Marking Scheme

Informatics College Pokhara

informatics
college pokhara

Application Development
CS6004NI
Course Work 1

Submitted By: **Binita Poudel**
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	any one component is missing or inappropriate data is shown
Algorithm used for sorting & proper sorting of data	Default sorting provided by .net is used
B. Documentation	
User Manual for running the application	User Manual is below average. Is textual only.

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

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

Overall Comment:

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

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

School of Computing

Applications Development

Coursework 1

CS6004NP

Submitted by: Binita Poudel

Submitted to: Mr. Ishwor Sapkota

Student ID: 17030520

Group: L3C2

Date of Submission: 10th Janruary 2020

Table of Contents

List of Figure

1. Introduction

In this 20 first century, In the period of innovative change, the record keeping framework should be maintaining and should be kept secure. So, the conventional way of keeping record should presently be changed into digitalized frame. As per in case of student, the student's record keeping framework should be changed to digital frame. That's why, this venture is all approximately.

The Project is about Students Detail record keeping framework, where it keeps the record of the students who get enlisted to diverse course in specific date. Besides, the framework is competent to track the record of week by week report.

1.1 Current Scenario

Talking about the current scenario, most of the schools and colleges are still using the traditional ways to keep the record of student which are unsafe to maintain it for a long period of a time. It will be difficult to generate weekly report with traditional method.

1.2 Proposed System

The developed software can be used in schools, college to keep the record of the students. The main target of developing this software is to generate the weekly report of the student during the time of course enrolment. A user with little computer knowledge can run it easily.

2. User Manual

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

Login Screen

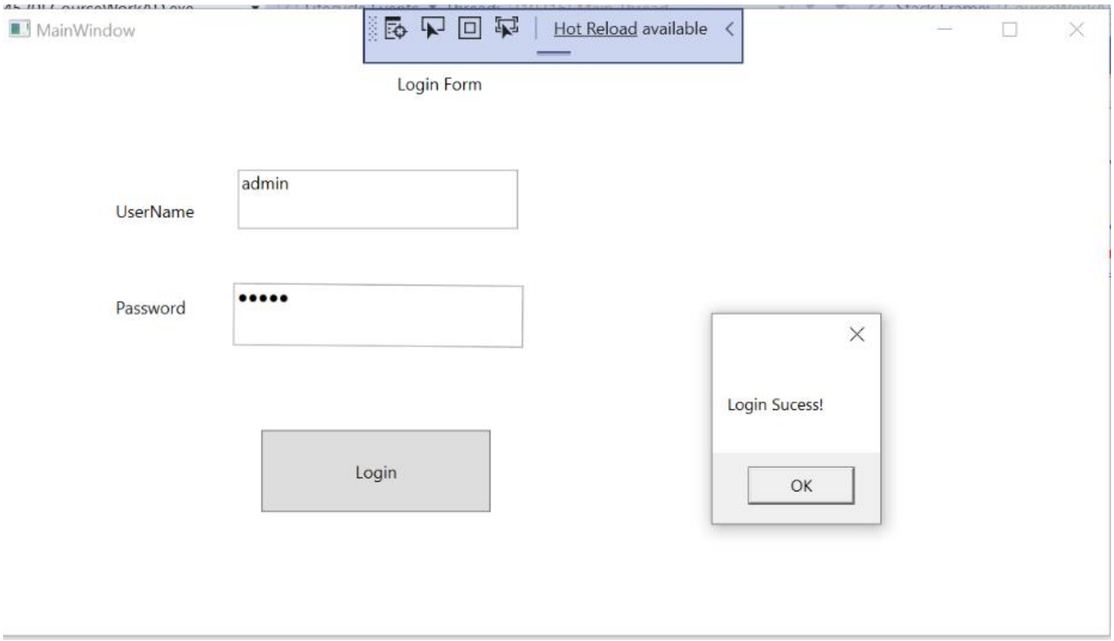


Figure 1 Login Screen

The username and password for this system is admin and admin respectively.

Student form Page

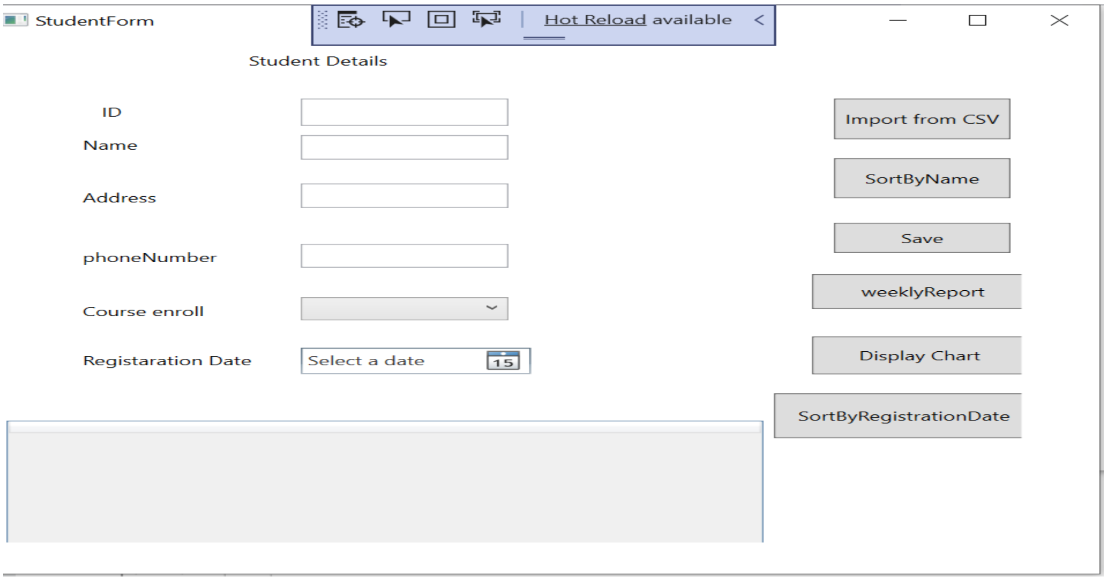


Figure 2 Student Detail Main Page

After logging into the system with correct credentials, the user will have six button to click for. These buttons perform different tasks which are described as follows:

Import from CSV: This CVS Button shows the data of .CVS file through the browser.

Sort by Name: This Button shows the data in grid with the alphabetical orders.

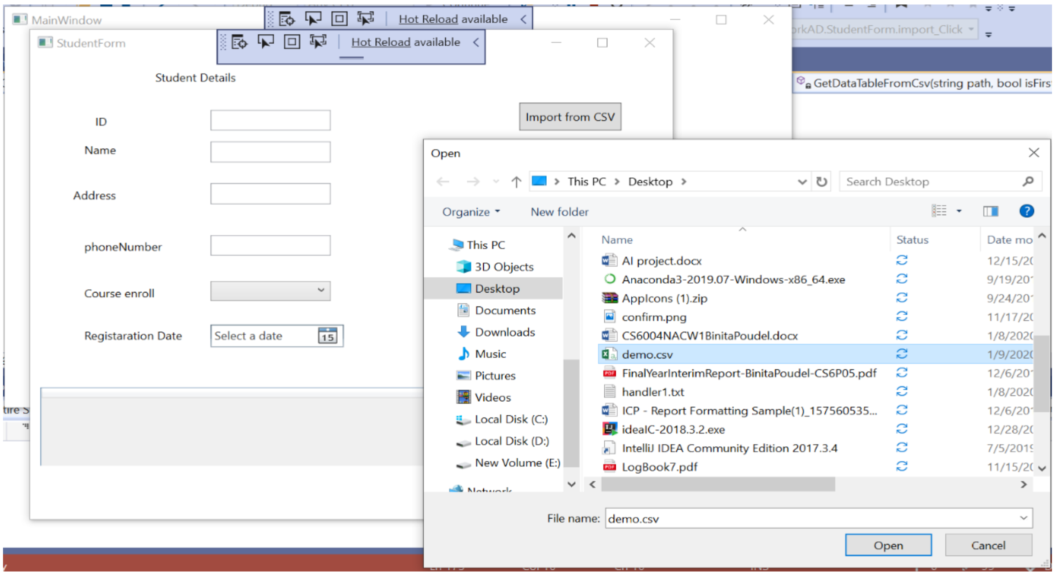
Save: The save button lets user to save the different detail of a student i.e. id, name, course, Contact and Registration date.

Weekly Report: The report button lets user to generate the report of weekly within a minutes.

Display Chart: The chart button lets user to view the chart of the total student who enrolled to the different course.

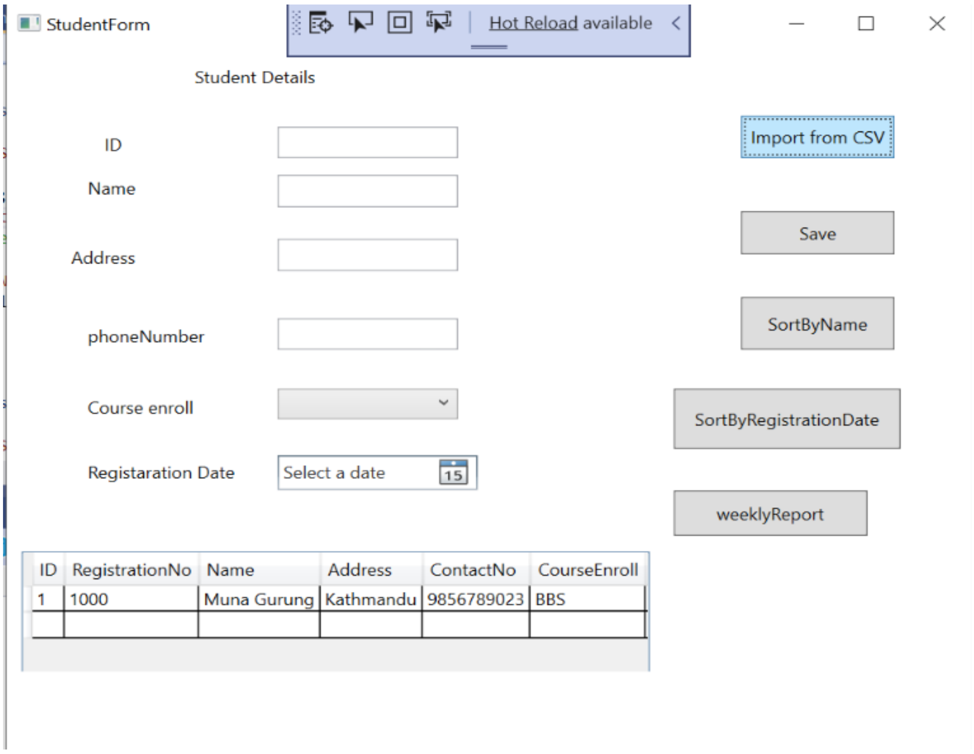
Sort by Registration Date: This button lets us to view the data of the student according to the registration date.

Import from the CSV Button



File name: demo.csv

The CVS file is named as demo.cvs.

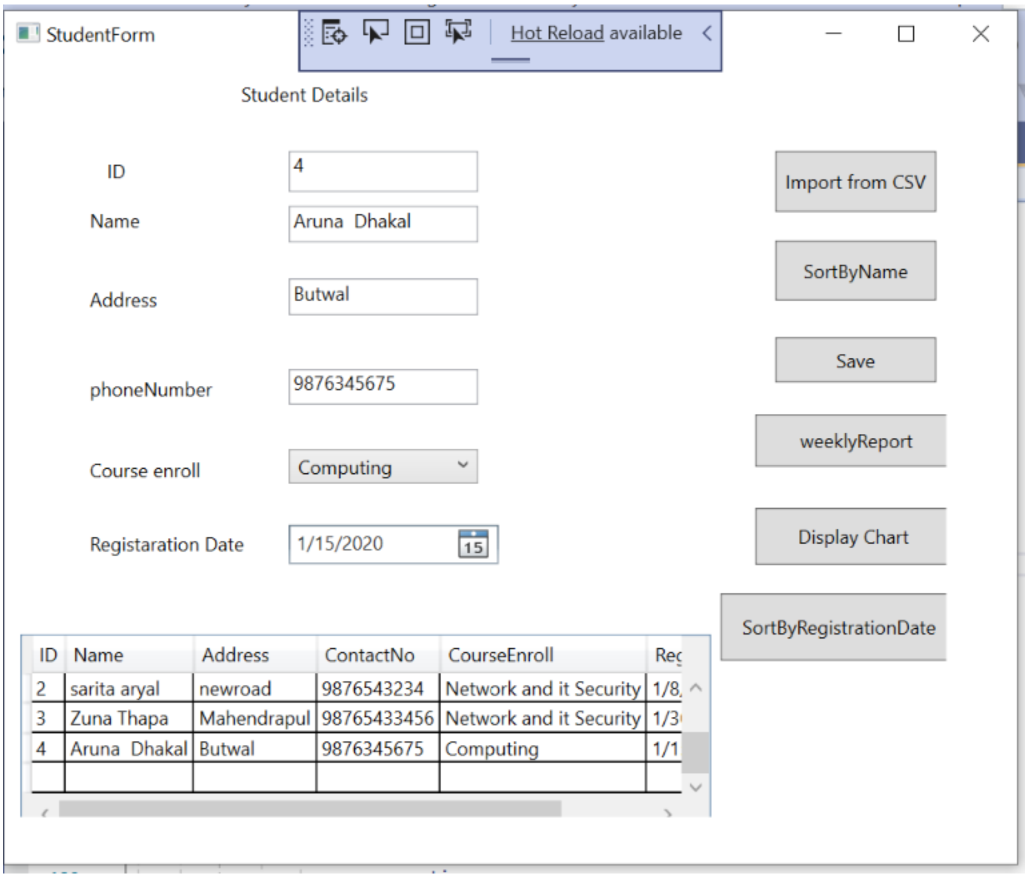


ID	RegistrationNo	Name	Address	ContactNo	CourseEnroll
1	1000	Muna Gurung	Kathmandu	9856789023	BBS

Figure 3 Data shown from CSV

The data imported from CVS file witch is named as demo.CSV file.

Sort by Name Button



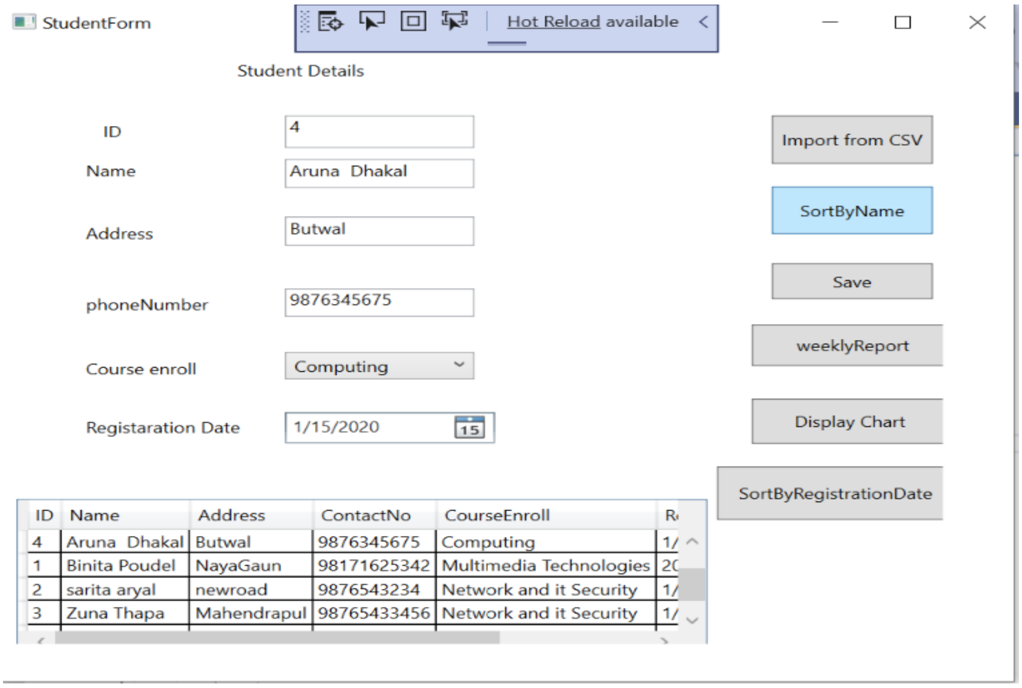
Student Details

ID	4
Name	Aruna Dhakal
Address	Butwal
phoneNumber	9876345675
Course enroll	Computing
Registaration Date	1/15/2020

ID	Name	Address	ContactNo	CourseEnroll	Reg
2	sarita aryal	newroad	9876543234	Network and it Security	1/8,
3	Zuna Thapa	Mahendrapul	98765433456	Network and it Security	1/3
4	Aruna Dhakal	Butwal	9876345675	Computing	1/1

Figure 4 Showing the unsorteddata from Sort by name

The data which was unsorted before.



Student Details

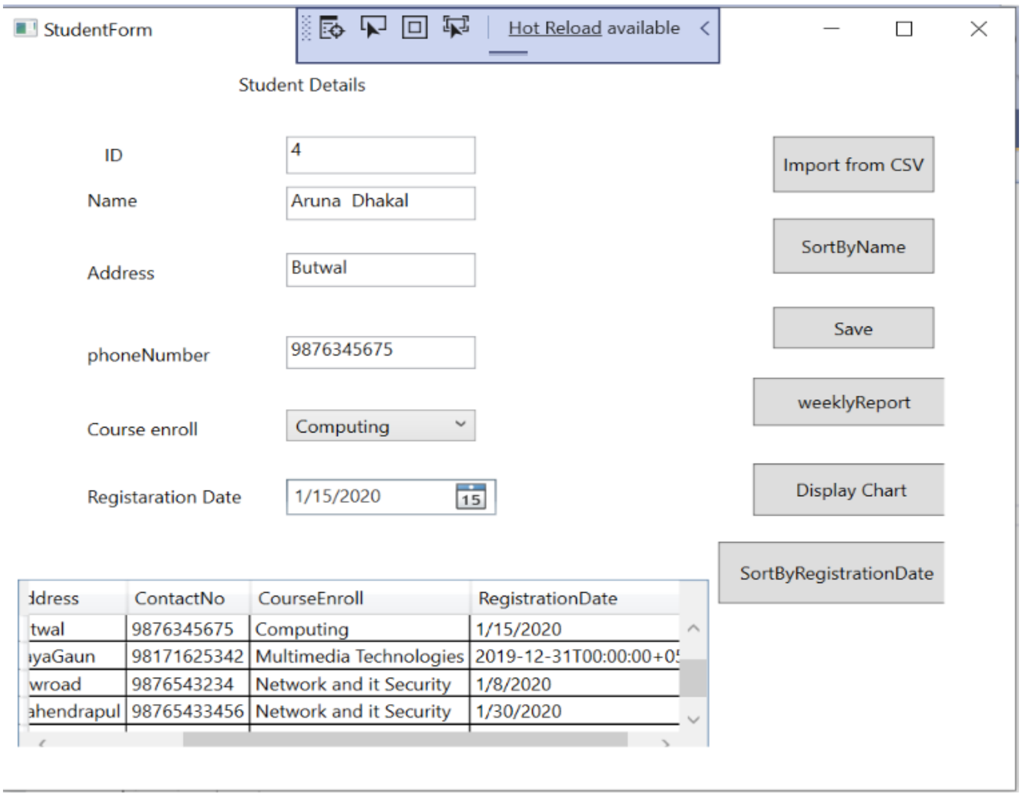
ID	4
Name	Aruna Dhakal
Address	Butwal
phoneNumber	9876345675
Course enroll	Computing
Registaration Date	1/15/2020

ID	Name	Address	ContactNo	CourseEnroll	R
4	Aruna Dhakal	Butwal	9876345675	Computing	1/
1	Binita Poudel	NayaGaun	98171625342	Multimedia Technologies	20
2	sarita aryal	newroad	9876543234	Network and it Security	1/
3	Zuna Thapa	Mahendrapul	98765433456	Network and it Security	1/

Figure 5 Sorted data shown by sort by name

The data which was sorted by the name.

Sort by Registration Date Button



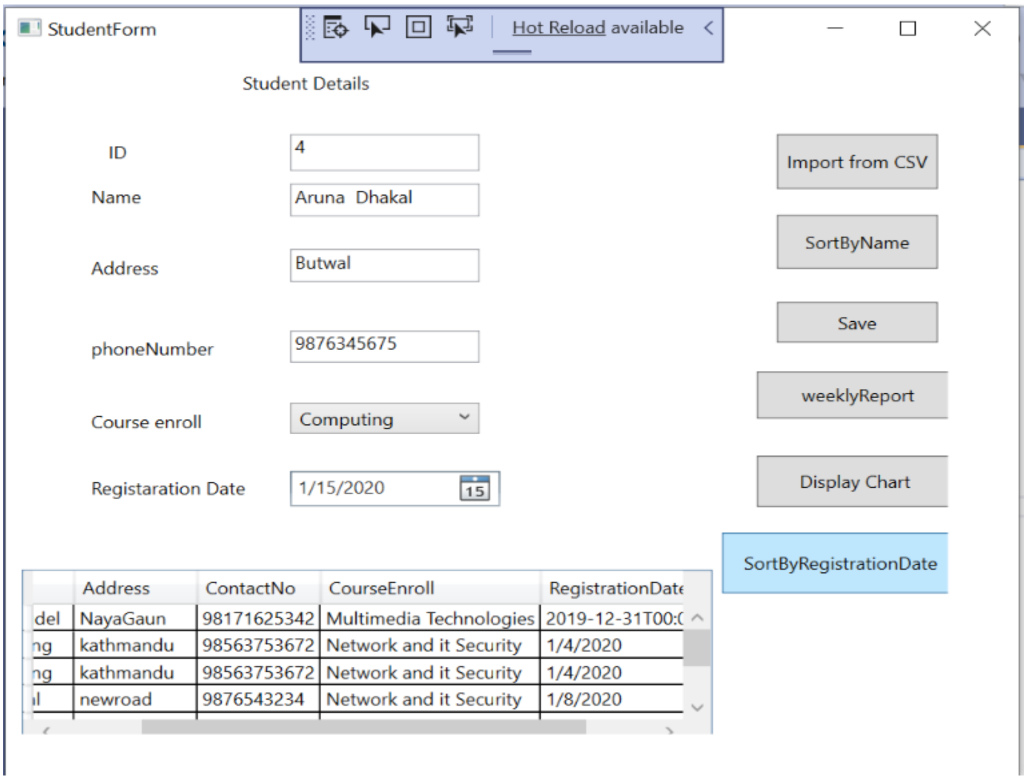
Student Details

ID	4
Name	Aruna Dhakal
Address	Butwal
phoneNumber	9876345675
Course enroll	Computing
Registaration Date	1/15/2020

ddress	ContactNo	CourseEnroll	RegistrationDate
twal	9876345675	Computing	1/15/2020
iyaGaun	98171625342	Multimedia Technologies	2019-12-31T00:00:00+0
wroad	9876543234	Network and it Security	1/8/2020
ahendrapul	98765433456	Network and it Security	1/30/2020

Figure 6 The unsorted data shown by sort by registration date

The unsorted Registration data.



Student Details

ID: 4

Name: Aruna Dhakal

Address: Butwal

phoneNumber: 9876345675

Course enroll: Computing

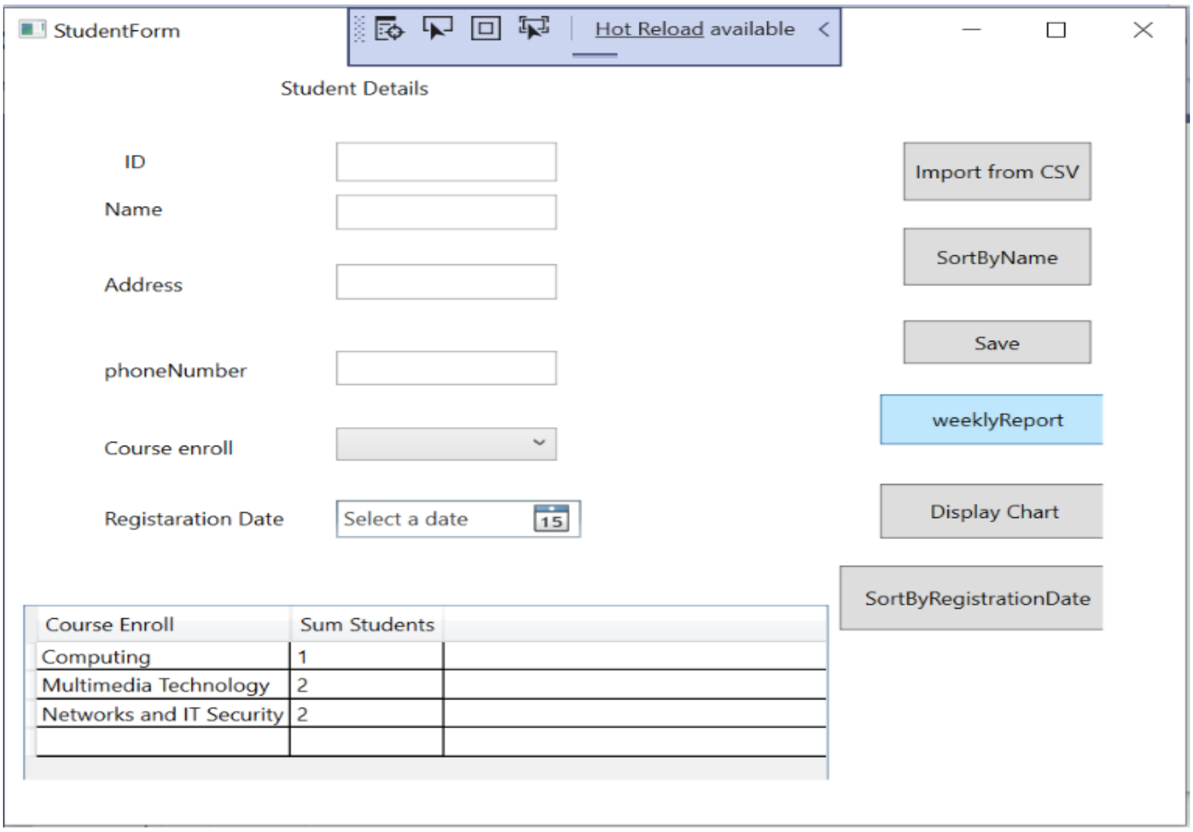
Registaration Date: 1/15/2020

	Address	ContactNo	CourseEnroll	RegistrationDate
del	NayaGaun	98171625342	Multimedia Technologies	2019-12-31T00:(
ng	kathmandu	98563753672	Network and it Security	1/4/2020
ng	kathmandu	98563753672	Network and it Security	1/4/2020
il	newroad	9876543234	Network and it Security	1/8/2020

Figure 7 The sorted data shown by sort by registration date

The Data Shows Sorted Registration Date.

Weekly Report Button

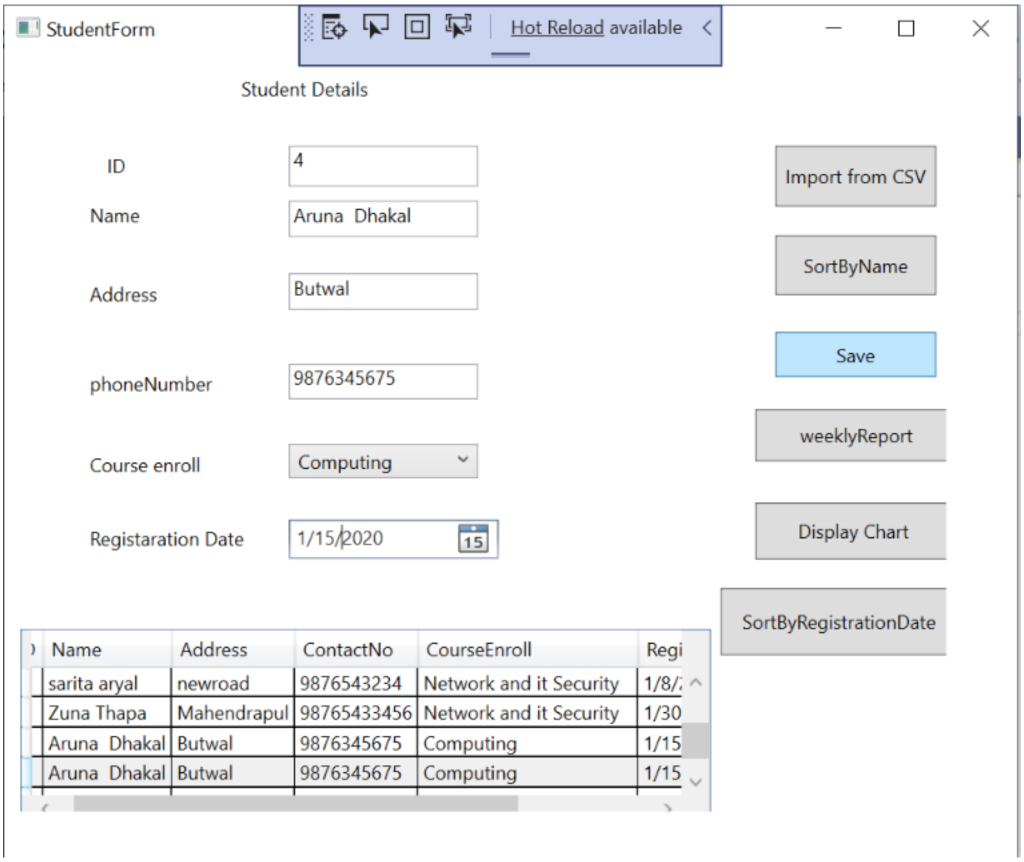


Course Enroll	Sum Students
Computing	1
Multimedia Technology	2
Networks and IT Security	2

Figure 8 It shows the report of week

The weekly report shows the total number of student who got enrolled to the different subjects.

Save Button



Student Details

ID 4

Name Aruna Dhakal

Address Butwal

phoneNumber 9876345675

Course enroll Computing

Registaration Date 1/15/2020

Name	Address	ContactNo	CourseEnroll	Regi
sarita aryal	newroad	9876543234	Network and it Security	1/8/:
Zuna Thapa	Mahendrapul	98765433456	Network and it Security	1/30
Aruna Dhakal	Butwal	9876345675	Computing	1/15
Aruna Dhakal	Butwal	9876345675	Computing	1/15

Figure 9 Saved Data

The data has been saved.

Display Chart Button

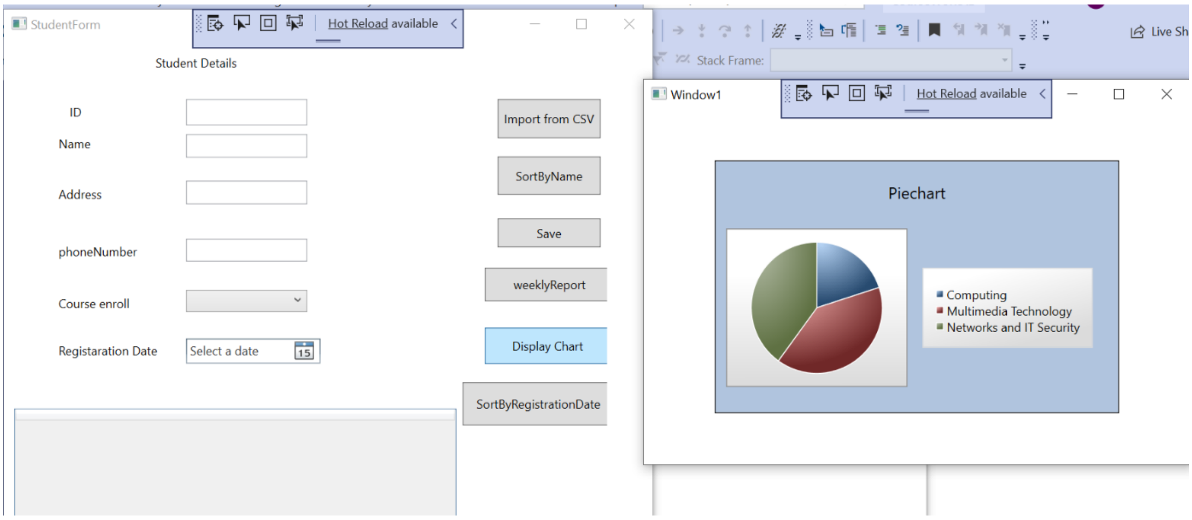


Figure 10 Displaying the chart.

The report of the students who got enrolled in different subject have shown in chart in above figure.

3.Jornal Articles

A number of problems associated with student academic record management include improper course registration, late release of students' results, inaccuracy due to manual and tedious calculation and retrieval difficulties/inefficiency.

Digital Repository and Automated Results Processing (DRARP) system provides an efficient means of processing, preserving and displaying students' results, academic records and other relevant notices to students. As part of its benefits, DRARP enables stress-free, speed-up the processing of students' examination results and eradicate vandalization of students' records that have characterized the traditional method

Development of an automated mail management system (MMS) is to replace the current paper records. While the Universities in developed countries already have a well-established administrative management system, some of their counterparts in the developing countries like Nigeria still adopt the manual or semi-automated method which provides only some basic elements of mail management and is time consuming, demanding and are often prone to varieties of errors and disaster (www.researchgate.net, 2020)

4.System Architecture

The figure represents the architecture of the developed system. At first, user needs to login to the system for which the user needs to input the correct credentials. After logging into the system with correct credentials, the system will display the main form which is the main panel of the students Details. Using the button on the main form, the user can record the students name, registration time and date. along with the date the user can record the student name, contact number and address. Moreover, the user can generate the report of the student in chart.

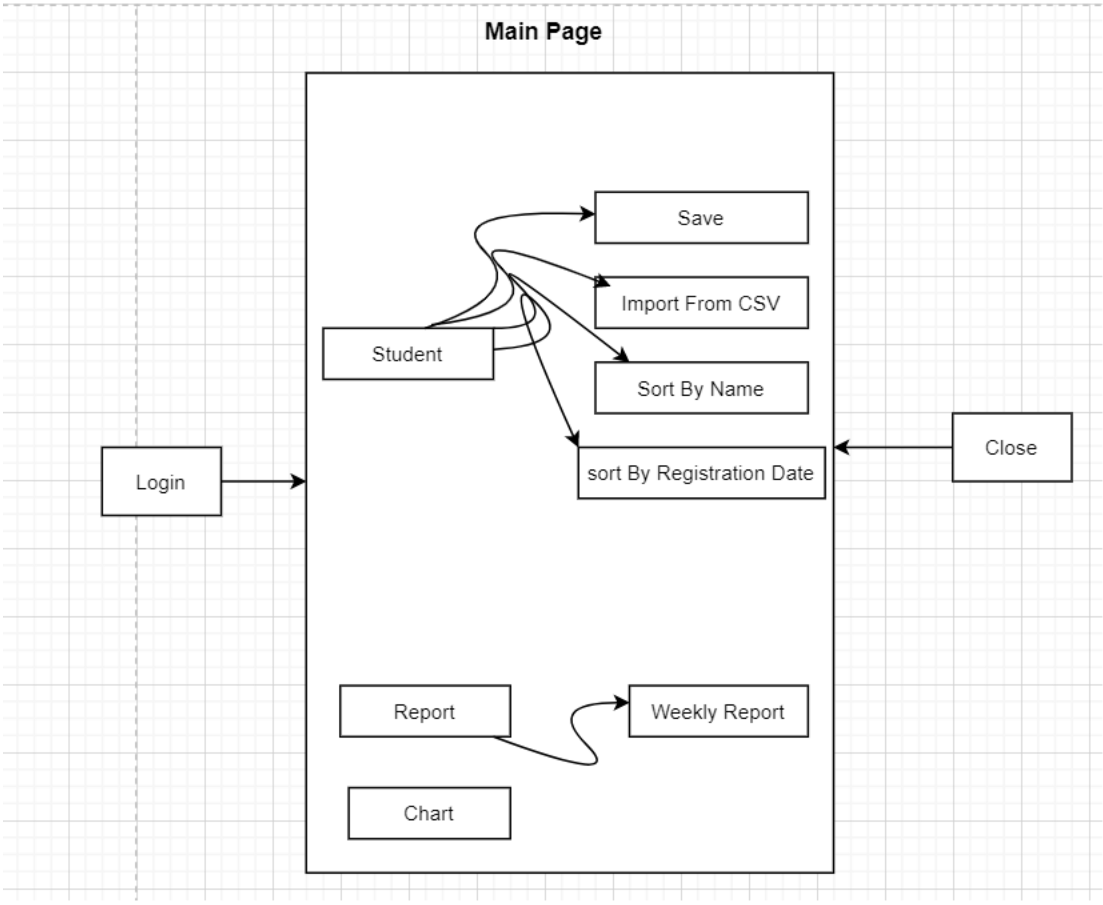


Figure 11 system Architecture

4.1 Class Diagram

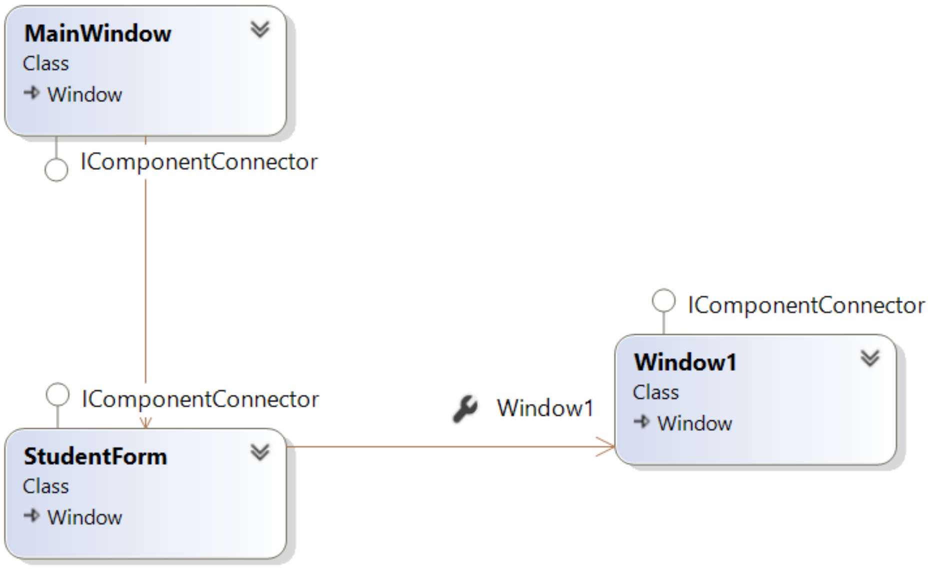


Figure 12 class diagram

4.3Flowchart for Report

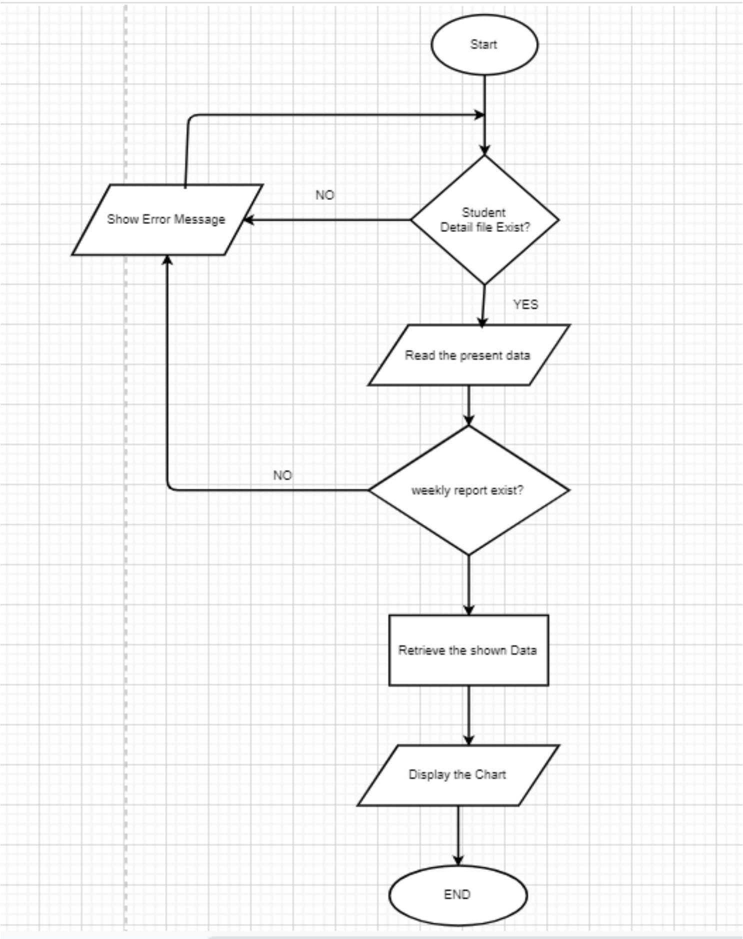


Figure 13 Flow chart of weekly report

5.Sorting Algorithm

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Example:

First Pass:
(5 1 4 2 8) –> (1 5 4 2 8), Here, algorithm compares the first two elements, and swaps since 5 > 1.
(1 5 4 2 8) –> (1 4 5 2 8), Swap since 5 > 4
(1 4 5 2 8) –> (1 4 2 5 8), Swap since 5 > 2
(1 4 2 5 8) –> (1 4 2 5 8), Now, since these elements are already in order (8 > 5), algorithm does not swap them.
Second Pass:
(1 4 2 5 8) –> (1 4 2 5 8)
(1 4 2 5 8) –> (1 2 4 5 8), Swap since 4 > 2
(1 2 4 5 8) –> (1 2 4 5 8)
(1 2 4 5 8) –> (1 2 4 5 8)
Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one whole pass without any swap to know it is sorted.
Third Pass:
(1 2 4 5 8) –> (1 2 4 5 8)
(1 2 4 5 8) –> (1 2 4 5 8)
(1 2 4 5 8) –> (1 2 4 5 8)
(1 2 4 5 8) –> (1 2 4 5 8)
(www.geeksforgeeks.org, 2020)

6.Reflection

Developing the project in Microsoft Visual Studios 2019 keeping C# as primary programming language is new experience for me. Developing a record keeping system for student 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. With the growing of technology, the visual studio and its community helps new developer like us to pace our development speed.

7.Conclusion

The beginning coursework for the module CS6004NA Application Development was to construct up a student detail recording system. It required a long time to construct up the assignment in Visual Studio Venture 2019 utilizing C# programming. The system has login screen to add security to the task. After login, the system appears an essential screen where each one of the functionalities is found. Aside from different shape components, course diagram for each one of the structures and classes were utilized. I would like to thank my supervisor Mr. Ishwor Sapkota for guiding me throughout the coursework.

8.Refrences

www.geeksforgeeks.org, 2020. *www.geeksforgeeks.org.* [Online] Available at: https://www.geeksforgeeks.org/bubble-sort/

www.researchgate.net, 2020. *www.researchgate.net.* [Online] Available at: https://www.researchgate.net/publication/266243197_The_Design_and_Implementation_of_Student_Academic_Record_Management_System

9.Appendix

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

namespace CourseWorkAD
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
            LoadPieChartData();

        }
        private void LoadPieChartData()
        {
            int sum_computing = 1;
            int sum_mediatechnology = 2;
            int sum_networksanditsecurity = 2;


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

            for (int i = 0; i < dtable.Rows.Count; i++)
            {
                String lol = dtable.Rows[i]["CourseEnroll"].ToString();
                if (lol == "Computing")
                {
                    sum_computing++;
                }
                else if (lol == "Multimedia Technology")
                {
                    sum_mediatechnology++;
                }
                else if (lol == "Networks and IT Security")
                {
                    sum_networksanditsecurity++;
                }

            }

            dtable.Rows.Add("Computing", sum_computing);
            dtable.Rows.Add("Multimedia Technology", sum_mediatechnology);
            dtable.Rows.Add("Networks and IT Security",
sum_networksanditsecurity);


((System.Windows.Controls.DataVisualization.Charting.PieSeries)Pie).ItemsSource
=
                new KeyValuePair<string, int>[]{
        new KeyValuePair<string,int>("Computing", sum_computing),
        new KeyValuePair<string,int>("Multimedia Technology",
sum_mediatechnology),
        new KeyValuePair<string,int>("Networks and IT Security",
sum_networksanditsecurity) };

        }
    }
}




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;

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

        public StudentForm StudentForm
        {
            get => default;
            set
            {
            }
        }

        private void TextBox_TextChanged(object sender, TextChangedEventArgs e)
        {

        }

        private void Login_Click(object sender, RoutedEventArgs e)
        {
            if(Password.Password!=""&& Useraname.Text!="")
            {
                if(Password.Password=="admin"&& Useraname.Text== "admin")
                    {
                    MessageBox.Show("Login Sucess!");
                    StudentForm studentform = new StudentForm();
                    studentform.ShowDialog();
                }
            }
        }
    }
}


using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Data;
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;
using DataHandler;
using Microsoft.Win32;
using System.Data.OleDb;
using System.Globalization;

namespace CourseWorkAD
{
    /// <summary>
    /// Interaction logic for StudentForm.xaml
    /// </summary>
    public partial class StudentForm : Window
    {
        DataTable dataTable;
        public StudentForm()
        {
            InitializeComponent();
        }

        public Window1 Window1
        {
            get => default;
            set
            {
            }
```

```
}

public class Student
{
    public int ID { get; set; }
    public string Name { get; set; }
    public string Address { get; set; }
    public string phoneNumber { get; set; }
    public string Courseenroll { get; set; }
    public string RegDate { get; set; }
}

private void buttonclick(object sender, RoutedEventArgs e)
{
    var handler = new Handler();
    var dataSet = new DataSet();

    if (File.Exists(@"D:\student.xml"))
    {
        dataSet.ReadXml(@"D:\student.xml");
    }
    else
    {

        dataSet = handler.CreateDataSet();


    }
    AddSampleData(dataSet);
    dataSet.WriteXml(@"D:\student.xml");
    LoadGrid();
}

void LoadGrid()
{
    if (File.Exists(@"D:\student.xml"))
    {
        var dataSet = new DataSet();
        dataSet.ReadXml(@"D:\student.xml");
        dataTable = dataSet.Tables["Student"];
        datagrid.ItemsSource = dataTable.DefaultView;
    }
}

private void AddSampleData(DataSet dataSet)
{
    var dr1 = dataSet.Tables["Student"].NewRow();
    dr1["ID"] = studentid.Text;
    dr1["Name"] = studentname.Text;
    dr1["Address"] = studentaddress.Text;
    dr1["ContactNo"] = studentphone.Text;
    dr1["CourseEnroll"] = studentcourse.Text;
    dr1["RegistrationDate"] = studentdate.Text;
    dataSet.Tables["Student"].Rows.Add(dr1);
}



private void sortname_Click(object sender, RoutedEventArgs e)
{
    if (File.Exists(@"D:\student.xml"))
    {
```

```
                dataTable.DefaultView.Sort = "Name ASC";
                // dataSet.ReadXml(@"D:\student.xml");
                datagrid.ItemsSource = dataTable.DefaultView;
            }
        }

        private void sortregd_Click(object sender, RoutedEventArgs e)
        {

               if (File.Exists(@"D:\student.xml"))
            {
                var dataSet = new DataSet();
                dataTable.DefaultView.Sort = "Name ASC";
                dataSet.ReadXml(@"D:\student.xml");
                datagrid.ItemsSource = dataSet.Tables["Student"].DefaultView;
            }

        }

        private void btnweeklyreport_Click(object sender, RoutedEventArgs e)
        {
            int sum_computing = 1;
            int sum_mediatechnology = 2;
            int sum_networksanditsecurity = 2;


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

            for (int i = 0; i < dtable.Rows.Count; i++)
            {
                String lol = dtable.Rows[i]["CourseEnroll"].ToString();
                if (lol == "Computing")
                {
                    sum_computing++;
                }
                else if (lol == "Multimedia Technology")
                {
                    sum_mediatechnology++;
                }
                else if (lol == "Networks and IT Security")
                {
                    sum_networksanditsecurity++;
                }

            }

            dtable.Rows.Add("Computing", sum_computing);
            dtable.Rows.Add("Multimedia Technology", sum_mediatechnology);
            dtable.Rows.Add("Networks and IT Security",
sum_networksanditsecurity);


            datagrid.ItemsSource = dtable.DefaultView;
        }

        private void datagrid_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {

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

            string pathOnly = System.IO.Path.GetDirectoryName(path);
            string fileName = System.IO.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 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);
                datagrid.DataContext = tableStd.DefaultView;
            }



            //var dataSet = new DataSet();
            //dataSet.ReadXml(@"D:\student.xml");
            //OpenFileDialog openFileDialog = new OpenFileDialog();
            //if (openFileDialog.ShowDialog() == true)
            //{
            //    string filePath = openFileDialog.FileName;
            //    //read all std from file code copy

            //    using (var reader = new StreamReader(filePath))
            //    {
            //        reader.ReadLine();
            //        while (!reader.EndOfStream)
            //        {
            //            var line = reader.ReadLine();
            //            var values = line.Split(',');
            //            var newRow = dataSet.Tables["Student"].NewRow();
            //            newRow["ID"] = values[0];
            //            newRow["Name"] = values[1];
            //            newRow["Address"] = values[2];
            //            newRow["ContactNo"] = values[3];
            //            newRow["CourseEnrol"] = values[4];
            //            newRow["RegistrationDate"] = values[5];
            //            dataSet.Tables["Student"].Rows.Add(newRow);

            //            dataSet.WriteXml(@"D:\student.xml");
            //        }
            //     }

        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            Window1 window1 = new Window1();
            window1.Show();


        }
    }
    }
```

```
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());

            //ForeignKeyConstraint          courseWorkFK          =          new
ForeignKeyConstraint("courseWorkFK",
            //ds.Tables["Course"].Columns["ID"],
          //  ds.Tables["Student"].Columns["CourseEnroll"]);
           // courseWorkFK.DeleteRule = Rule.None;
            //ds.Tables["Student"].Constraints.Add(courseWorkFK);
            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("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(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"] };
            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.Columns.Add("CourseDuration", 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("Reg No", 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(DateTime));


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

```
<Window x:Class="CourseWorkAD.StudentForm"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkAD"
        mc:Ignorable="d"
        Title="StudentForm" Height="547.709" Width="656">
    <Grid Margin="0,0,45,59">
        <Label Content="Student Details" HorizontalAlignment="Left"
Margin="145,10,0,0" VerticalAlignment="Top" Width="122"/>
        <Label Content="ID" HorizontalAlignment="Left" Margin="60,59,0,0"
VerticalAlignment="Top"/>
        <Label Content="Name" HorizontalAlignment="Left" Margin="49,90,0,0"
VerticalAlignment="Top" RenderTransformOrigin="-0.228,1.441"/>
        <Label Content="Address" HorizontalAlignment="Left" Margin="49,140,0,0"
VerticalAlignment="Top"/>
        <Label Content="phoneNumber" HorizontalAlignment="Left"
Margin="49,197,0,0" VerticalAlignment="Top" />
        <Label Content="Course enroll" HorizontalAlignment="Left"
Margin="49,248,0,0" VerticalAlignment="Top" Width="80"/>
        <Label Content="Registration Date" HorizontalAlignment="Left"
Margin="49,295,0,0" VerticalAlignment="Top"/>
        <TextBox x:Name="studentaddress" HorizontalAlignment="Left" Height="23"
Margin="180,140,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>
        <TextBox x:Name="studentphone" HorizontalAlignment="Left" Height="23"
Margin="180,197,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>
        <TextBox x:Name="studentname" HorizontalAlignment="Left" Height="23"
Margin="180,94,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>
        <Button x:Name="save" Content="Save" HorizontalAlignment="Left"
Margin="488,177,0,0" VerticalAlignment="Top" Width="102" Height="29"
Click="buttonclick" RenderTransformOrigin="0.516,-0.898"/>
        <ComboBox x:Name="studentcourse" HorizontalAlignment="Left"
Margin="180,248,0,0" VerticalAlignment="Top" Width="120">
            <ComboBoxItem Content="Network and it Security"/>
            <ComboBoxItem Content="Multimedia Technologies"/>
            <ComboBoxItem Content="Computing"/>
        </ComboBox>
        <DatePicker x:Name="studentdate" HorizontalAlignment="Left"
Margin="180,296,0,0" VerticalAlignment="Top" Width="133" Height="25"/>
        <Button x:Name="import" Content="Import from CSV"
HorizontalAlignment="Left" Margin="488,59,0,0" VerticalAlignment="Top"
Width="102" Height="39" Click="import_Click"/>
        <DataGrid x:Name ="datagrid" HorizontalAlignment="Left" Height="124"
Margin="10,365,0,-29.333" VerticalAlignment="Top" Width="437"
ItemsSource="{Binding}" SelectionChanged="datagrid_SelectionChanged">

        </DataGrid>
        <Button x:Name="sortname" Content="SortByName"
HorizontalAlignment="Left" Margin="488,116,0,0" VerticalAlignment="Top"
Width="102" Height="38" Click="sortname_Click"/>
        <Button x:Name="sortregd" Content="SortByRegistrationDate"
HorizontalAlignment="Left" Margin="453,339,0,0" VerticalAlignment="Top"
Width="151" Height="43" Click="sortregd_Click"/>
        <Button Name="btnweeklyreport" Content="weeklyReport"
HorizontalAlignment="Left" Margin="475,226,0,0" VerticalAlignment="Top"
Width="129" Height="33" Click="btnweeklyreport_Click"/>
        <Button x:Name="displaychart" Content="Display Chart"
HorizontalAlignment="Left" Margin="475,285,0,0" VerticalAlignment="Top"
Width="125" Height="36" Click="Button_Click"/>
        <TextBox x:Name="studentid" HorizontalAlignment="Left" Height="26"
Margin="180,59,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>
```

```
    </Grid>
</Window>


<Window x:Class="CourseWorkAD.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkAD"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Grid>
        <Label Content="Login Form" HorizontalAlignment="Left"
Margin="277,10,0,0" VerticalAlignment="Top" Width="125" Height="39"/>
        <Label Content="UserName" HorizontalAlignment="Left" Margin="78,100,0,0"
VerticalAlignment="Top" Width="97"/>
        <Label Content="Password" HorizontalAlignment="Left" Margin="78,168,0,0"
VerticalAlignment="Top" Width="89"/>
        <Button x:Name="Login" Content="Login" HorizontalAlignment="Left"
Margin="186,268,0,0" VerticalAlignment="Top" Width="162"
RenderTransformOrigin="0.323,0.645" Click="Login_Click" Height="58"/>
        <TextBox x:Name="Useraname" HorizontalAlignment="Left" Height="42"
Margin="169,84,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="198"
TextChanged="TextBox_TextChanged"/>
        <PasswordBox x:Name="Password" HorizontalAlignment="Left"
Margin="166,165,0,0" VerticalAlignment="Top" Height="44"
RenderTransformOrigin="0.5,0.5" Width="205">
            <PasswordBox.RenderTransform>
                <TransformGroup>
                    <ScaleTransform/>
                    <SkewTransform/>
                    <RotateTransform Angle="0.503"/>
                    <TranslateTransform/>
                </TransformGroup>
            </PasswordBox.RenderTransform>
        </PasswordBox>

    </Grid>
</Window>
```