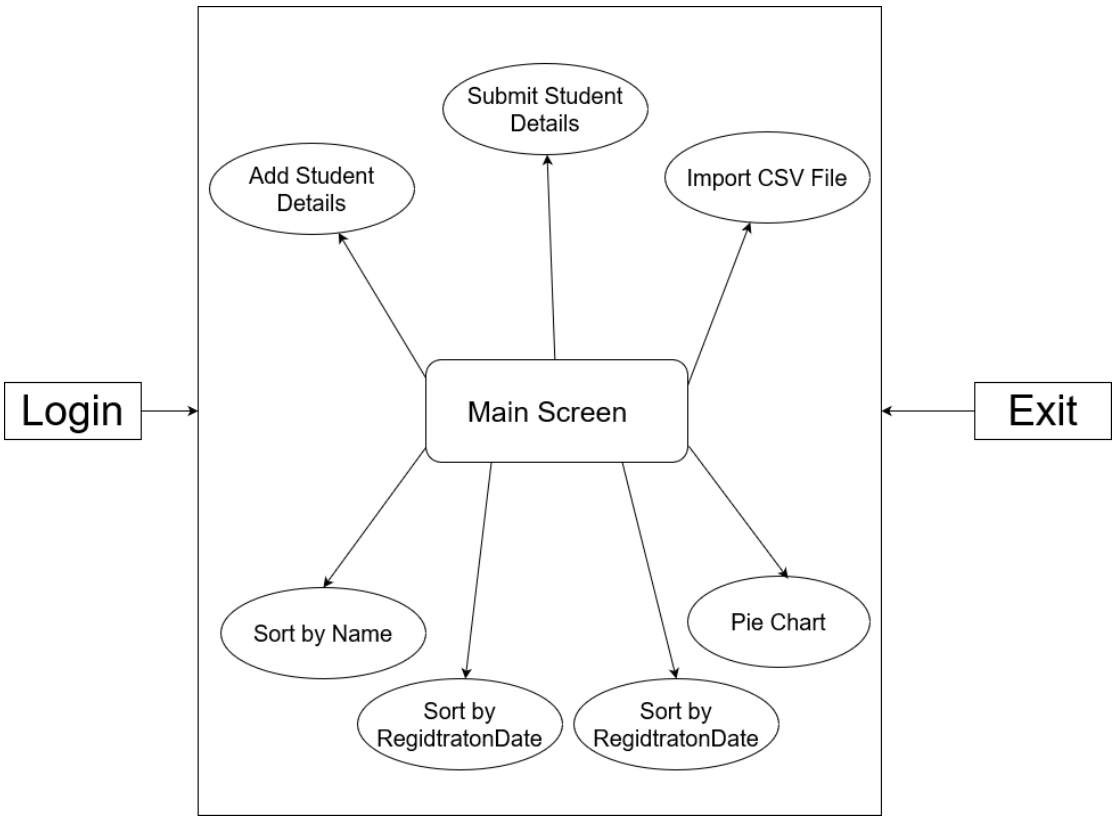



*Figure 19: Architecture Diagram*

Given above figure represents the architecture of the developed system. At first, user needs to login to the system for which the user to input the correct credentials. After logging into the system with correct credentials, the system will display the main form which is panel of developed system. Using the buttons stripe on the main form, the user can keep student details, contact number, email, address and other things. Moreover, the user can generate the weekly report of students in pie chart.

## 4.2 Class Diagram

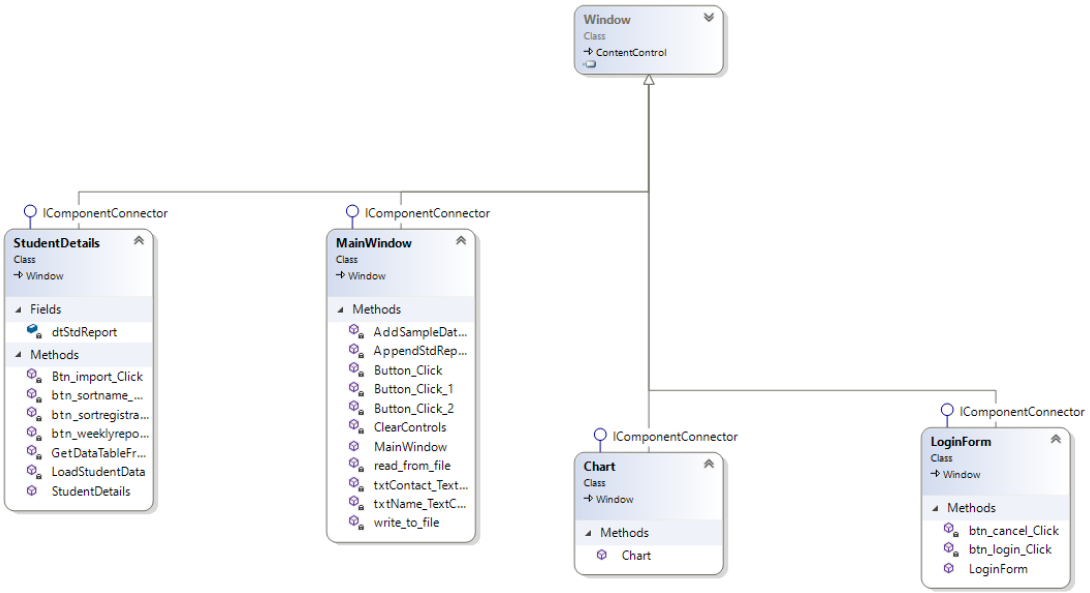



*Figure 20: Class Diagram*

## 4.3 Individual Diagram

- **Login Form**

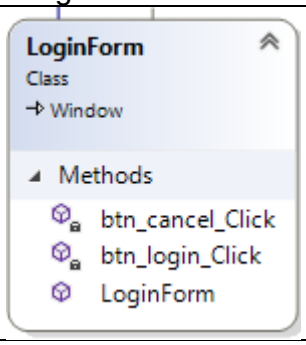
| Methods | Description | Diagram |
|---|---|---|
| btn_cancel_Click | Application will close. |  |
| btn_login_Click | Application operate next page. | |
| LoginForm | Security page. | |
| | | |



*Table 1: Individual Diagram of Login Form*

- Main Window (Add Students Details)

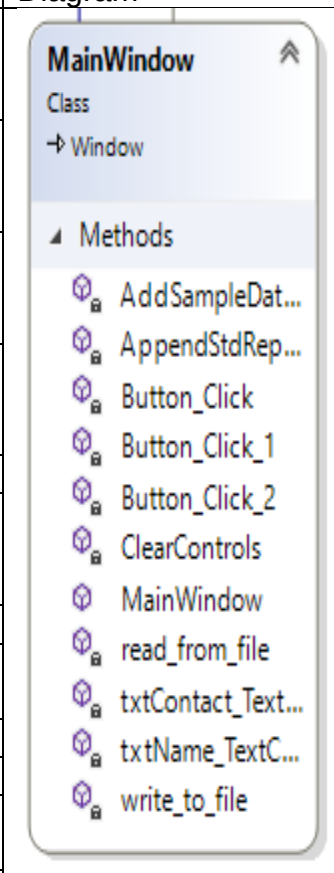
| Methods | Description | Diagram |
|---|---|---|
| AddSampleDataforStd() | This method exports individual data in xml. |  |
| AppendStdReprt() | This method exports all data in xml which is input by user. | |
| Button_Click() | Submit button which read all data and valid input data. | |
| Button_Click_1() | Add all student details which is input by users. | |
| Button_Click_2() | Clear all textbox. | |
| ClearControls() | Clear all textbox after pressing submit buton. | |
| MainWindow() | Display all GUI. | |
| Read_From_File | Read all data and show on table. | |
| txtContact_TextChange() | For Validation. | |
| txtName_TextChange() | For Validation. | |
| Write_to_file() | Write all data and show on table. | |
| | | |



*Table 2: Individual diagram of Main window (Add Students Data)*

- **Student Details**

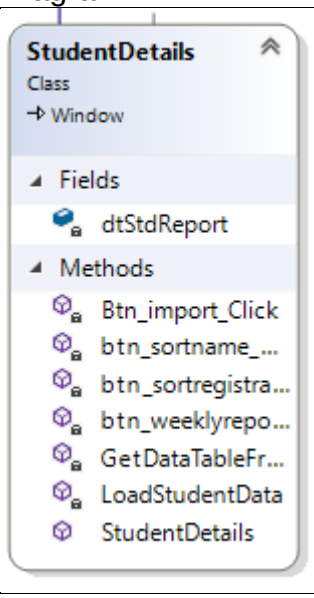
| Method | Description | Diagram |
|---|---|---|
| Btn_import_Click_Button() | Import CSV file. |  |
| Btn_sortname_Click_Button() | Sort by name. | |
| Btn_sortregistration_Click_Button() | Sort by registration date. | |
| Btn_weeklyreport_Click_Button | Generate weekly report. | |
| GetDataFromCSVFile() | Import CSV file. | |
| LoadStudentData() | Load all data in table. | |
| | | |



*Table 3: Individual diagram of Students Details*

- **Chart**

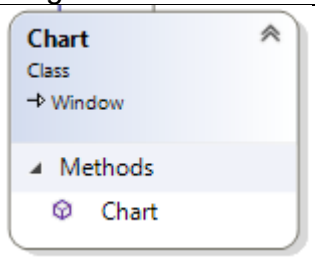
| Method | Description | Diagram |
|---|---|---|
| Chart() | To display total number students its shows the pie-chart. |  |



*Table 4: Individual diagram of Chart*

## 4.4 Flow-Chart of Enrol Student

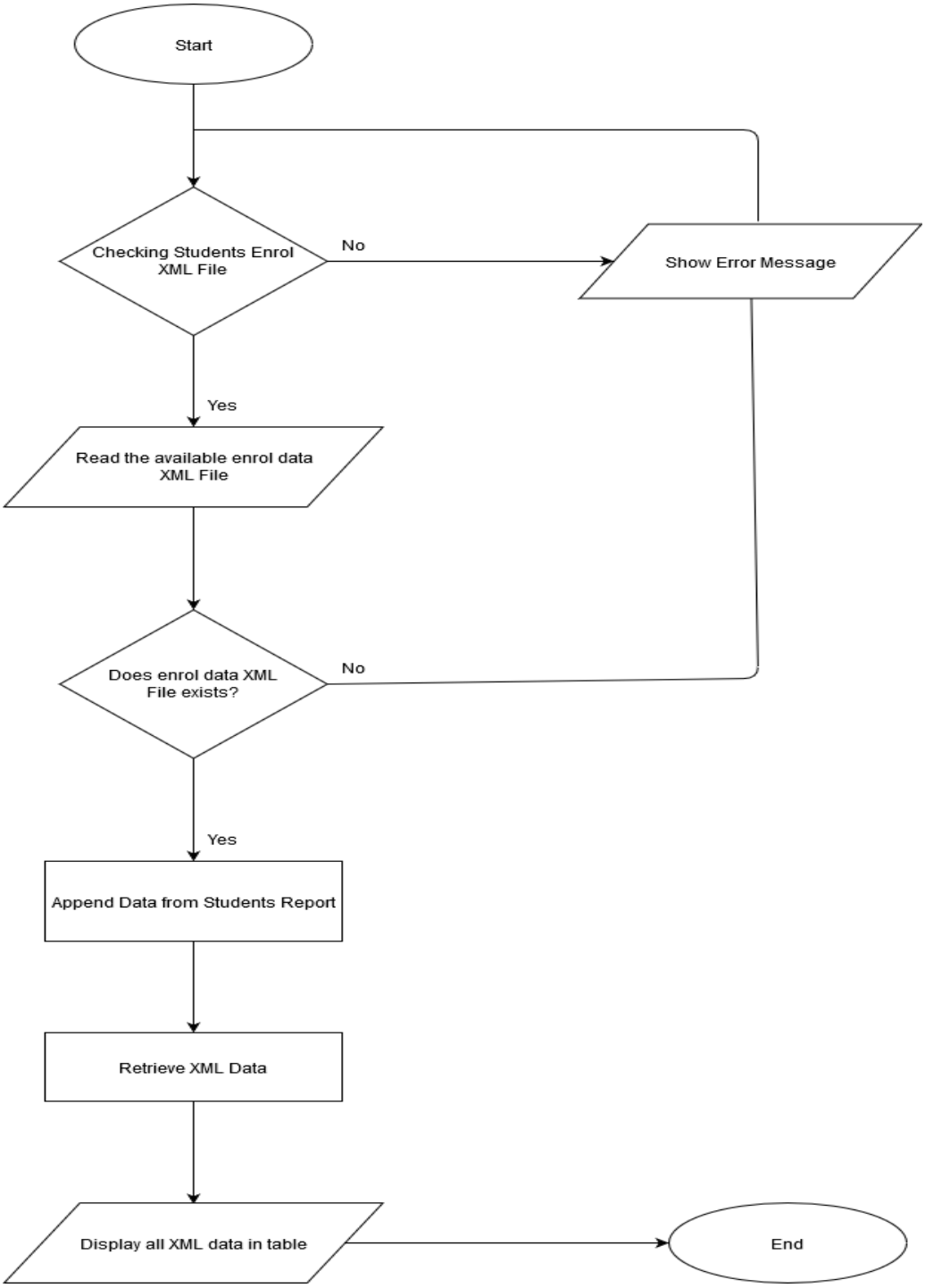



*Figure 21: Flow-Chart of Enrol Students*

## 4.5 Flow-Chart of Import Data from CSV Flow Chart

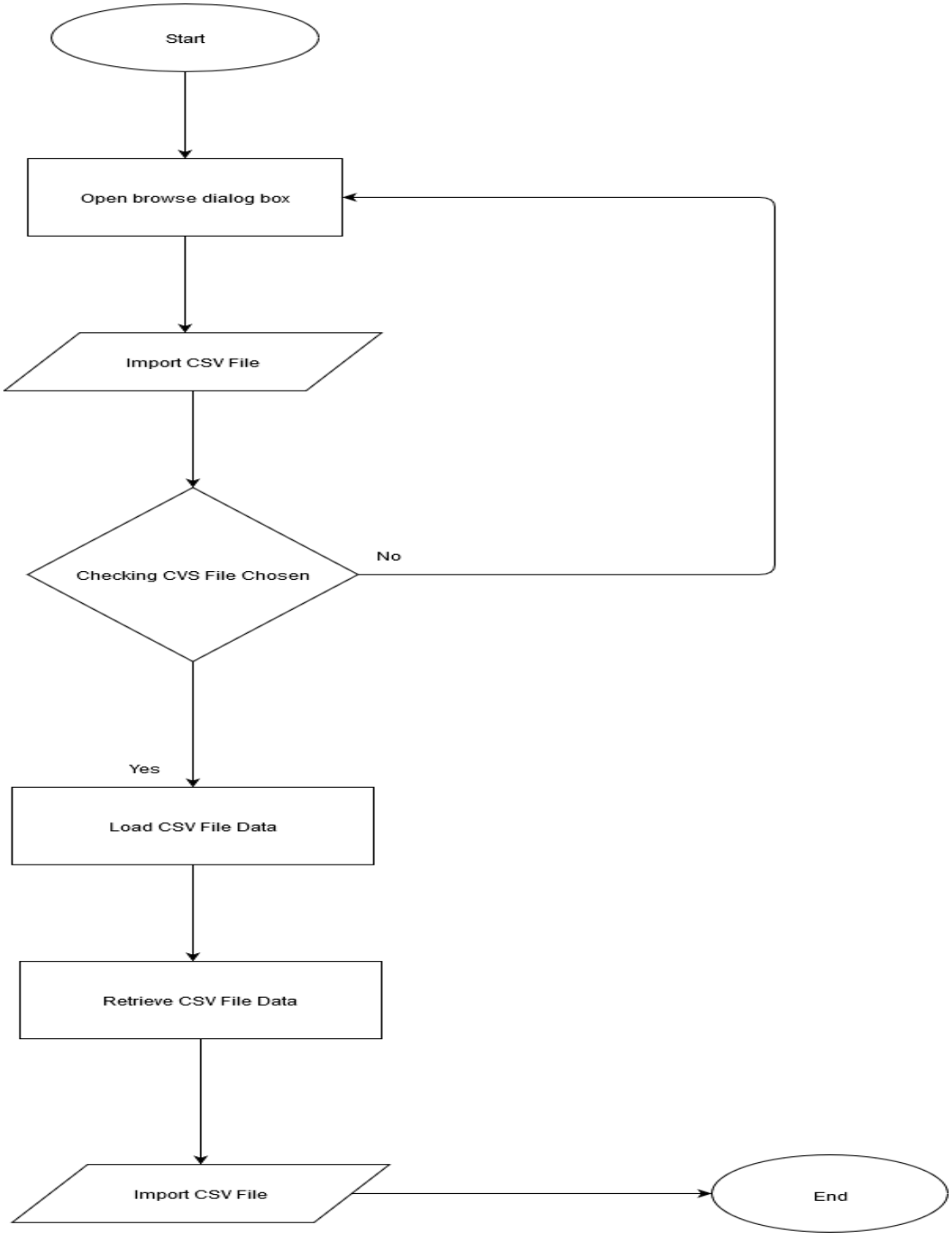



*Figure 22: Flow-Chart of Import Data from CSV Flow Chart*

## 5. Sorting Algorithm

The sorting Algorithm used in the School Management System is bubble sorting algorithm.

A bubble sort is an internal exchange sort. It is considered one of the simplest methods to sort an array of objects. It is also known as a sinking sort (because the smallest items "sink" to the bottom of the array).

Instead of searching an array as a whole, the bubble sort works by comparing adjacent pairs of objects in the array. If the objects are not in the correct ordered, they are swapped so that the largest of the two moves up. This process continues until the largest of the objects, eventually "bubbles" up to the highest position in the array. After this occurs, the search for the next largest object begins. The swapping continues until the whole array is in the correct order.

- **Working Mechanism:**

  We have ten elements in an array. That means we need to make 9 comparisons so that the largest element will bubble to the top of the array. Why 9 comparisons?

  n = the number of elements in the array
  n - 1 = the number of times the comparison takes place
  Therefore: 10 - 1 = 9.

- **Step 1:** Compare the first two elements '4' and '7'.
  No switch is necessary because '4' is smaller than '7'.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **4** | **7** | **2** | **1** | **10** | **8** | **3** | **5** | **6** | **9** |

- **Step 2:** Compare second and third elements '7' and '1'.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 4 | **2** | **7** | **1** | **10** | **8** | **3** | **5** | **6** | **9** |

  Switch, because '7' is larger than '2'.

- **Step 3:** Compare third and fourth elements '7' and '1'.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **4** | **2** | **1** | **7** | **10** | **8** | **3** | **5** | **6** | **9** |

  They had to switch.

- **Step 4:** Compare fourth and fifth elements '7' and '10'.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **4** | **2** | **1** | **7** | **10** | **8** | **3** | **5** | **6** | **9** |

  No switch, because they are in order.

- **Step 5:** Compare fifth and sixth elements '10' and '8'.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **4** | **2** | **1** | **7** | **8** | **10** | **3** | **5** | **6** | **9** |

  They had to switch.

- **Step 6:** Compare sixth and seventh elements '10' and '3'.

| 4 | 2 | 1 | 7 | 8 | 3 | 10 | 5 | 6 | 9 |
|---|---|---|---|---|---|---|---|---|---|

They had to switch.

- **Step 7:** Compare seventh and eighth elements '10' and '5'.

| 4 | 2 | 1 | 7 | 8 | 3 | 5 | 10 | 6 | 9 |
|---|---|---|---|---|---|---|---|---|---|

They had to switch.

- **Step 8:** Compare eighth and ninth elements '10' and '6'.

| 4 | 2 | 1 | 7 | 8 | 3 | 5 | 6 | 10 | 9 |
|---|---|---|---|---|---|---|---|---|---|

They had to switch.

- **Step 9:** Compare ninth and tenth elements '10' and '9'.

| 4 | 2 | 1 | 7 | 8 | 3 | 5 | 6 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|

They had to switch. So, now you see that '10', the largest element has "bubbled" to the last place in the array.

Now, we have to compare the remaining 9 elements until all are in order.

## 6. Reflection

Developing the system in Microsoft Visual Studios 2019 keeping C# as primary programming language isn't new experience for me. But developing in C# environment is new for me. Developing a record keeping system for School Management System is really a tough task though. Serialization and deserialization are another new thing while developing the system. Though, creating new classes and methods helps to pace the development task. Importing and exporting of CSV file is also a new task and it really help me in gaining knowledge of file handling. Creating a class diagram within the visual studio helps me in documentation phase. With the growing of technology, the visual studio and its community helps newbie developer like us to pace our development speed.

With this experience I had got some plus point while doing this coursework. I came to get more working and experience with the language. Features like creating chart generating list. In addition to that, sorting of data form the grid was a new thing for me. Furthermore, import and exporting to CSV file was new aspect for me. Overall evaluation with the great support of Mr. Sachin Subedi and module leader help me to push the code on Git which was totally new for me hence, it will help me to know importance of Git in real-world. I had a great experience with the Application Development of the School Management System.

## 7. Conclusion

The School Management system is digitalized system. It is developed using Visual Studio 2019 with the C# language. The business logic used in the system refers to the real working environment of School. The GUI designed is highly user-friendly interface and user with basic system administration can operate the system. The framework has login screen to add security to the task. After login, the framework shows a primary screen where each and every functionality and features are found. Aside from various shape components, class outline for every one of the structures and classes were used. The concept of OOP that were gained in first year came in handy in this project. But then the exciting part was also there. Researching and creating new ideas were the exciting part to it. So, this was a great opportunity for us to create our own application and implement knowledge of OOP.

In addition, this WPF App (.NET Framework) is totally new for me. Finally, I would like to give great thank to Mr. Sachin Subedi and module teacher for guiding me throughout the project without any problems. This coursework helped to improve our programming skills and do tasks creatively.

# 8. Bibliography

Altınay, F. (2016). Digital Transformation in School Management and Culture. *Digital Transformation in School Management and Culture*(2020).

Babaheidari, S. M. (2014). ResearchGate. *Managing the Digitalization of Schools: an exploratory study of school principals' and IT managers' perceptions about ICT adoption and usefulness*(2020).

Breiter, A. (2018). ResearchGate. *Governing the Figurations of Mediatized Schools in England and Germany*(2020).

Demir, K. (2006). RsearchGate. *School management information systems in primary schools*(2020).

Zeng, D. (2010). ACM Digital liabrary. *ACM Transactions on Management Information Systems (TMIS)*(2020).

# 9. Appendix

## ➢ Login Page

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

     private void btn_login_Click(object sender, RoutedEventArgs e)
     {
         if (txt_username.Text != "admin")
         {
             MessageBox.Show("Username is incorrect!", "Alert",
MessageBoxButton.OK, MessageBoxImage.Error);
             txt_username.Clear();

         }
         else if (txt_password.Password != "admin")
         {
             MessageBox.Show("Password is incorrect!", "Alert",
MessageBoxButton.OK, MessageBoxImage.Error);
             txt_password.Clear();
         }
         else
         {
             MessageBox.Show("Logged in Successfully !!!", "Success",
MessageBoxButton.OK, MessageBoxImage.Information);
             MainWindow mainForm = new MainWindow();
             mainForm.Show();


         }
     }



     private void btn_cancel_Click(object sender, RoutedEventArgs e)
```

```
        {
            Process.GetCurrentProcess().Kill();

        }
    }
}
```

- **Main Window**

```
    using System;
    using System.Windows;
    using System.Data;
    using DataHandler;
    using System.IO;
     using System.Text.RegularExpressions;
     using System.Linq;

namespace CourseWorkSample
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {

        public MainWindow()
        {
            InitializeComponent();


            txtRegNo.Text = read_from_file();



        }


        private void AddSampleDataforStd(DataSet dataSet)
        {



            var dr1 = dataSet.Tables["Student"].NewRow();
            dr1["Name"] = txtName.Text;
            dr1["Address"] = txtAddress.Text;
            dr1["ContactNo"] = txtContact.Text;
            dr1["CourseEnroll"] = comboBox.Text;
            dr1["RegistrationDate"] = DateTime.Today.AddDays(-2);
            dataSet.Tables["Student"].Rows.Add(dr1);
```

```
        }

        private void AppendStdReport(DataSet dataSet)
        {

            var handler = new Handler();

            if (File.Exists(@"D:\StudentReport.xml"))
            {



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

                var dr2 = dataSet.Tables["StudentReport"].NewRow();
                dr2["RegNo"] = txtRegNo.Text;
                dr2["Name"] = txtName.Text;
                dr2["Address"] = txtAddress.Text;
                dr2["ContactNo"] = txtContact.Text;
                dr2["CourseEnroll"] = comboBox.Text;
                dr2["Email"] = txt_email.Text;
                DateTime? selectedDate = dateselect.SelectedDate;
                if (selectedDate.HasValue)
                {
                    string date = selectedDate.Value.ToString("dd/MM/yyyy");
                    dr2["RegistrationDate"] = date;
                }
                dataSet.Tables["StudentReport"].Rows.Add(dr2);
            }
            else
            {


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



        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
             if (txtName.Text == "")
            {
                string myStringVariable1 = string.Empty;
                MessageBox.Show("All Field is requierd");
                return;

            }
            else if (txtContact.Text == "")
            {
                string myStringVariable2 = string.Empty;
                MessageBox.Show("Contact Number is requierd");
                return;

            }
            else if (txt_email.Text == "")
            {
                string myStringVariable2 = string.Empty;
```

```
                MessageBox.Show("Email Address is requierd");
                return;


            }
            else if (txtAddress.Text == "")
            {
                string myStringVariable2 = string.Empty;
                MessageBox.Show("Address is requierd");
                return;


            }


            else if (comboBox.Text == "--Choose--")
            {
                string myStringVariable3 = string.Empty;
                MessageBox.Show("Select Corse Enroll");
                return;

            }
            else if (!this.txt_email.Text.Contains('@') ||
!this.txt_email.Text.Contains('.'))
            {
                MessageBox.Show("Please Enter A Valid Email", "Invalid Email
Input someone@someting.com", MessageBoxButton.OK, MessageBoxImage.Error);
                return;


            }

            var handler = new Handler();
            var dataSet = handler.CreateDataSet();
            AddSampleDataforStd(dataSet);
            AppendStdReport(dataSet);

            var regno = txtRegNo.Text;
            var name = txtName.Text;
            //dataSet.WriteXmlSchema(@"D:\StudentCWSchema1.xml");
            dataSet.Tables["StudentReport"].WriteXml(@"D:\" + name + "CWData" +
regno + ".xml");

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

            write_to_file(txtRegNo.Text);

            txtRegNo.Text = read_from_file();

            ClearControls();


            MessageBox.Show("Submit Sucessfully");

        }

        private void write_to_file(string text)
        {


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

```
        }
        private string read_from_file()
        {
            int i = 0001;
            if (File.Exists(@"D:\StudentReport.xml"))
            {

                string text = System.IO.File.ReadAllText(@"D:\count.txt");
                i = int.Parse(text.ToString());
                i = i + 1;
            }
            else
            {
                System.IO.File.WriteAllText(@"D:\count.txt", "23000");
            }

            return i.ToString();

        }

        private void ClearControls()
        {
            txtName.Text = "";
            txtAddress.Text = "";
            txtContact.Text = "";
            txt_email.Text = "";




        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            ClearControls();


        }

        private void Button_Click_2(object sender, RoutedEventArgs e)
        {
            StudentDetails studentDetails = new StudentDetails();
            studentDetails.Show();


        }

        private void txtName_TextChanged(object sender,
System.Windows.Controls.TextChangedEventArgs e)
        {
            Regex regex = new Regex("[^a-zA-Z]+");
            if (regex.IsMatch(txtName.Text))
            {
                MessageBox.Show("Invalid Input !");
            }
        }

        private void txtContact_TextChanged(object sender,
System.Windows.Controls.TextChangedEventArgs e)
        {
```

```
            Regex regex = new Regex("[^0-9]+");
            if (regex.IsMatch(txtContact.Text))
            {
                MessageBox.Show("Input Only Number !");
            }
        }


    }

}
```

- **StudentDetails**

```
    using DataHandler;
    using Microsoft.Win32;
    using System;
    using System.Collections.Generic;
    using System.Data;
    using System.Data.OleDb;
    using System.Globalization;
    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.Forms;
    using System.Windows.Input;
    using System.Windows.Media;
    using System.Windows.Media.Imaging;
    using System.Windows.Shapes;
    using MessageBox = System.Windows.Forms.MessageBox;
    using OpenFileDialog = System.Windows.Forms.OpenFileDialog;
    using Path = System.IO.Path;

namespace CourseWorkSample
{

    /// <summary>
    /// Interaction logic for StudentDetails.xaml
    /// </summary>
    public partial class StudentDetails : Window
    {
        DataTable dtStdReport;

        //public object BrowseTextBox { get; private set; }

        public StudentDetails()
        {
            InitializeComponent();
            LoadStudentData();
        }



        private void LoadStudentData()
        {
```

```
            if (System.IO.File.Exists(@"D:\StudentReport.xml"))
            {

                var dataSet = new DataSet();
                dataSet.ReadXml(@"D:\StudentReport.xml");

                //DataTable dtStdReport = new DataTable("dt");
                dtStdReport = dataSet.Tables[0];
                grdStd.DataContext = dtStdReport.DefaultView;
            }

        }

        private void btn_weeklyreport_Click(object sender, RoutedEventArgs e)
        {



            int sum_computing = 0;
            int sum_mediatechnology = 0;
            int sum_networksanditsecurity = 0;
            int sum_cybersecurityandethicalhacking = 0;

            DataTable dtable = new DataTable("tbl");
            dtable.Columns.Add("Course Enroll", typeof(String));
            dtable.Columns.Add("Sum Students", typeof(int));


            for (int i = 0; i < dtStdReport.Rows.Count; i++)
            {
                String col = dtStdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    sum_computing++;
                }
                else if (col == "Multimedia Technology")
                {
                    sum_mediatechnology++;
                }
                else if (col == "Networking and IT Security")
                {
                    sum_networksanditsecurity++;
                }
                else if (col == "Cyber Security and Ethical Hacking")
                {
                    sum_cybersecurityandethicalhacking++;
                }

            }

            dtable.Rows.Add("Computing", sum_computing);
            dtable.Rows.Add("Multimedia Technology", sum_mediatechnology);
            dtable.Rows.Add("Networking and IT Security",
sum_networksanditsecurity);
            dtable.Rows.Add("Cyber Security and Ethical Hacking",
sum_cybersecurityandethicalhacking);

            grdStd.DataContext = dtable.DefaultView;


        }
```

```
        private void btn_sortregistrationdate_Click(object sender,
RoutedEventArgs e)
        {
            if (System.IO.File.Exists(@"D:\StudentReport.xml"))
            {

                var dataSet = new DataSet();
                dataSet.ReadXml(@"D:\StudentReport.xml");

                //DataTable dtStdReport = new DataTable("dt");
                dtStdReport = dataSet.Tables[0];
                dtStdReport.DefaultView.Sort = "RegistrationDate ASC";
                grdStd.DataContext = dtStdReport.DefaultView;

            }
            MessageBox.Show("Sucessfully Import CSV File and Merge with XML
File");

        }

        private void btn_sortname_Click(object sender, RoutedEventArgs e)
        {
            if (System.IO.File.Exists(@"D:\StudentReport.xml"))
            {

                var dataSet = new DataSet();
                dataSet.ReadXml(@"D:\StudentReport.xml");

                //DataTable dtStdReport = new DataTable("dt");
                dtStdReport = dataSet.Tables[0];
                dtStdReport.DefaultView.Sort = "Name ASC";
                grdStd.DataContext = dtStdReport.DefaultView;
            }
            MessageBox.Show("Sucessfully Import CSV File and Merge with XML
File");

        }
        static DataTable GetDataTableFromCsv(string path, bool
isFirstRowHeader)
        {
            string header = isFirstRowHeader ? "Yes" : "No";

            string pathOnly = Path.GetDirectoryName(path);
            string fileName = Path.GetFileName(path);

            string sql = @"SELECT * FROM [" + fileName + "]";

            using (OleDbConnection connection = new OleDbConnection(
                      @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
pathOnly +
                      ";Extended Properties=\"Text;HDR=" + header + "\""))
            using (OleDbCommand command = new OleDbCommand(sql, connection))
            using (OleDbDataAdapter adapter = new OleDbDataAdapter(command))
            {
                DataTable dataTable = new DataTable();
                dataTable.Locale = CultureInfo.CurrentCulture;
                adapter.Fill(dataTable);
                return dataTable;
            }

        }
```

```
        private void Btn_import_Click(object sender, RoutedEventArgs e)
        {
            Microsoft.Win32.OpenFileDialog dlg = new
Microsoft.Win32.OpenFileDialog();
            dlg.DefaultExt = ".csv";
            Nullable<bool> result = dlg.ShowDialog();
            if (result == true)
            {
                DataTable tablestd = GetDataTableFromCsv(dlg.FileName, true);
                DataTable dataTable = tablestd.Clone();
                dataTable.Columns["ContactNo"].DataType = typeof(String);
                dataTable.Columns["Address"].DataType = typeof(String);
                dataTable.Columns["RegNo"].DataType = typeof(String);
                dataTable.Columns["RegistrationDate"].DataType =
typeof(String);
                foreach (DataRow row in tablestd.Rows)
                {
                    dataTable.ImportRow(row);
                }
                Handler handler = new Handler();
                DataSet dataSet = handler.CreateDataSet();

dataSet.Tables["StudentReport"].ReadXml(@"D:\StudentReport.xml");
                dataSet.Tables["StudentReport"].Merge(dataTable);

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

                var dataset1 = new DataSet();
                dataSet.ReadXml(@"D:\StudentReport.xml");
                DataTable table1 = dataset1.Tables["StudentReport"];
                grdStd.DataContext = tablestd.DefaultView;
            }




        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            MainWindow f1 = new MainWindow();
            f1.ShowDialog();
            this.Hide();
        }

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

- **Data Handler**

```
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", 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("Address", typeof(string));
            dt.Columns.Add("ContactNo", typeof(string));
            dt.Columns.Add("CourseEnroll", typeof(string));
            dt.Columns.Add("RegistrationDate", typeof(string));
            dt.Columns.Add("Email", typeof(string));


            dt.PrimaryKey = new DataColumn[] { dt.Columns["ID"] };
            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("RegNo", typeof(string));
            dt.Columns.Add("Name", 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(string));
            dt.Columns.Add("Email", typeof(string));



            return dt;
        }
    }
}
```