Marking Scheme

Informatics College Pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: **Satkar Acharya**
London Met ID: Enter ID Here

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	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	very poorly explained.
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	very poorly developed application

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

Informatics College Pokhara

Application development

CS6004

Coursework 1

Submitted By:		**Submitted To:**
Student Name:	Satkar Acharya	Mr. Ishwor Sapkota
Student ID:	NP04CP6A170056	Module Leader
Group:	L3C2	Application Development
Date:	9-Jan-2020	

List of Content

List of Figures

Introduction

The information system is important in collecting all data and information of all staff or member in one organization to be in one place. The system is normally provided very helpful task that will replace the human as to keep it in file as the inventory or other purposes. In order to design a helpful system in order to make ease to the user.

The system designed is about Student Information System. This system is designed and developed in C#. The functions and features which Student Information System required are fulfilled by this system. Features like auto registration ID is also added in the developed system and can view weekly report and chart. Detail like name, address, course enrol, registration date can be added. But at first login is required, without login the system cannot run. Additional features are also explained in report. (mastersoft, 2018)

Current Scenario

There is various system which do not contain various features and are old traditional system. There are some school collages universities which are still using old system and are facing various problems regarding to student information and not secured as well. The disadvantage of the existing system is not reliable for the user. In the case when a file is opened for each application, the previous system showed that there some problems occurred e.g. data loss and damage, the list of files is difficult to be viewed and difficult to be checked the student status. (S.R.Bharamagoudar, june 2013)

Proposed System

The project targeted several objectives that to be achieved at the end of this project. The objective is to improvement data security issue. The purposed system is specially designed to overcome the problems mentioned above. The system is more secure and without login student detail section cannot be used. The purposed system is more user-friendly, easy to understand and use.

User Manual

The screenshot below will illustrate user how to operate the system.

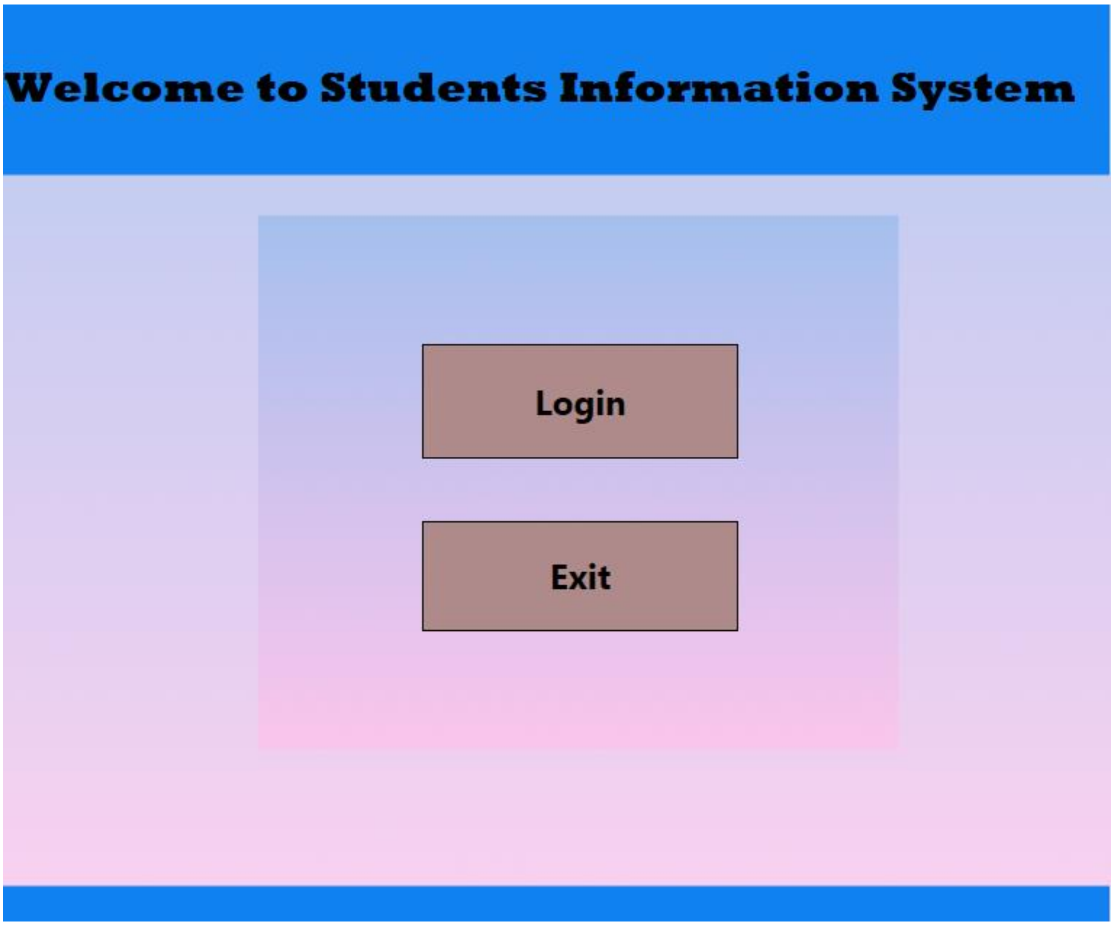


Figure 1: Homepage of Student Information System

This is the main page of the Student Information System. By clicking on Login we can enter to login page. Otherwise we can exit.

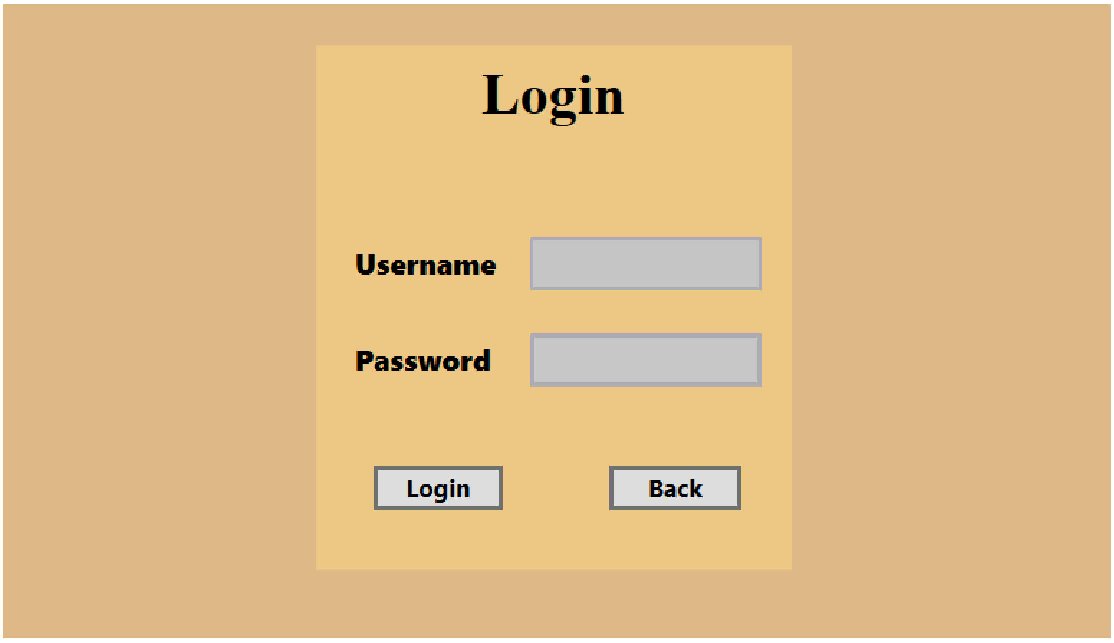
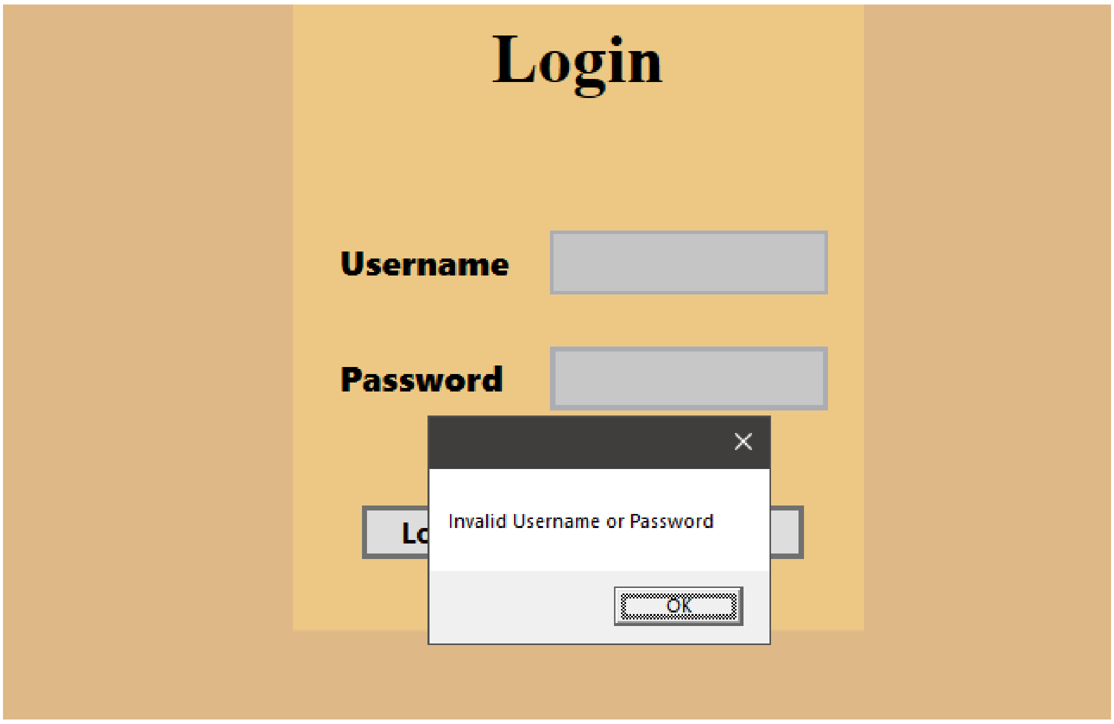


Figure 2: Login page

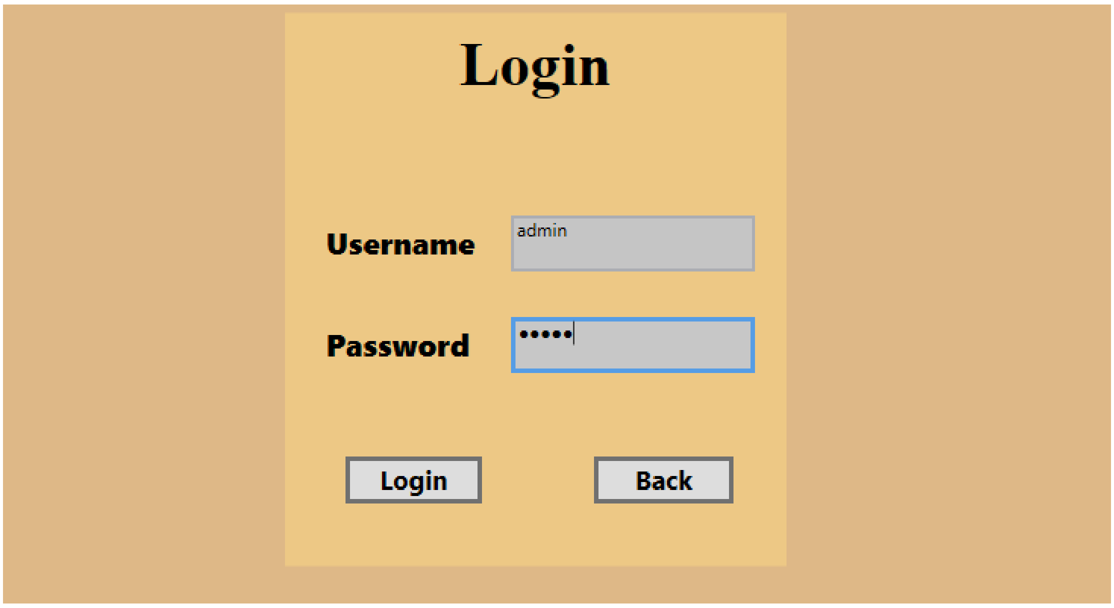
This is the login page of the System where we need to enter valid username and password.



Invalid Username or Password

Figure 3: Login Popup for wrong password

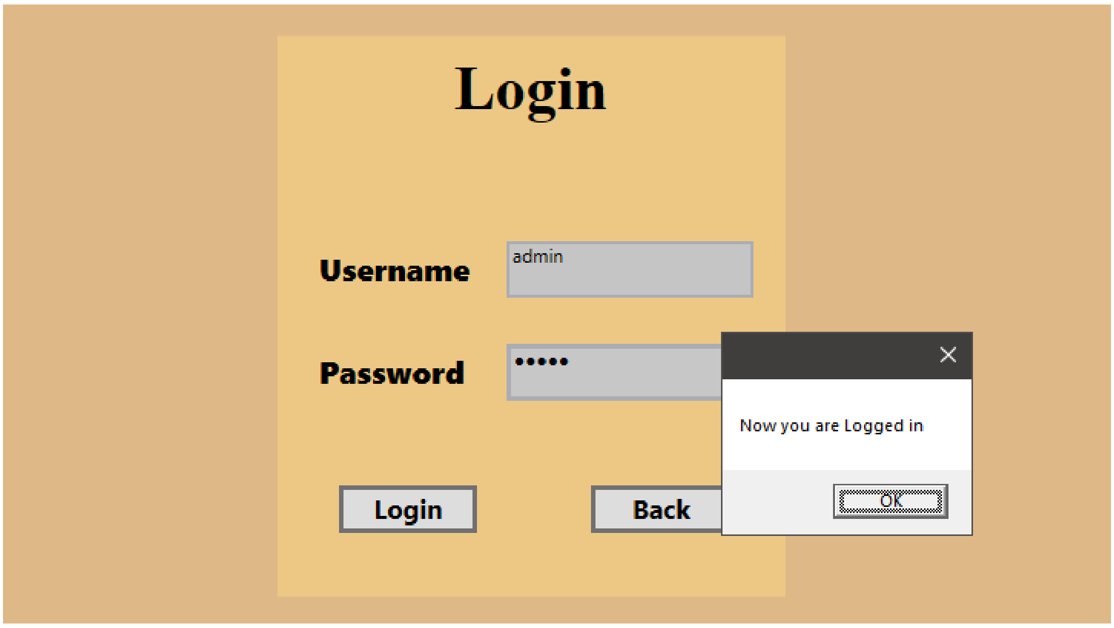
If we enter wrong username or password in the login page the popup message as show in the above figure.



Username admin

Password •••••

Figure 4: Login default Username and Password



Login

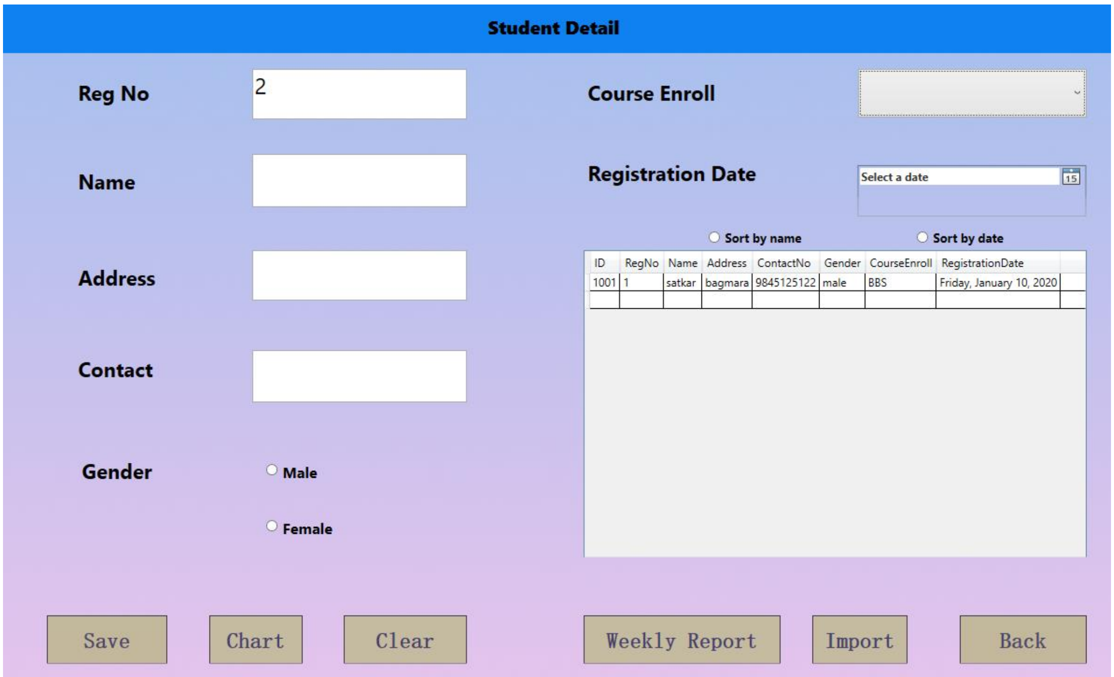
Username: admin

Password: •••••

Now you are Logged in

Figure 5: Login Popup for valid Username and Password

If we enter valid username and password, the above popup will appear and we can enter to main page of the form.



Student Detail

Reg No: 2

Name:

Address:

Contact:

Gender: ○ Male ○ Female

Course Enroll:

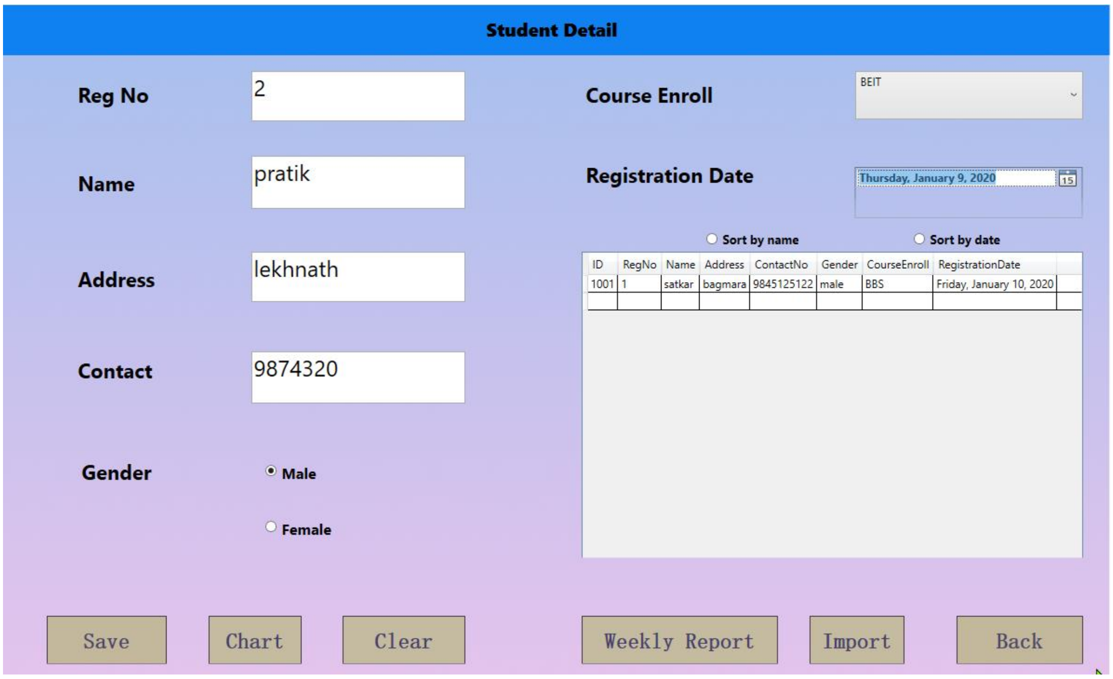
Registration Date: Select a date

○ Sort by name ○ Sort by date

ID	RegNo	Name	Address	ContactNo	Gender	CourseEnroll	RegistrationDate
1001	1	satkar	bagmara	9845125122	male	BBS	Friday, January 10, 2020

Figure 6: Main Form of Student Information System

This is the main form of the student Information System where we can enter student detail.



Reg No: 2

Name: pratik

Address: lekhnath

Contact: 9874320

Gender: ● Male ○ Female

Course Enroll: BEIT

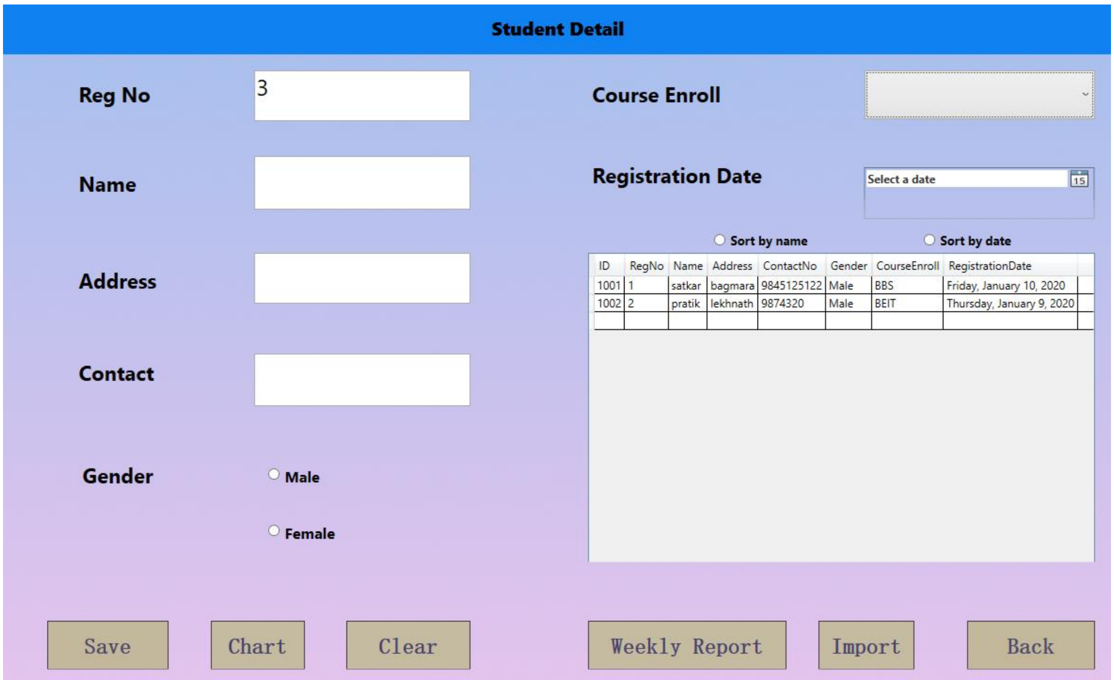
Registration Date: Thursday, January 9, 2020

○ Sort by name ○ Sort by date

ID	RegNo	Name	Address	ContactNo	Gender	CourseEnroll	RegistrationDate
1001	1	satkar	bagmara	9845125122	male	BBS	Friday, January 10, 2020

Figure 7: Entering Data of Student

Above figure shows how the user can enter data on the form.



Reg No: 3

Name:

Address:

Contact:

Gender: ○ Male ○ Female

Course Enroll:

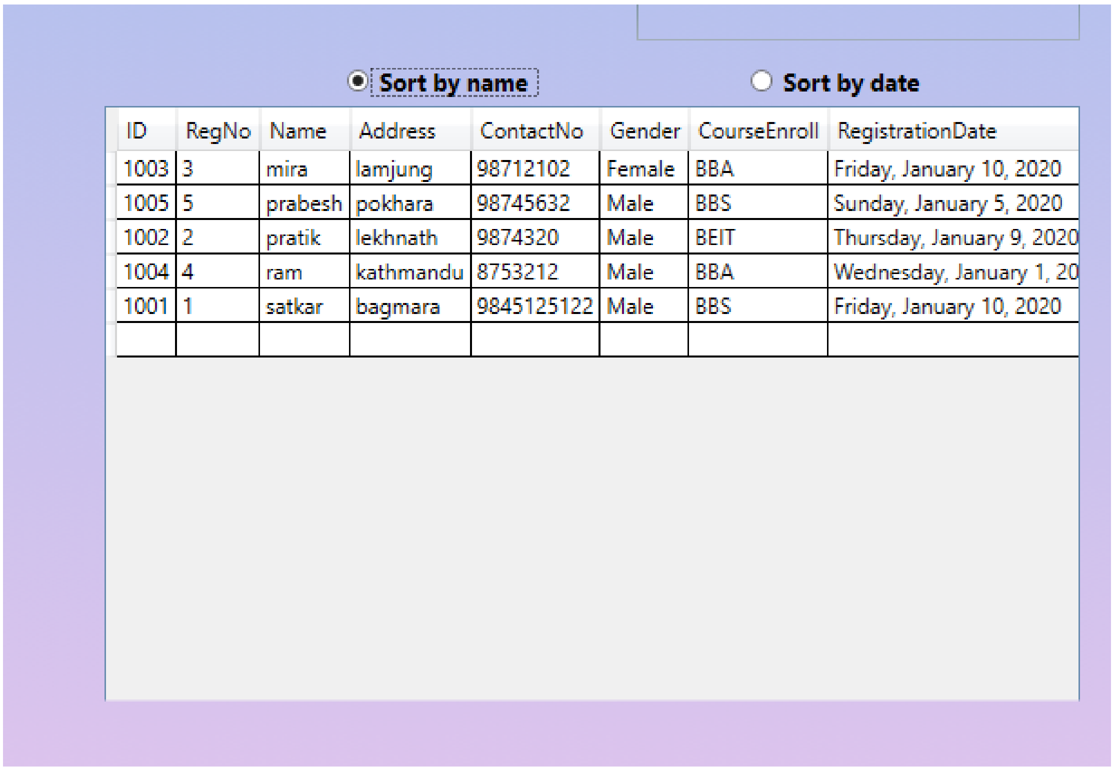
Registration Date: Select a date

○ Sort by name ○ Sort by date

ID	RegNo	Name	Address	ContactNo	Gender	CourseEnroll	RegistrationDate
1001	1	satkar	bagmara	9845125122	Male	BBS	Friday, January 10, 2020
1002	2	pratik	lekhnath	9874320	Male	BEIT	Thursday, January 9, 2020

Figure 8: Entered Data on Table

Recently entered data is shown in the table as shown in the figure.

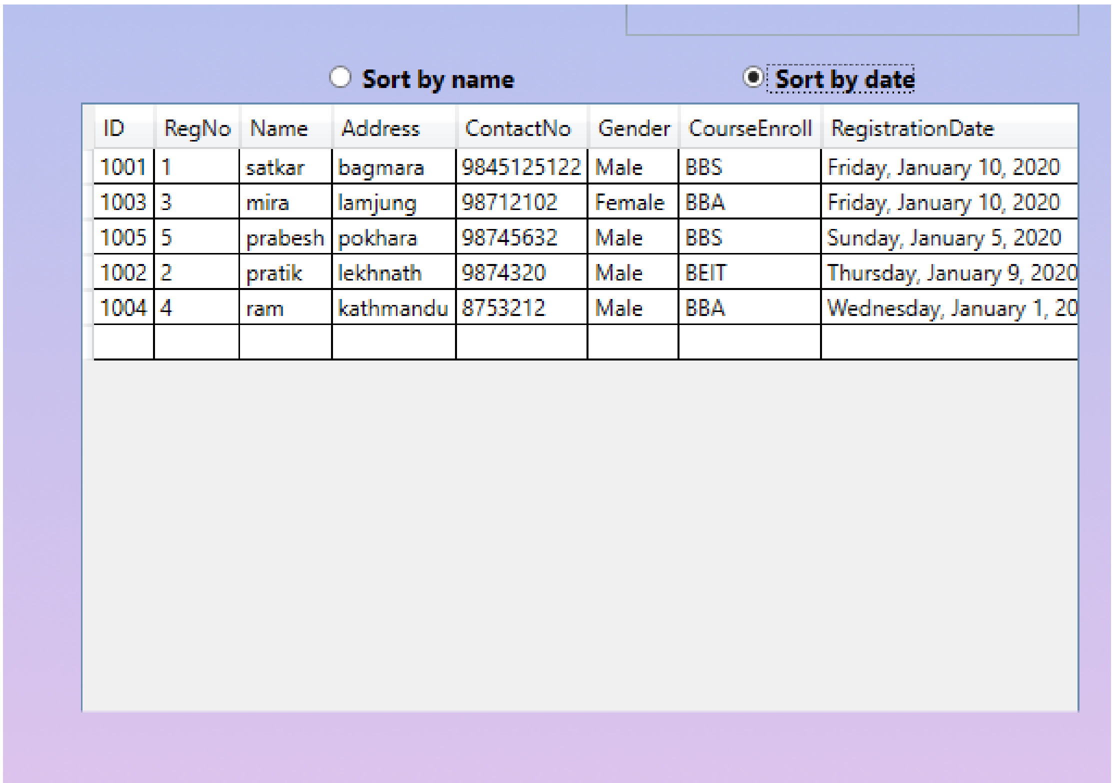


● Sort by name ○ Sort by date

ID	RegNo	Name	Address	ContactNo	Gender	CourseEnroll	RegistrationDate
1003	3	mira	lamjung	98712102	Female	BBA	Friday, January 10, 2020
1005	5	prabesh	pokhara	98745632	Male	BBS	Sunday, January 5, 2020
1002	2	pratik	lekhnath	9874320	Male	BEIT	Thursday, January 9, 2020
1004	4	ram	kathmandu	8753212	Male	BBA	Wednesday, January 1, 20
1001	1	satkar	bagmara	9845125122	Male	BBS	Friday, January 10, 2020

Figure 9: Sorting of data by name in table

Above figure shows sorting by name which makes user easy to find the detail they search for.

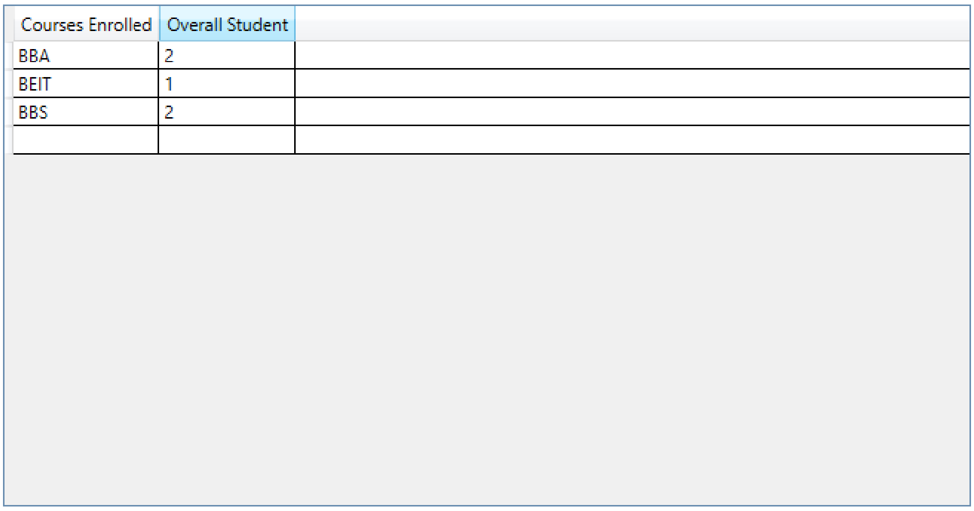


○ Sort by name ● Sort by date

ID	RegNo	Name	Address	ContactNo	Gender	CourseEnroll	RegistrationDate
1001	1	satkar	bagmara	9845125122	Male	BBS	Friday, January 10, 2020
1003	3	mira	lamjung	98712102	Female	BBA	Friday, January 10, 2020
1005	5	prabesh	pokhara	98745632	Male	BBS	Sunday, January 5, 2020
1002	2	pratik	lekhnath	9874320	Male	BEIT	Thursday, January 9, 2020
1004	4	ram	kathmandu	8753212	Male	BBA	Wednesday, January 1, 20

Figure 10: Sorting of data by Date in table

The above figure shows sorting of data by date so that user can search data according to date of registration.



Courses Enrolled	Overall Student
BBA	2
BEIT	1
BBS	2

Figure 11: Weekly Report of Students

The above figure shows overall students according to course enrolled.

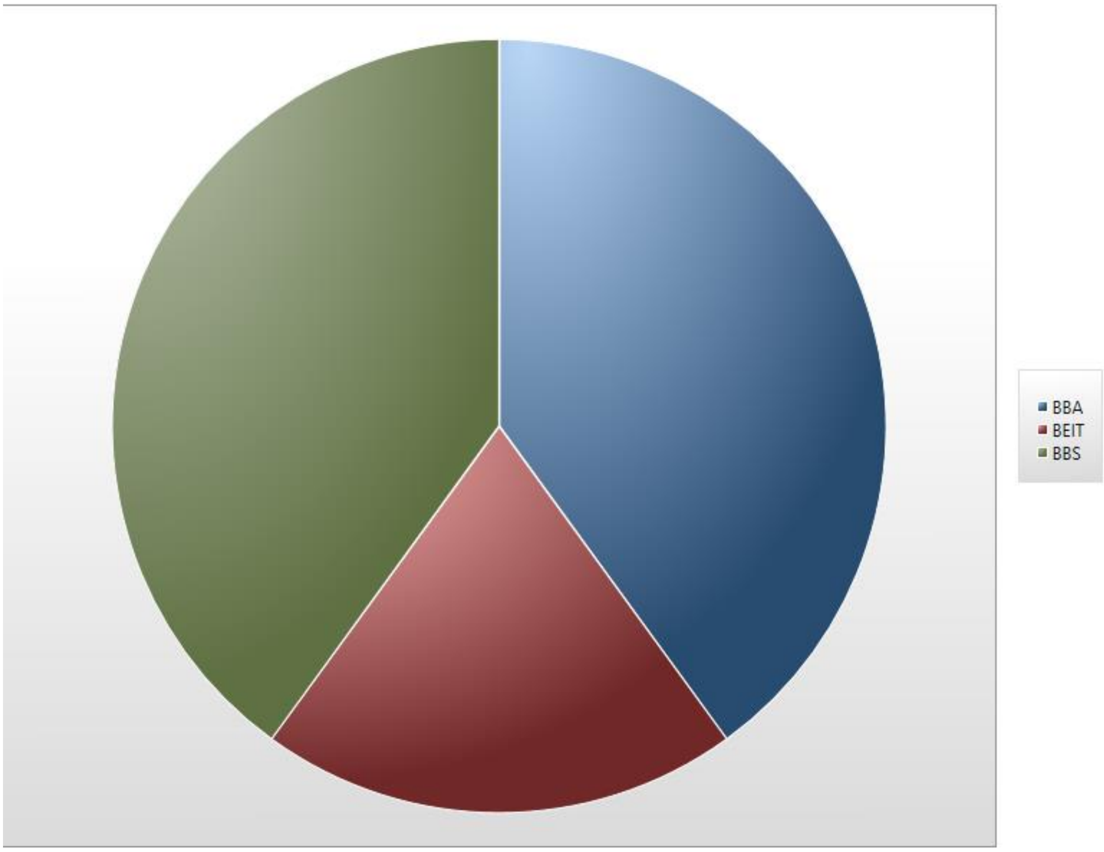


Figure 12: Showing Data in Pie chart

The above figure shows all the entered data in pie chart.

Journal Articles

I. Previously, the college relied heavily on paper records for this initiative. While paper records are a traditional way of managing student data there are several drawbacks to this method. First, to convey information to the students it should be displayed on the notice board and the student has to visit the notice board to check that information. It takes a very long time to convey the information to the student. Paper records are difficult to manage and track. (S.R.Bharamagoudar, june 2013)

II. As an example of SIMS (student information management system) developed independently by School of Information Science & Engineering of Shandong Normal University, this paper introduces database design, specific realization of each function module and key technologies used in the system. In the .NET environment, using ASP.NET technology, Visual C# and JavaScript as programming language, this system accesses the database of Microsoft SQL Server 2005 with ADO.NET technology, and could be employed by users with high security following the access control mechanism of RBAC (role-based access control) on Web services. (Yu-FangTang & Zhang, 2009)

III. In this paper, Student Information Supervision System gives path for maintaining of student information & also provides the guideline for students in different areas. It can be used by educational institute or colleges to continue the records of students without problems. New Teaching learning process is also required in current market scenario so as to aware about the students regarding the recent advances in teaching and educational field. The formation and organization of exact, up-to-date information concerning a students' educational profession is extensively important in the university as well as colleges. Student information system deals with all kind of student details, learning related information, college details, course particulars, set of courses, group details, placement details and other resource related details too. (Ketaki S. Kadam, 2020)

IV. Student Information System is one of the key systems for facilitating the management and development of Higher Education Institutions. Its use for academic decision-making purposes as well as other academic tasks is crucial. Therefore, this paper aims to understand the impact of System Quality, Information Quality and Information Presentation on Student Information System satisfaction of academic and administrative staff. In this study, System satisfaction survey is carried out and factor analysis and regression tests are applied to interpret the collected data. The results show that only Information Quality has direct effect on satisfaction. Then the impact of decision-making as a mediator factor on system satisfaction is measured and the results reveal that System Quality and Information Quality has indirect significant effect whereas Information Presentation does not have direct nor indirect effect on system satisfaction. (Cannur Gurkut, 2018)

System Architecture

Architecture Diagram

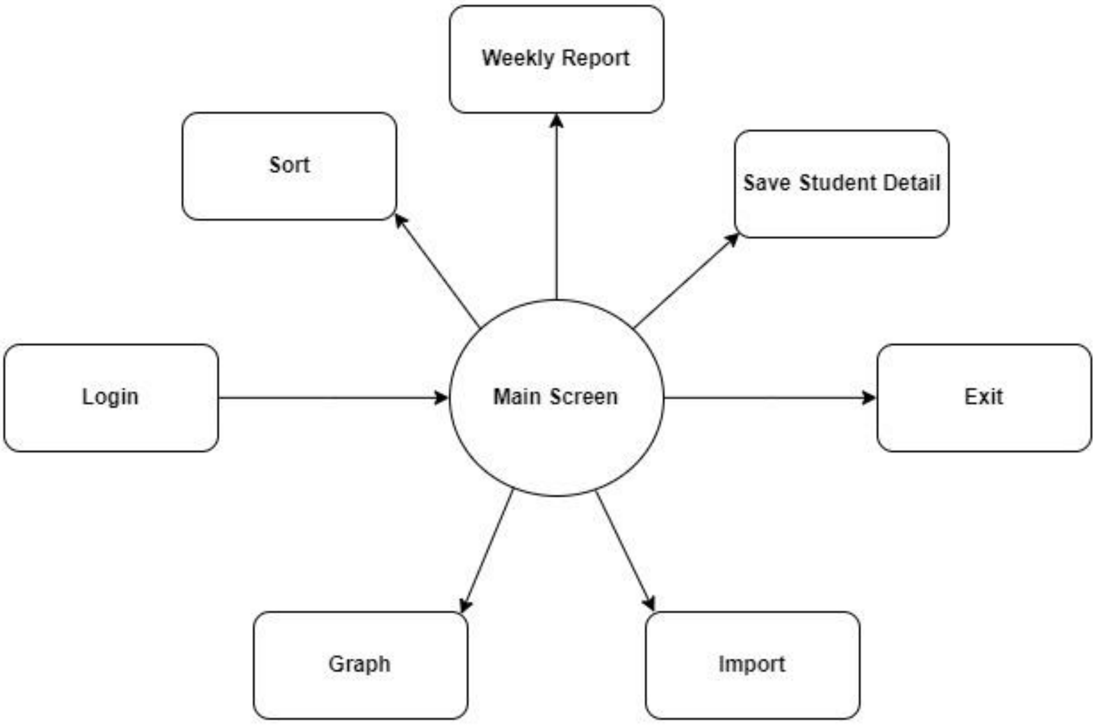


Figure 13: Architecture Diagram

Flow Chart for Report

Weekly Report

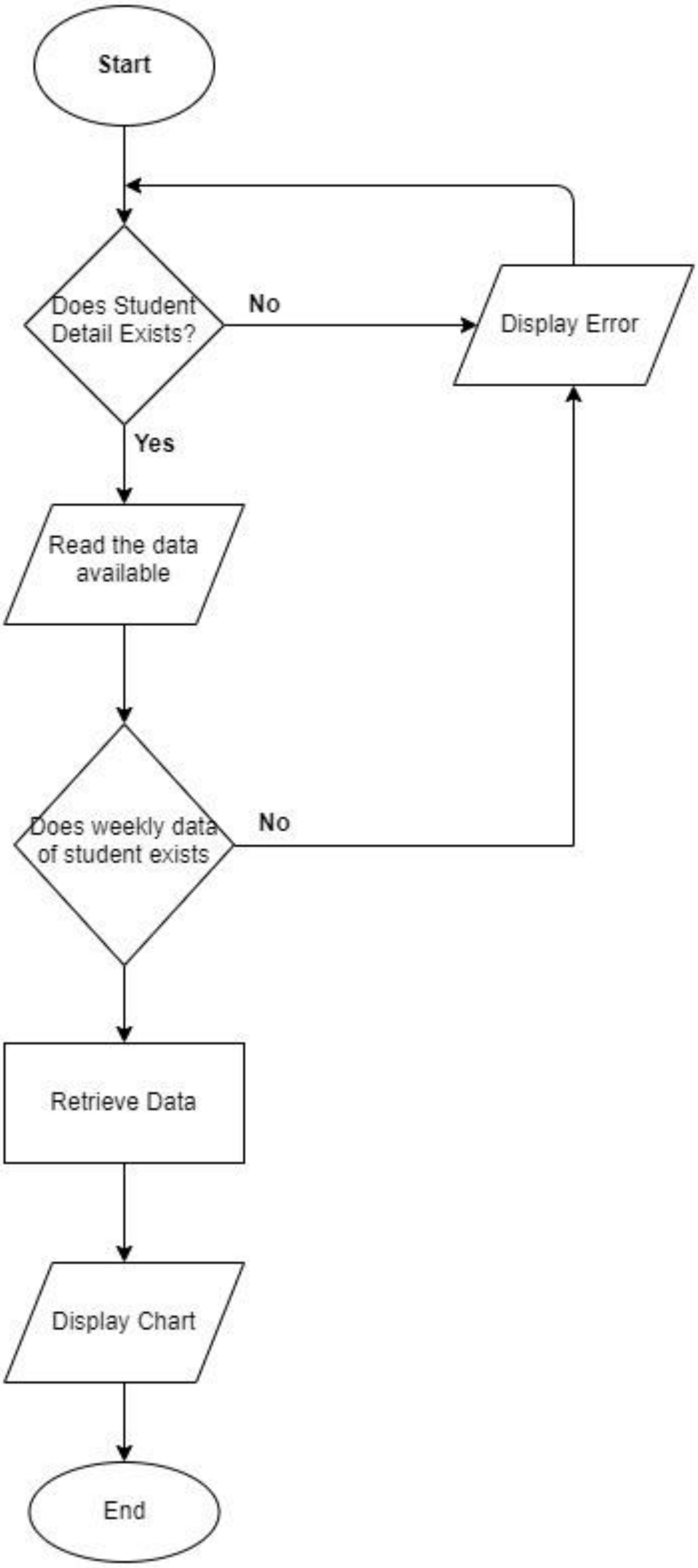


Figure 14: Flow Chart of Weekly Report

Algorithms of Reports

Weekly Report

Steps:

1. Start
2. Check whether the Student Detail Exists or not
3. If Student Data does not exist, then Display Error and Restart
4. If Exist, then read the Student data available
5. Check whether the student weekly data exist or not
6. If Student Weekly Data does not exist, then Display Error and restart
7. If Student Weekly Data exist, then Retrieve Data
8. Display the student data in Chart
9. End

Reflection

The system developed is about Student Information System. The system is developed using Visual Studio 2019 with C# language. GUI designed is user-friendly so that anyone can understand and easy to use. The logic used in the developed system reflects the working environment of the Universities, Collages.

The developed system has facilities of registration date of student along with their name. The registration Id is automatically given by the system. Details like Name, Address, Contact, Gender can be added manually. As well as the User can check weekly chart according to course enrolled by number of students. The system also supports sorting data on the basis of name and date which makes easy to locate data. And the project has login page which is used as security. Without username and password, the system cannot be continuing. And the default username and password is admin and admin.

Due having some previous experience with Visual Studio makes easy in doing this project. This project was not easy but having some experience with visual studio I got plus point in doing this project. Additional features like sorting validation, chart and import from CSV are completely new for me.

Conclusion

As the given coursework is now completed which is about Student Information System which is to developed in visual studio using C# programming language. The system has login page which is used as security. Without login the system cannot be used. And the username and password of the login page is set to default as admin and admin. After login the main page is loaded where student detail can be added.

The given task was not easy for me due being completely new to c# programming language. To complete the task, I had to go through many problems and with the help of friends, serving on internet, and with the help of supervisor the task was completed. The task was completed with hard effort. I learn many more things during this project. To gain the knowledge about the C# is very useful because it is becoming very important in today’s world. So to be familiar with such topic is very useful for us. This coursework has made my confidence to increase for the best performance in my upcoming coursework as it has lift up my development skills. And lastly, I’d like to thank our module teacher as he has helped us the most for every problem faced.

Bibliography

Cannur Gurkut, M. N. (2018). Important Factors Affecting Student Information System Quality and Satisfaction. *Eurasua journal of mathematics, science and technology education*, 14.

Ketaki S. Kadam, O. V. (2020). A Review paper on Student Information Supervision System. *International Journal of Research In Science & Engineering*.

mastersoft. (2018). Retrieved from https://www.iitms.co.in/products/student-information-system-sis/

S.R.Bharamagoudar. (june 2013). Student Information Management. *International Journal of Advanced Research in Computer and Communication Engineering*.

Yu-FangTang, & Zhang, Y.-S. (2009). *Design and implementation of college Student Information Management System2.*

Appendix

Main Window.

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

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



        }

        public void Startup()
        {
            main.Content = new HomePage();
        }

    }
}
```

Login Page.

```
using DataHandler;
using Microsoft.Win32;
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 CW1Appdevelopment
{
    /// <summary>
    /// Interaction logic for Studentdetails.xaml
    /// </summary>
    public partial class Studentdetails : Page
    {
        public Studentdetails()
        {
            InitializeComponent();
            txtRegNo.Text = read_from_file();
            LoadStudentData();
        }
        List<string> courseList = new List<string>();
        private string fileName;
        public void Startup()
        {
            //var handler = new Handler();
            //var dataSet = handler.CreateDataSet();
            //AddSampleData(dataSet);
            //dataSet.WriteXmlSchema(@"D:\StudentCWSchema.xml");
            //dataSet.WriteXml(@"D:\StudentCWData.xml");

            //var dataSet = new DataSet();
            //dataSet.ReadXmlSchema(@"D:\StudentCWSchema.xml");
            //dataSet.ReadXml(@"D:\StudentCWData.xml");

            //var i = 0;
        }


        private void AddSampleDataforStd(DataSet dataSet)
        {

            var dr = dataSet.Tables["Course"].NewRow();

            dataSet.Tables["Course"].Rows.Add(dr);

            var dr1 = dataSet.Tables["Student"].NewRow();
            dr1["Name"] = txtName.Text;
            dr1["Address"] = txtAddress.Text;
            dr1["ContactNo"] = txtContact.Text;
            dr1["CourseEnroll"] = courseEnroll.Text;
            dr1["RegistrationDate"] = datePicker.Text;
            dataSet.Tables["Student"].Rows.Add(dr1);


        }

        private void AppendStdReport(DataSet dataSet)
        {

            var handler = new Handler();

dataSet.Tables["StudentReport"].ReadXml(@"files\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"] = courseEnroll.Text;
            dr2["RegistrationDate"] = datePicker.Text;
            dataSet.Tables["StudentReport"].Rows.Add(dr2);


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


        }
        private void Save(object sender, RoutedEventArgs e)
        {

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

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

            //dataSet.Tables[2].WriteXml(@"files\StudentReport.xml");

            write_to_file(txtRegNo.Text);
```

```
    txtRegNo.Text = read_from_file();

    ClearControls();
    LoadStudentData();
}

public void Clear(object sender, EventArgs e)
{

    txtName.Clear();
    txtAddress.Clear();
    txtContact.Clear();



}

private void write_to_file(string text)
{

    System.IO.File.WriteAllText(@"files\count.txt", text);

}
private string read_from_file()
{
    string text = System.IO.File.ReadAllText(@"files\count.txt");

    int i;

    i = int.Parse(text.ToString());
    i = i + 1;
```

```
        return i.ToString();

    }

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

    private void LoadStudentData()
    {
        if (System.IO.File.Exists(@"files\StudentReport.xml"))
        {
            var handler = new Handler();

            var dataSet = new DataSet();



            dataSet.ReadXml(@"files\StudentReport.xml");

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

        }
    }
```

```
}




private void Back(object sender, RoutedEventArgs e)
{
   this.NavigationService.Navigate(new Homepage());
}



private void Weeklyreport_Click(object sender, RoutedEventArgs e)
{
   this.NavigationService.Navigate(new Weeklyreport());
}

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



      var dataSet = new DataSet();


      dataSet.ReadXml(@"files\StudentReport.xml");


      DataTable dtStdReport = new DataTable();
      dtStdReport = dataSet.Tables[0];
      dtStdReport.DefaultView.Sort = "Name";
      grdStd.DataContext = dtStdReport.DefaultView;
   }
```

```
}

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



      var dataSet = new DataSet();


      dataSet.ReadXml(@"files\StudentReport.xml");


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

private void Chart(object sender, RoutedEventArgs e)
{
   this.NavigationService.Navigate(new Chart());
}




private void Import(object sender, RoutedEventArgs e)
{
   try
   {
```

```
var dataSet = new DataSet();
dataSet.ReadXml((@"files/StudentReport.xml"));
OpenFileDialog openfile = new OpenFileDialog();
openfile.Filter = "CSV Files|*.csv";
openfile.DefaultExt = ".csv";
openfile.FilterIndex = 1;
openfile.Multiselect = false;
bool? fileselect = openfile.ShowDialog();
if (fileselect != null || fileselect == true)
{
    fileName = openfile.FileName;

    using (var reader = new StreamReader(fileName))
    {
        reader.ReadLine();
        while (!reader.EndOfStream)
        {
            var line = reader.ReadLine();
            var values = line.Split(',');
            var dr1 = dataSet.Tables["StudentReport"].NewRow();
            dr1["RegNo"] = values[1];
            dr1["Name"] = values[2];
            dr1["Address"] = values[3];
            dr1["ContactNo"] = values[4];
            dr1["Gender"] = values[5];
            dr1["CourseEnroll"] = values[6];
            dr1["RegistrationDate"] = values[7];

            dataSet.Tables["Student"].Rows.Add(dr1);
            dataSet.WriteXml(@"files/StudentReport.xml");
```

```
                    MessageBox.Show("Fill the field!!");
                 }
              }
              grdStd.ItemsSource = dataSet.Tables["Student"].DefaultView;
              MessageBox.Show("Fill the field!!");
           }
        }
        catch (Exception)
        {
        }
     }
  }
}
```

Student Detail

```
using DataHandler;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Data;
using Microsoft.Win32;
using System.IO;

namespace ADCourseWork
{
    /// <summary>
    /// Interaction logic for StudentDetail.xaml
    /// </summary>
    public partial class StudentDetail : Page
    {
        public StudentDetail()
        {
            InitializeComponent();
            Startup();

            txtRegNo.Text = read_from_file();

            LoadStudentData();
        }
        public string gender = "Male";
        List<string> courseList = new List<string>();
        private string fileName;



        public void Startup()
        {


        }
```

```
private void AddSampleData(DataSet dataSet)
{

    var dr = dataSet.Tables["Course"].NewRow();

    dataSet.Tables["Student"].Rows.Add(dr);

}

private void AddSampleDataforStd(DataSet dataSet)
{

    //var dr = dataSet.Tables["Course"].NewRow();
    //dataSet.Tables["Course"].Rows.Add(dr);

    //var dr1 = dataSet.Tables["Student"].NewRow();
    //dr1["Name"] = txtName.Text;
    //dr1["Address"] = txtAddress.Text;
    //dr1["ContactNo"] = txtContact.Text;

    //dr1["CourseEnroll"] = CourseEnroll.Text;
    //dr1["RegistrationDate"] = datePicker.Text;
    //dataSet.Tables["Student"].Rows.Add(dr1);



}

private void AppendStdReport(DataSet dataSet)
{
```

```
            var handler = new Handler();


dataSet.Tables["StudentReport"].ReadXml(@"file\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["Gender"] = btnMale1.IsChecked;
            dr2["Gender"] = btnFemale1.IsChecked;
            dr2["CourseEnroll"] = CourseEnroll.Text;
            dr2["RegistrationDate"] = datePicker.Text;

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

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




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

            var regno = txtRegNo.Text;
            var name = txtName.Text;
            dataSet.Tables["Student"].WriteXml(@"Indivisual   Student   Data\"   +
name + "CWData" + regno + ".xml");



            write_to_file(txtRegNo.Text);


            txtRegNo.Text = read_from_file();


            ClearControls();
            LoadStudentData();
        }

        private void write_to_file(string text)
        {


            System.IO.File.WriteAllText(@"file\count.txt", text);



        }
        private string read_from_file()
        {

            string text = System.IO.File.ReadAllText(@"file\count.txt");

            int i;

            i = int.Parse(text.ToString());
            i = i + 1;
            return i.ToString();

        }

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

        private void LoadStudentData()
        {

            if (System.IO.File.Exists(@"file\StudentReport.xml"))
            {
                var dataSet = new DataSet();

                dataSet.ReadXml(@"file\StudentReport.xml");

                DataTable dtStdReport = dataSet.Tables[0];
                grdStd.DataContext = dtStdReport.DefaultView;
            }

        }
```

```
private void ClearControls(object sender, RoutedEventArgs e)
{
    txtName.Text = "";
    txtAddress.Text = "";
    txtContact.Text = "";
}

private void back_button(object sender, RoutedEventArgs e)
{
    this.NavigationService.Navigate(new Login());
}

private void btnImport_Click(object sender, RoutedEventArgs e)
{

    this.NavigationService.Navigate(new WeeklyStudentReport());

}

private void grdStd_SelectionChanged(object sender, SelectionChangedEventArgs e)
{

}

private void SortBy_Name(object sender, RoutedEventArgs e)
{
    if (System.IO.File.Exists(@"file\StudentReport.xml"))
    {
        var dataSet = new DataSet();

        dataSet.ReadXml(@"file\StudentReport.xml");

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

private void SortBy_Date(object sender, RoutedEventArgs e)
{

    if (System.IO.File.Exists(@"file\StudentReport.xml"))
    {


        var dataSet = new DataSet();

        dataSet.ReadXml(@"file\StudentReport.xml");

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

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

```
{
    this.gender = (string)(sender as RadioButton).Content;



}

private void btnFemale(object sender, RoutedEventArgs e)
{
    this.gender = (string)(sender as RadioButton).Content;
}

private void GoToGraph(object sender, RoutedEventArgs e)
{
    this.NavigationService.Navigate(new Chart());
}

private void ImportToCSV(object sender, RoutedEventArgs e)
{
    try
    {
        var dataSet = new DataSet();
        dataSet.ReadXml((@"files/StudentReport.xml"));
        OpenFileDialog openfile = new OpenFileDialog();
        openfile.Filter = "CSV Files|*.csv";
        openfile.DefaultExt = ".csv";
        openfile.FilterIndex = 1;
        openfile.Multiselect = false;
        bool? fileselect = openfile.ShowDialog();
        if (fileselect != null || fileselect == true)
        {
```

```
        fileName = openfile.FileName;

        using (var reader = new StreamReader(fileName))
        {
            reader.ReadLine();
            while (!reader.EndOfStream)
            {
                var line = reader.ReadLine();
                var values = line.Split(',');
                var dr1 = dataSet.Tables["StudentReport"].NewRow();
                dr1["RegNo"] = values[1];
                dr1["Name"] = values[2];
                dr1["Address"] = values[3];
                dr1["ContactNo"] = values[4];
                dr1["Gender"] = values[5];
                dr1["CourseEnroll"] = values[6];
                dr1["RegistrationDate"] = values[7];

                dataSet.Tables["Student"].Rows.Add(dr1);
                dataSet.WriteXml(@"files/StudentReport.xml");
                MessageBox.Show("Fill the mandetory field!!");
            }
        }
        grdStd.ItemsSource = dataSet.Tables["Student"].DefaultView;
        MessageBox.Show("Fill the mandetory field!!");
    }
}
catch (Exception)
{
}
```

```
        }
    }
 }
```

Weekly Student Report

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

namespace ADCourseWork
{
    /// <summary>
    /// Interaction logic for WeeklyStudentReport.xaml
    /// </summary>
    public partial class WeeklyStudentReport : Page
    {
        public WeeklyStudentReport()
```

```
{
    InitializeComponent();
     weeklyStudentList();
}
private void weeklyStudentList()
{
    var dataSet = new DataSet();

    dataSet.ReadXml(@"file\StudentReport.xml");

    DataTable dtStdReport = dataSet.Tables[0];

    int Total_BBA = 0;
    int Total_BEIT = 0;
    int Total_BBS = 0;

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



    for (int i = 0; i < dtStdReport.Rows.Count; i++)
    {

        String column = dtStdReport.Rows[i]["CourseEnroll"].ToString();

        if (column == "BBA")
        {
            Total_BBA++;
            }
            else if (column == "BEIT")
            {
               Total_BEIT++;

            }
            else if (column == "BBS")
            {
               Total_BBS++;

            }
         }
         Week.Rows.Add("BBA", Total_BBA);
         Week.Rows.Add("BEIT", Total_BEIT);
         Week.Rows.Add("BBS", Total_BBS);



         StudentReport.DataContext = Week.DefaultView;



      }

      private void Button_Back(object sender, RoutedEventArgs e)
      {
         this.NavigationService.Navigate(new StudentDetail());
      }
   }
}
```

Chart

```
using DataHandler;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Data;
using Microsoft.Win32;
using System.Windows.Controls.DataVisualization.Charting;

namespace ADCourseWork
{
    /// <summary>
    /// Interaction logic for Chart.xaml
    /// </summary>
    public partial class Chart : Page
    {
        public Chart()
        {
            InitializeComponent();
            LoadPieChartData();
        }
        public void LoadPieChartData()
        {
            var dataSet = new DataSet();
            if (System.IO.File.Exists(@"file\StudentReport.xml"))
            {
                dataSet.ReadXml(@"file\StudentReport.xml");

                DataTable dtStdReport = dataSet.Tables[0];

                int Total_BBA = 0;
                int Total_BEIT = 0;
                int Total_BBS = 0;

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


                for (int i = 0; i < dtStdReport.Rows.Count; i++)
                {

                    String column = dtStdReport.Rows[i]["CourseEnroll"].ToString();

                    if (column == "BBA")
                    {
                        Total_BBA++;

                    }
```

```
          else if (column == "BEIT")
          {
             Total_BEIT++;

          }
          else if (column == "BBS")
          {
             Total_BBS++;

          }
       }
       Week.Rows.Add("BBA", Total_BBA);
       Week.Rows.Add("BEIT", Total_BEIT);
       Week.Rows.Add("BBS", Total_BBS);

       ((PieSeries)chartEnroll).ItemsSource =
    new KeyValuePair<string, int>[]{
    new KeyValuePair<string,int>("BBA", Total_BBA),
    new KeyValuePair<string,int>("BEIT", Total_BEIT),
    new KeyValuePair<string,int>("BBS", Total_BBS)};

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

 }

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

```
        {
            this.NavigationService.Navigate(new StudentDetail()) ;
        }
    }
}
```