S.T.HINDU COLLEGE

Nagercoil-629 002

DEPARTMENT OF COMPUTER APPLICATIONS

[Estd.: 1952]

PRACTICAL
RECORD OF
DOT NET TECHNOLOGIES LAB

2025

S.T.HINDU COLLEGE

Nagercoil-629 002

DEPARTMENT OF COMPUTER APPLICATIONS

[Estd.: 1952]

CERTIFICATE

This is to Certify that Mr/Ms…… Register Number…… is a bonafide student of II M.C.A in the academic year 2025-2026 has completed the work in Dot Net Technologies in the computer lab of our college.

STAFF IN-CHARGE **HEAD OF THE DEPARTMENT**

EXTERNAL EXAMINERS

1.

2.

INDEX

STAFF – IN CHARGE

EXP.NO:1	
DATE:	

AIM:

Procedure :

Program:

```
using System;

namespace PolymorphismDemo
{
   class Animal
   {
      public virtual void Speak()
      {
         Console.WriteLine("Animal speaks");
      }

      public void Info()
      {
         Console.WriteLine("This is a general animal.");
      }

      public void Info(string type)
      {
         Console.WriteLine("This is a " + type);
      }
   }

   class Dog : Animal
   {
      public override void Speak()
      {
         Console.WriteLine("Dog barks");
      }

      public void Info(string type, string breed)
      {
         Console.WriteLine("This is a " + type + " of breed " + breed);
      }
   }

   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("=== Method Overriding ===");
```

```
        Animal a = new Animal();
        Dog d = new Dog();
        Animal ad = new Dog();

        a.Speak();
        d.Speak();
        ad.Speak();

        Console.WriteLine("\n=== Method Overloading ===");
        a.Info();
        a.Info("Mammal");
        d.Info("Dog", "Labrador");

        Console.ReadLine();
    }
  }
}
```

Output:

```
=== Method Overriding ===
Animal speaks
Dog barks
Dog barks

=== Method Overloading ===
This is a general animal.
This is a Mammal
This is a Dog of breed Labrador
```

Result:

EXP.NO:2	
DATE:	

Aim:

Procedure:

Program:

```
using System;

namespace ClassObjectDemo
{
   class Student
   {
      public string name;
      public int age;

      public void ShowDetails()
      {
         Console.WriteLine("Student Name: " + name);
         Console.WriteLine("Student Age: " + age);
      }
   }

   class Program
   {
      static void Main(string[] args)
      {
         Student s1 = new Student();
         s1.name = "Arun";
         s1.age = 20;
         s1.ShowDetails();

         Console.ReadLine();
      }
   }
}
```

Output:
Student Name: Arun
Student Age: 20

Result:

EXP.NO:3	
DATE:	

Aim:

Procedure :

Program:

```
namespace MultilevelInheritanceDemo
{
   class Person
   {
      public void DisplayPerson()
      {
         Console.WriteLine("I am a Person.");
      }
   }

   class Student : Person
   {
      public void DisplayStudent()
      {
         Console.WriteLine("I am a Student.");
      }
   }

   class CollegeStudent : Student
   {
      public void DisplayCollegeStudent()
      {
         Console.WriteLine("I am a College Student.");
      }
   }

   class Program
   {
      static void Main(string[] args)
      {
         CollegeStudent cs = new CollegeStudent();
         cs.DisplayPerson();
         cs.DisplayStudent();
         cs.DisplayCollegeStudent();
         Console.ReadLine();
      }
   }
}
```

Output:

I am a Person.
I am a Student.
I am a College Student.

Result:

EXP.NO:4	
DATE:	

Aim:

Procedure :

Program:

```
using System;

namespace InterfaceDemo
{
   interface Area
   {
      void AreaCal(int l, int b);
   }

   class Rectangle : Area
   {
      public void AreaCal(int l, int b)
      {
         Console.WriteLine("\nArea of Rectangle is " + (l * b));
      }
   }

   class RectArea
   {
      static void Main(string[] args)
      {
         Rectangle s1 = new Rectangle();

         Console.WriteLine("\tINTERFACE\n");

         Console.Write("Enter Length Value: ");
         int l = Convert.ToInt32(Console.ReadLine());

         Console.Write("Enter Breadth Value: ");
         int b = Convert.ToInt32(Console.ReadLine());

         s1.AreaCal(l, b);

         Console.ReadLine();
      }
   }
}
```

Output:

INTERFACE

Enter Length Value: 8
Enter Breadth Value: 6

Area of Rectangle is 48

Result:

EXP.NO:5	
DATE:	

Aim:

Procedure :

Program:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;

namespace Exchndl
{
   using System;
   class ExceptionHandling
   {
      static void Main()
      {
         Console.Write("\t\tException Handling\n\n");

         try
         {
            Console.Write("Enter the totaL Number (Greater than 1) : ");
            int n = int.Parse(Console.ReadLine());
            int[] arr = new int[n];
            Console.WriteLine();

            for (int i = 0; i < n; i++)
            {
               Console.Write("Enter {0} element: ", i);
               arr[i] = Convert.ToInt32(Console.ReadLine());
            }

            int divisionResult = arr[1] / arr[0];

            Console.Write("\nEnter the index of the element to be searched: ");
            int search = int.Parse(Console.ReadLine());

            Console.WriteLine("Array element: {0}  is at index : {1}",
arr[search], search);
         }
         catch (IndexOutOfRangeException)
         {
            Console.WriteLine("Given Number is out of Range");
         }
```

```
        catch (FormatException)
        {
           Console.WriteLine("Acccept only positive integer");
        }
        catch (DivideByZeroException)
        {
           Console.WriteLine("\nFirst Number should not be Zero");
        }
        finally
        {
           Console.WriteLine("\nProgram has ended....\n");
        }
     }
  }
}
```

Output:

```
      Exception Handling

Enter the totaL Number (Greater than 1) : 3

Enter 0 element: 5
Enter 1 element: 10
Enter 2 element: 20

Enter the index of the element to be searched: 2
Array element: 20  is at index : 2

Program has ended....
```

Result:

EXP.NO:6	
DATE:	

Aim:

Procedure :

Program:

```
using System;
using System.IO;
using System.Text;

public class RandomAccessExample
{
   public static void Main(string[] args)
   {
      string filePath = Path.Combine(Environment.CurrentDirectory,
"random_file.txt");

      File.WriteAllText(filePath,
         "This is the first record.\n\nThis is the second record.\nThis is the
third record.");

      try
      {
         using (FileStream fileStream = new FileStream(filePath,
FileMode.Open, FileAccess.ReadWrite))
         {
            long secondLineStart = -1;
            int byteRead;
            int newlineCount = 0;
            long position = 0;

            while ((byteRead = fileStream.ReadByte()) != -1)
            {
               position++;
               if (byteRead == '\n')
               {
                  newlineCount++;
                  if (newlineCount == 1)
                  {
                     secondLineStart = position;
                     break;
                  }
               }
            }

            if (secondLineStart != -1)
```

```
            {
                fileStream.Seek(secondLineStart, SeekOrigin.Begin);

                byte[] buffer = Encoding.UTF8.GetBytes("Overwritten second
record.");
                fileStream.Write(buffer, 0, buffer.Length);

                fileStream.Seek(secondLineStart, SeekOrigin.Begin);
                byte[] readBuffer = new byte[buffer.Length];
                fileStream.Read(readBuffer, 0, buffer.Length);
                string modifiedRecord = Encoding.UTF8.GetString(readBuffer);

                Console.WriteLine("Modified Record:");
                Console.WriteLine(modifiedRecord);
            }
            else
            {
                Console.WriteLine("Second line not found.");
            }
        }
    }
    catch (Exception e)
    {
        Console.WriteLine("Error: " + e.Message);
    }
  }
}
```

Output:

```
Modified Record:
Overwritten second record.
Press any key to continue . . .
```

Result:

EXP.NO:7	
DATE:	

Aim:

Procedure :

Program:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace EmployeeManagementLINQ
{
   // Employee class
   public class Employee
   {
      public int ID;
      public string Name;
      public int DepartmentID;
      public double Salary;
   }

   // Department class
   public class Department
   {
      public int ID;
      public string Name;
   }

   class Program
   {
      static void Main(string[] args)
      {
         // Sample department data
         List<Department> departments = new List<Department>()
         {
            new Department { ID = 1, Name = "HR" },
            new Department { ID = 2, Name = "IT" },
            new Department { ID = 3, Name = "Finance" }
         };

         // Sample employee data
         List<Employee> employees = new List<Employee>()
         {
            new Employee { ID = 101, Name = "Alice", DepartmentID = 1, Salary
= 50000 },
```

```
            new Employee { ID = 102, Name = "Bob", DepartmentID = 2, Salary
= 60000 },
            new Employee { ID = 103, Name = "Charlie", DepartmentID = 2,
Salary = 70000 },
            new Employee { ID = 104, Name = "David", DepartmentID = 3,
Salary = 55000 },
            new Employee { ID = 105, Name = "Eve", DepartmentID = 1, Salary
= 52000 }
        };

        Console.WriteLine("=== All Employees ===");
        foreach (var emp in employees)
        {
            Console.WriteLine("ID: " + emp.ID + ", Name: " + emp.Name + ",
Dept ID: " + emp.DepartmentID + ", Salary: " + emp.Salary);
        }

        Console.WriteLine("\n=== Employees in IT Department ===");
        var itEmployees = from e in employees
                          where e.DepartmentID == 2
                          select e;

        foreach (var emp in itEmployees)
        {
            Console.WriteLine("Name: " + emp.Name + ", Salary: " +
emp.Salary);
        }

        Console.WriteLine("\n=== Employees Sorted by Salary (Descending)
===");
        var sorted = from e in employees
                     orderby e.Salary descending
                     select e;

        foreach (var emp in sorted)
        {
            Console.WriteLine(emp.Name + " - " + emp.Salary);
        }

        Console.WriteLine("\n=== Group Employees by Department ===");
        var grouped = from e in employees
                      group e by e.DepartmentID into deptGroup
```

```
                select deptGroup;

        foreach (var group in grouped)
        {
            string deptName = departments.FirstOrDefault(d => d.ID ==
group.Key).Name;
            Console.WriteLine("\nDepartment: " + deptName);
            foreach (var emp in group)
            {
                Console.WriteLine(" - " + emp.Name);
            }
        }

        Console.ReadLine();
    }
  }
}
```

Output:

```
=== All Employees ===
ID: 101, Name: Alice, Dept ID: 1, Salary: 50000
ID: 102, Name: Bob, Dept ID: 2, Salary: 60000
ID: 103, Name: Charlie, Dept ID: 2, Salary: 70000
ID: 104, Name: David, Dept ID: 3, Salary: 55000
ID: 105, Name: Eve, Dept ID: 1, Salary: 52000

=== Employees in IT Department ===
Name: Bob, Salary: 60000
Name: Charlie, Salary: 70000

=== Employees Sorted by Salary (Descending) ===
Charlie - 70000
Bob - 60000
David - 55000
Eve - 52000
Alice - 50000
```

=== Group Employees by Department ===

Department: HR
- Alice
- Eve

Department: IT
- Bob
- Charlie

Department: Finance
- David

Result:

EXP.NO:8	
DATE:	

Aim:

Procedure :

Program:

Student.cs:

```
namespace StudentManagementSystem1
{
   public class Student
   {
      public int Id { get; set; }
      public string Name { get; set; }
      public string Course { get; set; }
      public int Age { get; set; }
   }
}
```

StudentRepository.cs:

```
using System.Collections.Generic;
namespace StudentManagementSystem1
{
   public static class StudentRepository
   {
      private static List<Student> students = new List<Student>();
      private static int nextId = 1;

      public static List<Student> GetAll()
      {
         return students;
      }

      public static void Add(Student student)
      {
         student.Id = nextId++;
         students.Add(student);
      }
   }
}
```

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
  <title>Student List</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>All Students</h2>
      <asp:GridView ID="GridViewStudents" runat="server"
AutoGenerateColumns="false">
        <Columns>
          <asp:BoundField DataField="Id" HeaderText="ID" />
          <asp:BoundField DataField="Name" HeaderText="Name" />
          <asp:BoundField DataField="Course" HeaderText="Course" />
          <asp:BoundField DataField="Age" HeaderText="Age" />
        </Columns>
      </asp:GridView>
      <br />
      <asp:HyperLink ID="lnkAdd" runat="server"
NavigateUrl="AddStudent.aspx">Add New Student</asp:HyperLink>
    </div>
  </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;
using System.Web.UI;
using StudentManagementSystem1;
public partial class _Default : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
    if (!IsPostBack)
    {
      GridViewStudents.DataSource = StudentRepository.GetAll();
```

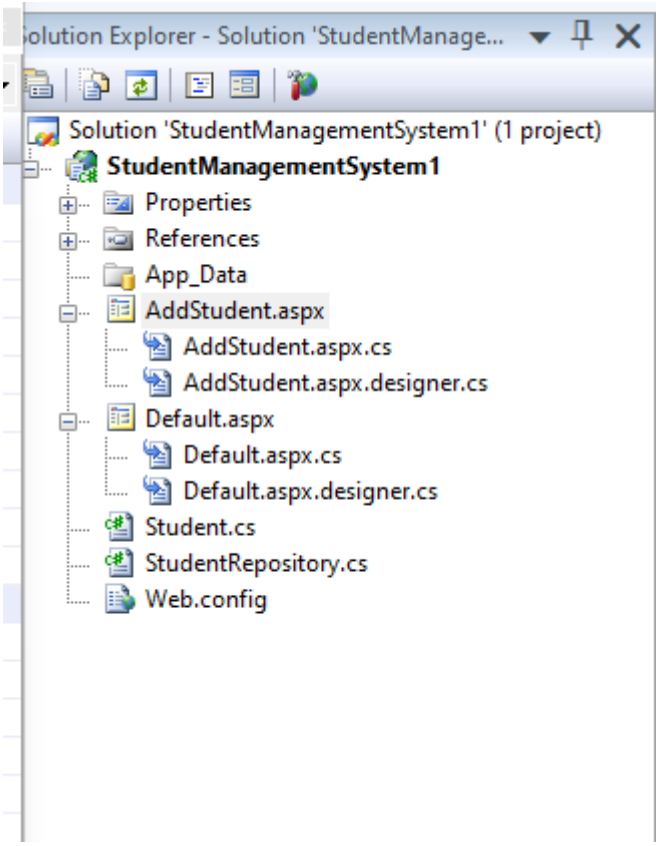
```
        GridViewStudents.DataBind();
      }
    }
}
```

AddStudent.aspx:

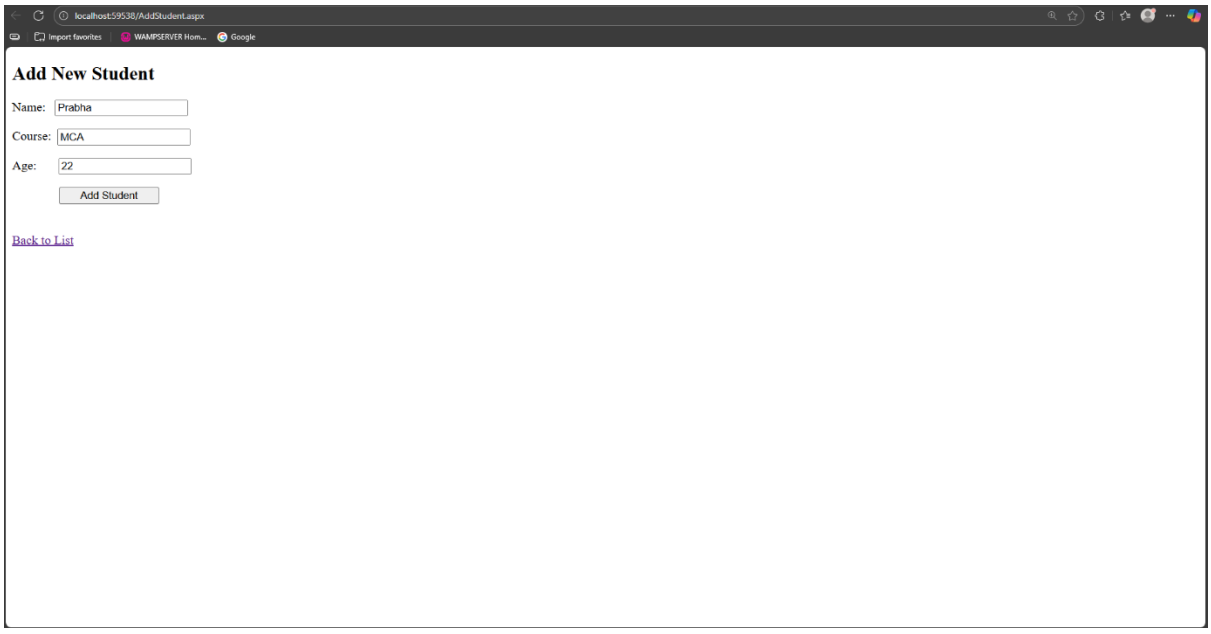
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="AddStudent.aspx.cs" Inherits="AddStudent" %>
<!DOCTYPE html>
<html>
<head>
  <title>Add Student</title>
</head>
<body>
  <h2>Add New Student</h2>
  <form id="form1" runat="server">
    <p>
      Name:   <asp:TextBox ID="txtName" runat="server" />
    </p>
    <p>
      Course:  <asp:TextBox ID="txtCourse" runat="server" />
    </p>
    <p>
      Age:       <asp:TextBox
ID="txtAge" runat="server" />
    </p>

          &nb
sp;   
    <asp:Button ID="btnAdd" runat="server" Text="Add Student"
      OnClick="btnAdd_Click" Width="127px" />
    <br /><br />
    <asp:Label ID="lblMessage" runat="server"
ForeColor="Green"></asp:Label>
    <br /><a href="Default.aspx">Back to List</a>
  </form>
</body>
</html>
```

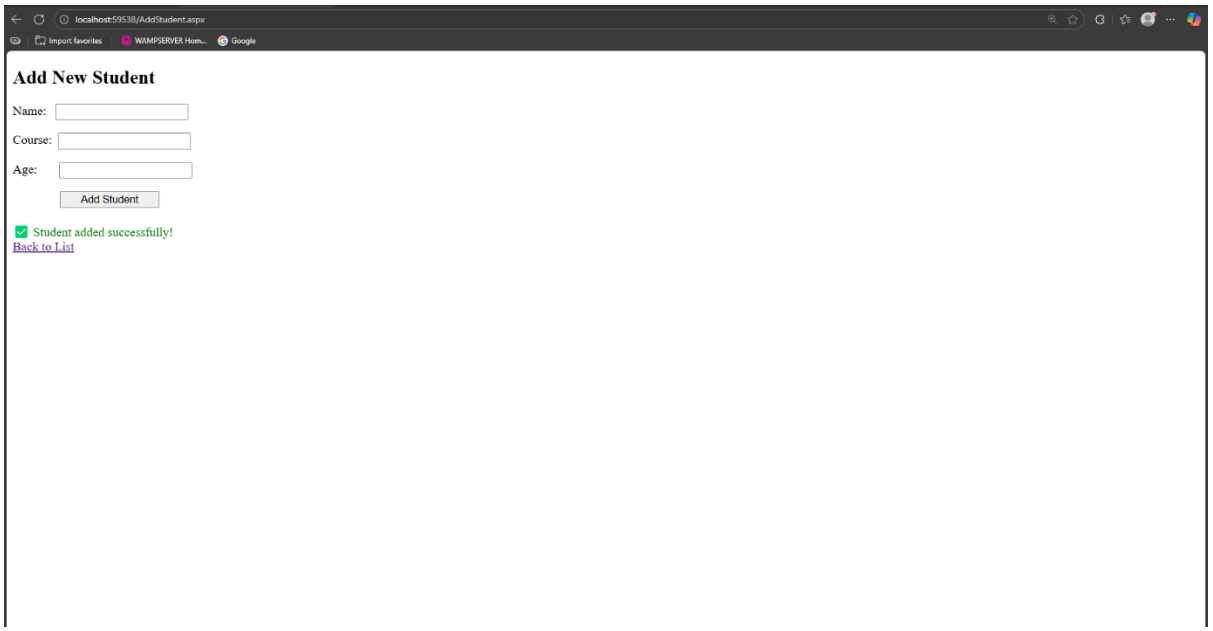
AddStudent.aspx.cs:

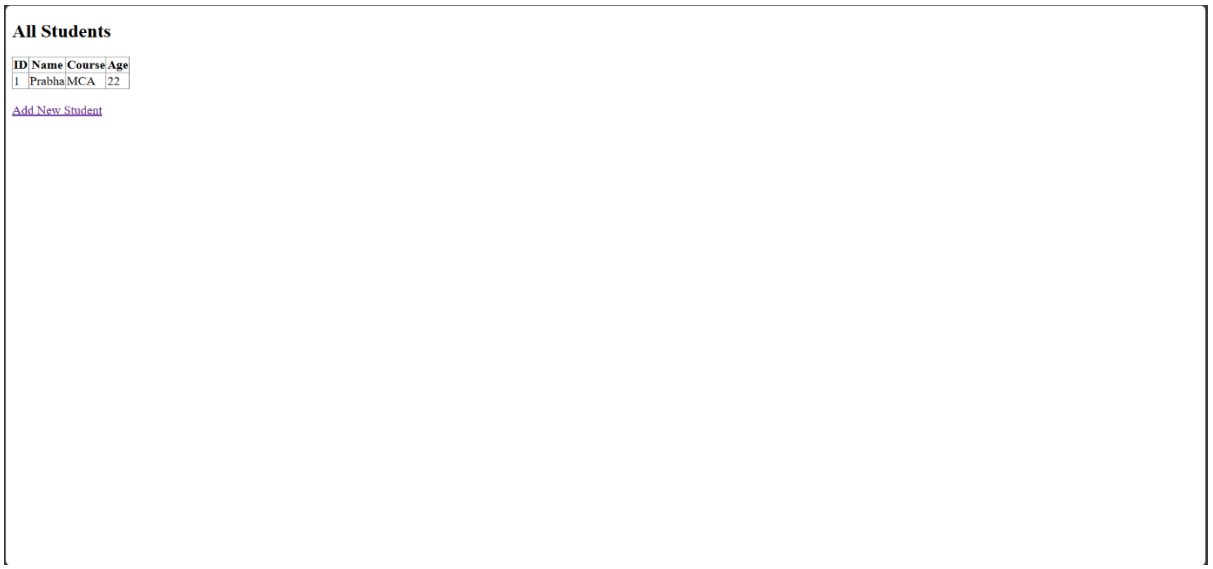
```
using System;
using StudentManagementSystem1;
public partial class AddStudent : System.Web.UI.Page
{
   protected void btnAdd_Click(object sender, EventArgs e)
   {
      try
      {
         Student student = new Student
         {
            Name = txtName.Text,
            Course = txtCourse.Text,
            Age = int.Parse(txtAge.Text)
         };
         StudentRepository.Add(student);
         lblMessage.Text = " ✅ Student added successfully!";
         // Optional: clear fields
         txtName.Text = "";
         txtCourse.Text = "";
         txtAge.Text = "";
      }
      catch (FormatException)
      {
         lblMessage.Text = " ❌ Please enter a valid number for Age.";
      }
      catch (Exception ex)
      {
         lblMessage.Text = " ❌ Error: " + ex.Message;
      }
   }
}
```


olution Explorer - Solution 'StudentManage...
Solution 'StudentManagementSystem1' (1 project)
StudentManagementSystem1
Properties
References
App_Data
AddStudent.aspx
AddStudent.aspx.cs
AddStudent.aspx.designer.cs
Default.aspx
Default.aspx.cs
Default.aspx.designer.cs
Student.cs
StudentRepository.cs
Web.config

Output:


localhost:59538/AddStudent.aspx
Import favorites
WAMPSERVER Hom...
Google
Add New Student
Name: Prabha
Course: MCA
Age: 22
Add Student
Back to List


localhost:59538/AddStudent.aspx
Import favorites
WAMPSERVER Hom...
Google
Add New Student
Name:
Course:
Age:
Add Student
Student added successfully!
Back to List


All Students
ID Name Course Age
1 Prabha MCA 22
Add New Student

Result:

EXP.NO:9	
DATE:	

Aim:

Procedure:

Program:

APP.xaml:

```
<Application
x:Class="App2.App"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
RequestedTheme="Light">
</Application>
```

App.xaml.cs:

```
using Windows.ApplicationModel.Activation;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;

namespace App2
{
sealed partial class App : Application
{
public App()
{
this.InitializeComponent();
}

protected override void OnLaunched(LaunchActivatedEventArgs e)
{
Frame rootFrame = Window.Current.Content as Frame;
```

```
if (rootFrame == null)
{
rootFrame = new Frame();
Window.Current.Content = rootFrame;
}

if (rootFrame.Content == null)
{
rootFrame.Navigate(typeof(MainPage), e.Arguments);
}

Window.Current.Activate();
}
}
}
```

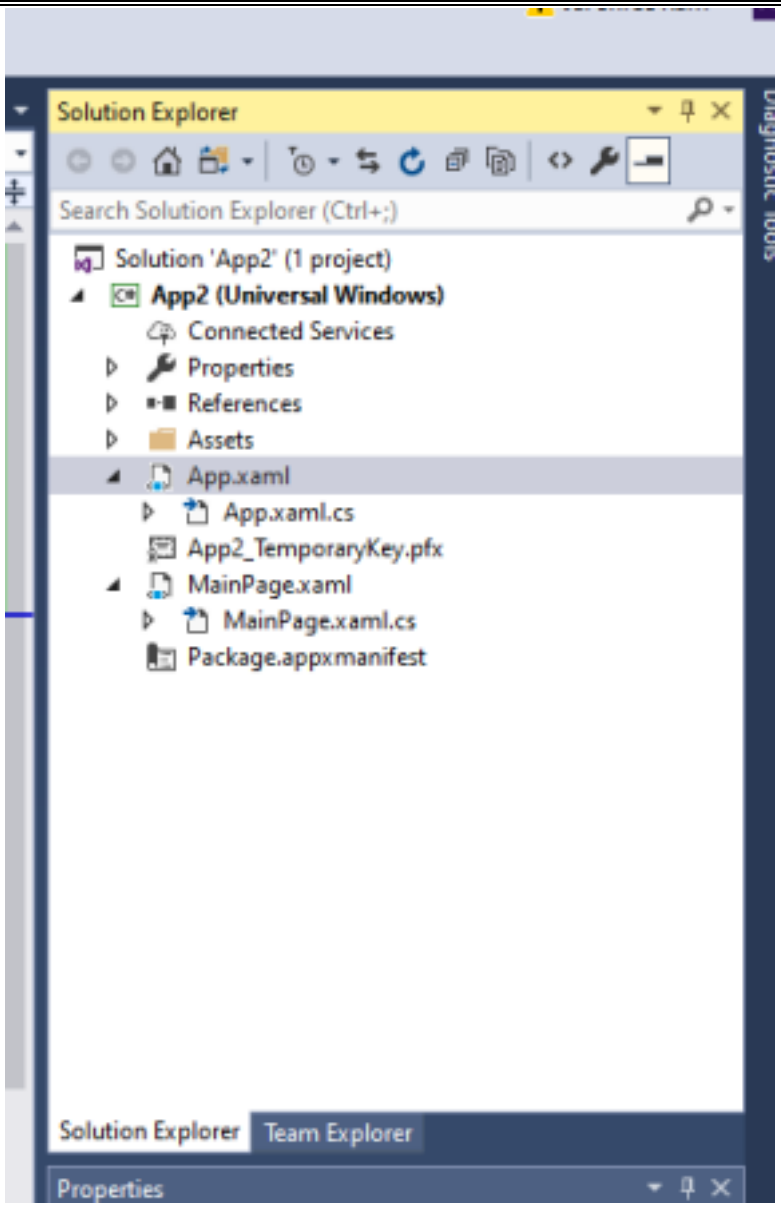
MainPage.xaml:

```
<Page
x:Class="App2.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:local="using:App2"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
```

```
mc:Ignorable="d">
<Grid Background="{ThemeResource
ApplicationPageBackgroundThemeBrush}">
<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
<Button Content="Click Me" Click="Button_Click"/>
<TextBlock x:Name="txtMessage" FontSize="24" Margin="0,20,0,0"/>
</StackPanel>
</Grid>
</Page>
```

MainPage.xaml.cs:

```
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
namespace App2
{
public sealed partial class MainPage : Page
{
public MainPage()
{
this.InitializeComponent();
}
private void Button_Click(object sender, RoutedEventArgs e)
{
txtMessage.Text = "Hello, Universal Windows Platform!";
}
}
}
```


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'App2' (1 project)
App2 (Universal Windows)
Connected Services
Properties
References
Assets
App.xaml
App.xaml.cs
App2_TemporaryKey.pfx
MainPage.xaml
MainPage.xaml.cs
Package.appxmanifest
Solution Explorer
Team Explorer
Properties

Output:


App2
Click Me


App2
Click Me
Hello, Universal Windows Platform!

Result: