# ADVANCED WEB PROGRAMMING JOURNAL

# INDEX

| | | | |
|---|---|---|---|
| **7** | **Working with Databases** | | |
| A | Web application to display Databinding using Dropdownlist contol | | |
| B | Web application to display Title of an Employee using database. | | |
| C | Web Application to display data using Disconnected Data Access and Data Binding using GridView control | | |
| **8** | **Working with AJAX and XML.** | | |
| A | Web application to demonstrate reading and writing operation with XML. | | |
| B | Web application to demonstrate Form Security with Authentication and Authorization. | | |
| C | Web application to demonstrate to demonstrate use of various Ajax Controls. | | |
| 9 | Programs to create and use DLL. | | |

## Practical 1 : Working with bacic C# and ASP.NET

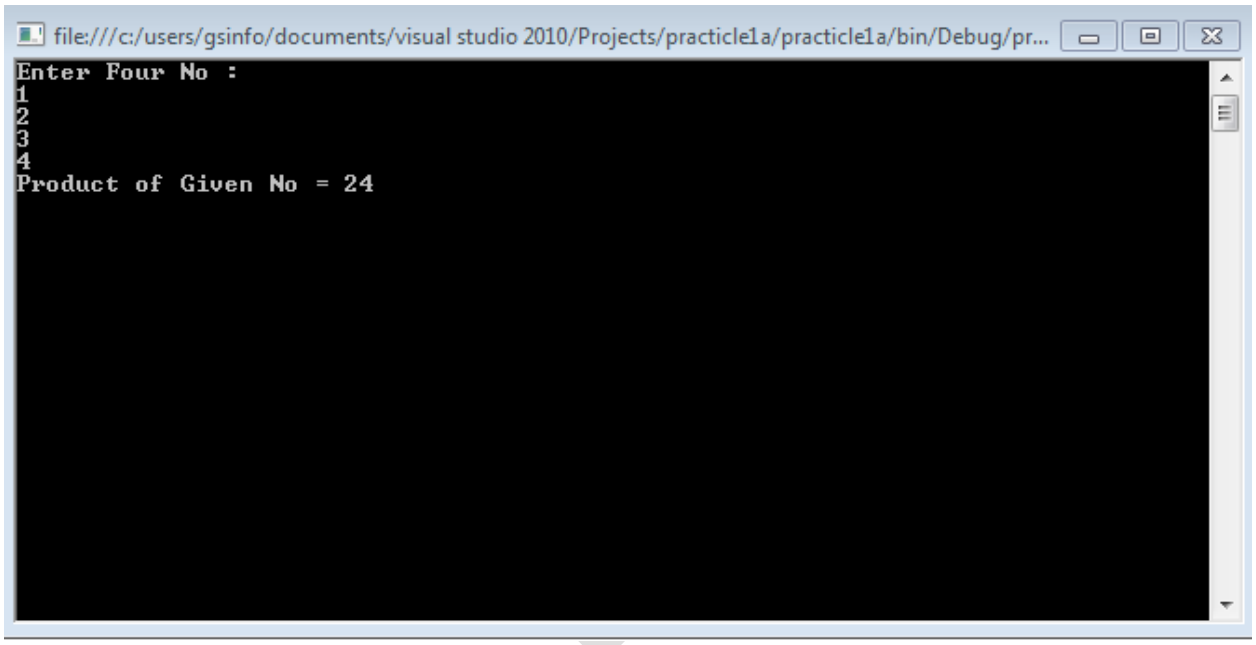
### Practical No: 1(A)

**Aim:** Create an application that obtains four int values from the user and displays the product.

**Source Code:**

```
using System.Text;
namespace practicle1a
{
class Program
{
static void Main(string[] args)
{
int no1, no2, no3, no4, product;
Console.WriteLine("Enter Four No : ");
no1 = Convert.ToInt32(Console.ReadLine());
no2 = Convert.ToInt32(Console.ReadLine());
no3 = Convert.ToInt32(Console.ReadLine());
no4 = Convert.ToInt32(Console.ReadLine());
product = no1 * no2 * no3 * no4;
Console.WriteLine("Product of Given No = " + product);
Console.ReadLine();
}
}
}
```

**Output:**


file:///c:/users/gsinfo/documents/visual studio 2010/Projects/practicle1a/practicle1a/bin/Debug/pr...
Enter Four No :
1
2
3
4
Product of Given No = 24

**Practical No: 1(B)**

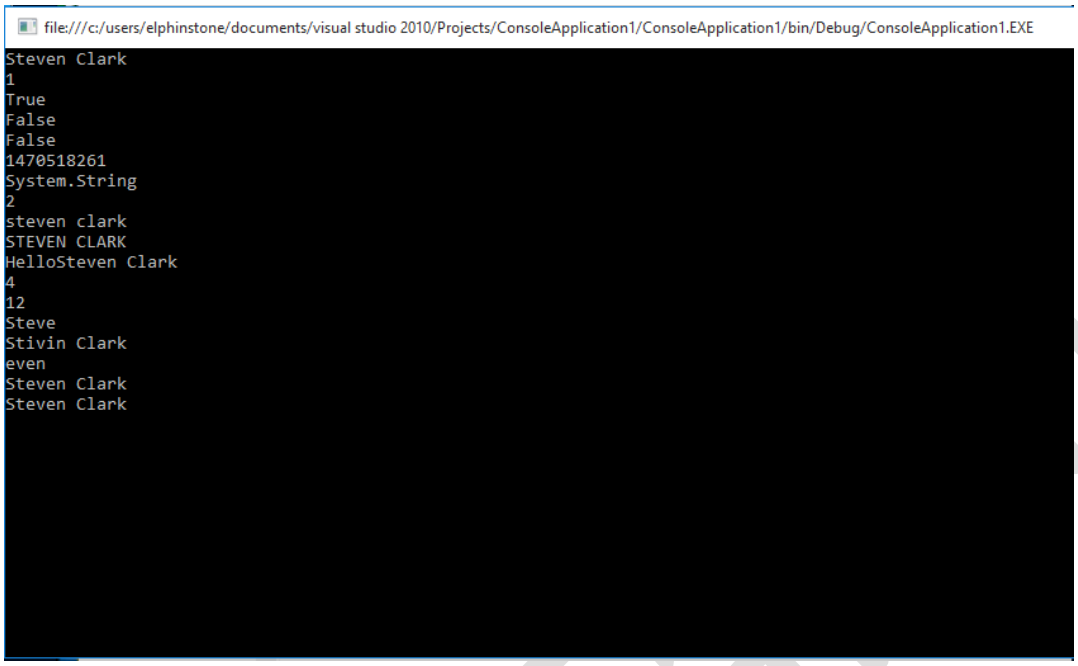
**Aim:** Create an application to demonstrate string operations.

**Source Code:**

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace String_Operations

{

class Program

{

static void Main(string[] args)

{

string firstname;
string lastname;
                    firstname = "Steven Clark";
lastname = "Clark";


Console.WriteLine(firstname.Clone());
// Make String Clone

Console.WriteLine(firstname.CompareTo(lastname));
//Compare two string value and returns 0 for true and 1 for false

Console.WriteLine(firstname.Contains("ven"));
//Check whether specified value exists or not in string

Console.WriteLine(firstname.EndsWith("n"));
//Check whether specified value is the last character of string

Console.WriteLine(firstname.Equals(lastname));
//Compare two string and returns true and false
```

```
Console.WriteLine(firstname.GetHashCode());
//Returns HashCode of String

Console.WriteLine(firstname.GetType());
//Returns type of string

Console.WriteLine(firstname.IndexOf("e"));
//Returns the first index position of specified value the first index position of specified value

Console.WriteLine(firstname.ToLower());
//Covert string into lower case

Console.WriteLine(firstname.ToUpper());
//Convert string into Upper case

Console.WriteLine(firstname.Insert(0, "Hello "));
//Insert substring into string

Console.WriteLine(firstname.LastIndexOf("e")); //Returns the last index position of specified value

Console.WriteLine(firstname.Length);
//Returns the Length of String

Console.WriteLine(firstname.Remove(5));
//Deletes all the characters from begining to specified index.

Console.WriteLine(firstname.Replace('e','i'));
// Replace the character

Console.WriteLine(firstname.Substring(2, 5));
//Returns substring

Console.WriteLine(firstname.ToCharArray());
//Converts an string into char array.

Console.WriteLine(firstname.Trim());
//It removes starting and ending white spaces from string.

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

**Output:**



**Practical No: 1(C)**

**Aim:** Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data entered.

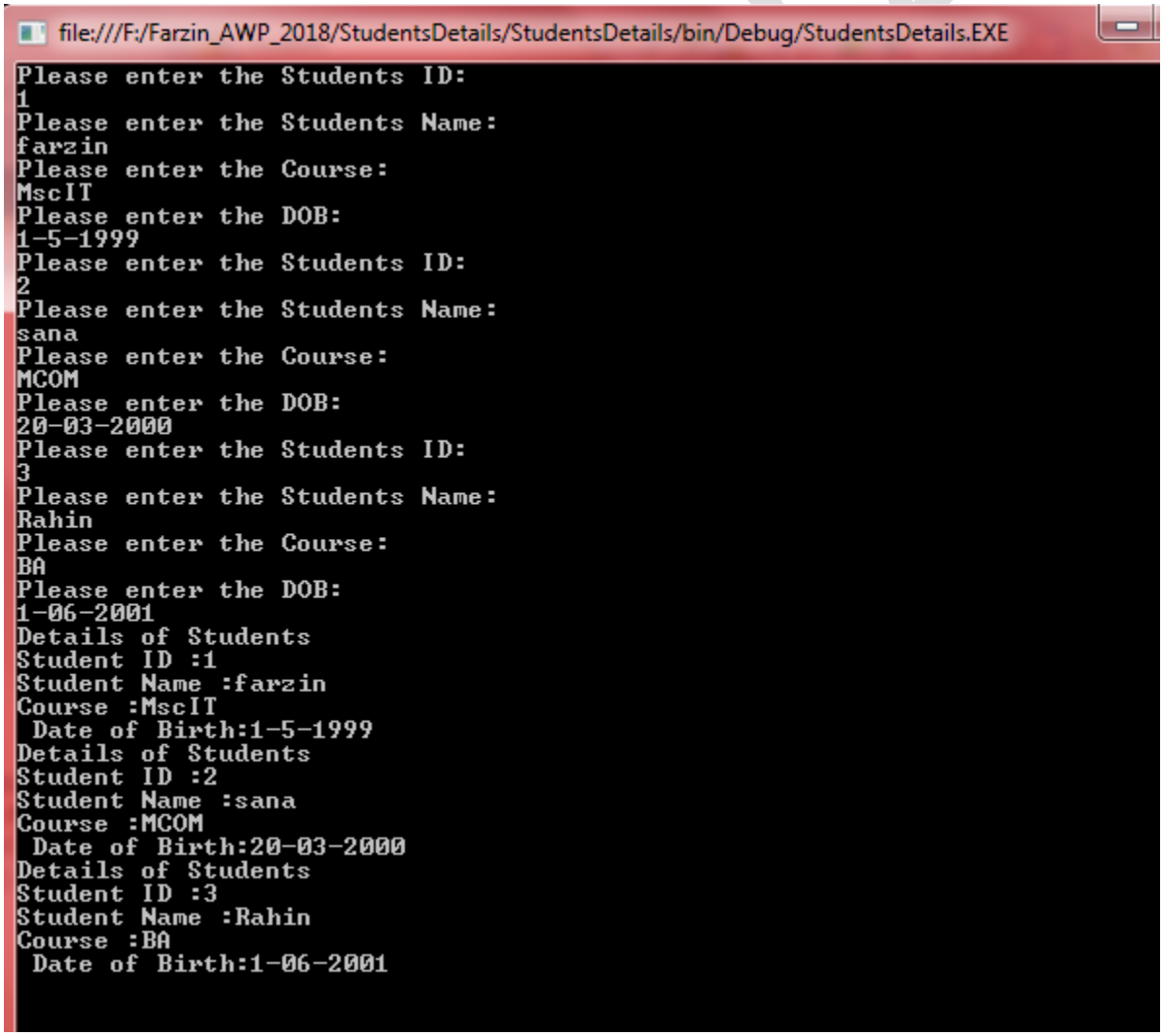
**Source Code:**

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

namespace Student
{
class Program
{
public struct StudentDetails
{
public string StudID;
public string Name;
public string Course;
public string DOB;
}
StudentDetails[,] stu = new StudentDetails[3, 1];
void input()
{
for (int row = 0; row < 3; row++)
{
for (int col = 0; col < 1; col++)
```

```
{
Console.WriteLine("Please enter the Students ID: ");
stu[row, col].StudID = Console.ReadLine();
Console.WriteLine("Please enter the Students Name: ");
stu[row, col].Name = Console.ReadLine();
Console.WriteLine("Please enter the Course: ");
stu[row, col].Course = Console.ReadLine();
Console.WriteLine("Please enter the DOB: ");
stu[row, col].DOB = Console.ReadLine();
}
}
}

void Display()
{
for (int row = 0; row < 3; row++)
{
Console.WriteLine("Details of Students", row + 1);
for (int col = 0; col < 1; col++)
{
Console.WriteLine("Student ID :{0}", stu[row, col].StudID);
Console.WriteLine("Student Name :{0}", stu[row, col].Name);
Console.WriteLine("Course :{0}", stu[row, col].Course);
Console.WriteLine(" Date of Birth:{0}", stu[row, col].DOB);
}
}
}
```

```
static void Main(string[] args)
{
Program p = new Program();
p.input();
p.Display();
Console.ReadLine();
}
}
}
```

**Output:**



**Practical No: 1(D)**

**Aim:** Create an application to demonstrate following operations

i. Generate Fibonacci series.

ii. Test for prime numbers.

iii. Test for vowels.

iv. Use of foreach loop with arrays

v. Reverse a number and find sum of digits of a number.

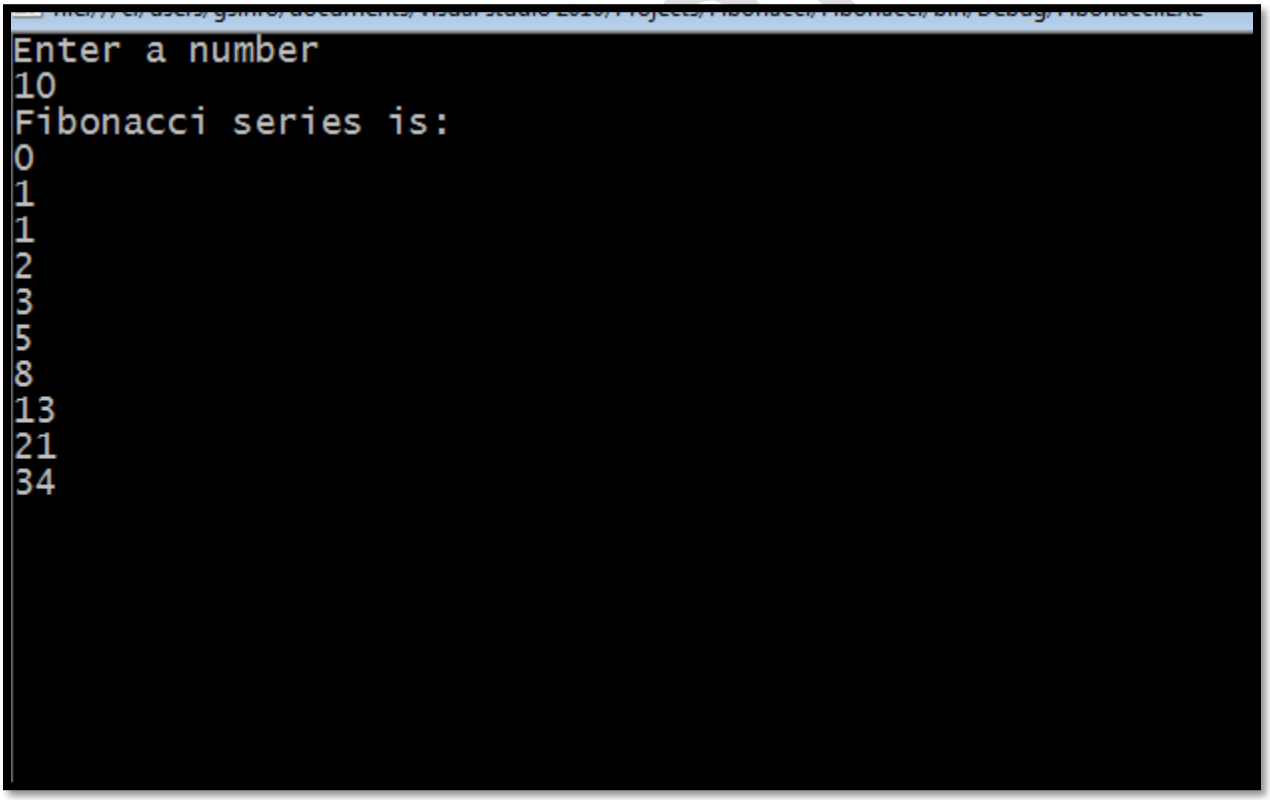
**Source Code:**

i). Fibonacci series:

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

namespace Fibonacci
{
class Program
{
public static int Fibonaci(int n)
{
int a = 0;
int b = 1;
for (int i = 0; i < n; i++)
{
int temp = a;
a = b;
b = temp + b;
}
return a;
}
static void Main(string[] args)
{
```

```
Console.WriteLine("Enter a number");
int n = Int16.Parse(Console.ReadLine());
Console.WriteLine("Fibonacci series is:");
for (int i = 0; i < n; i++)
{
Console.WriteLine(Fibonaci(i));
}

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

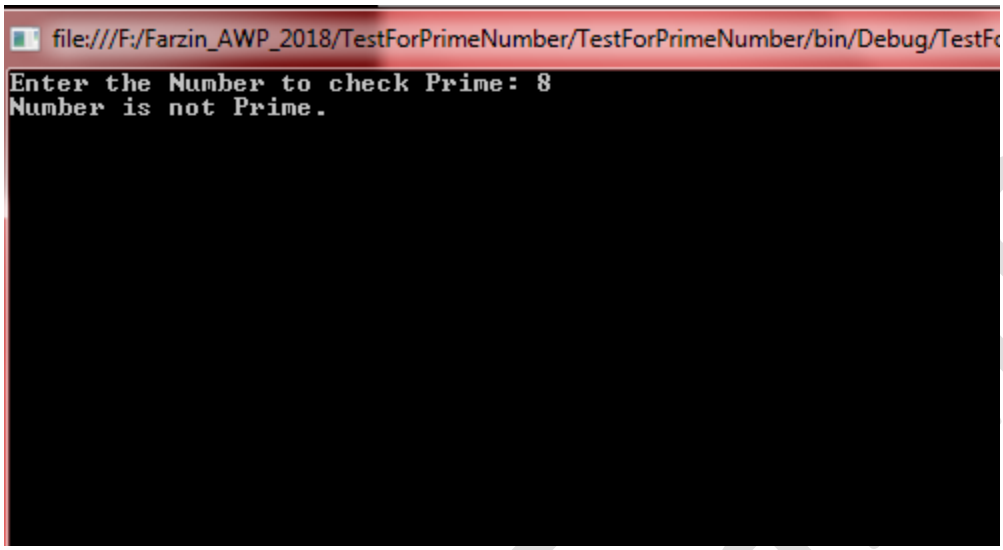
**Output:**



ii). Test for prime numbers:

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

namespace TestForPrimeNumber
{

class Program
{
static void Main(string[] args)
{
int n, i, m = 0, flag = 0;
Console.Write("Enter the Number to check Prime: ");
n = int.Parse(Console.ReadLine());
m = n / 2;

for (i = 2; i <= m; i++)
{

if (n % i == 0)
{
Console.Write("Number is not Prime.");
flag = 1;
break;
}
}

if (flag == 0)
Console.Write("Number is Prime.");

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

**Output:**



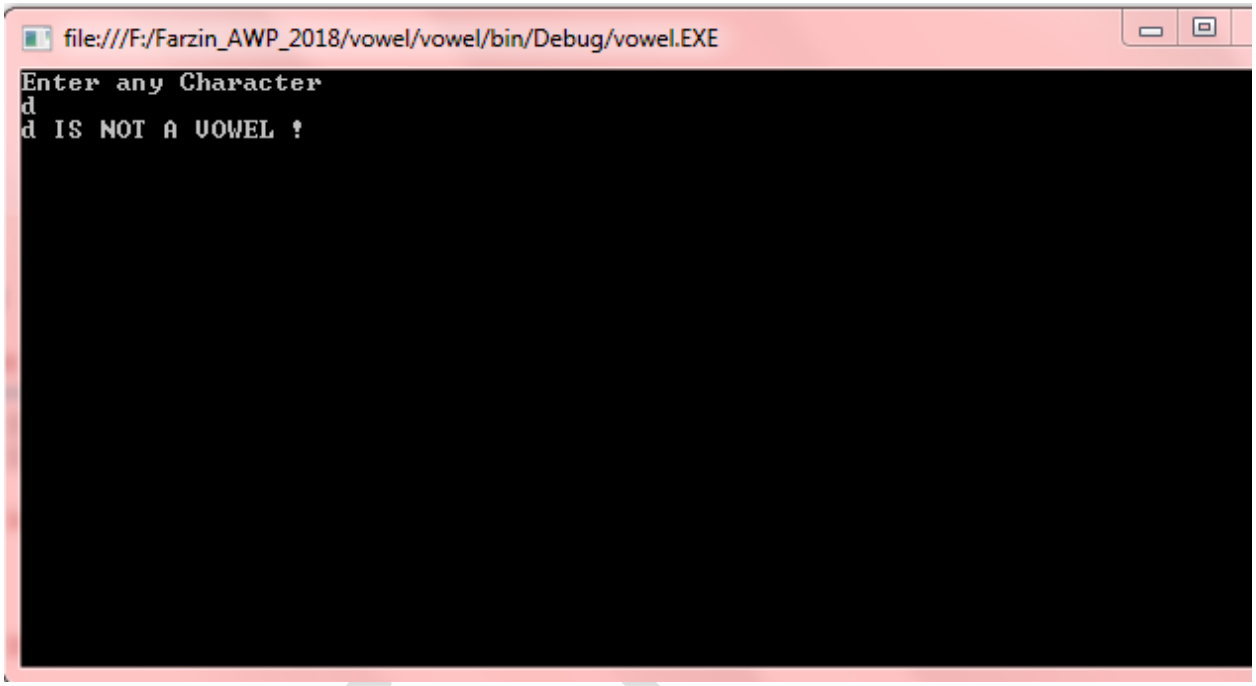
iii). Test for vowels:

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

namespace pract_f_vowels
{
class Program
{
static void Main(string[] args)
{
Console.WriteLine("enter a caracter");
char ch = Convert.ToChar(Console.ReadLine());
if (ch=='a'|| ch== 'A' || ch=='e' || ch=='E' || ch== 'I' || ch=='i' ||
ch=='o' || ch == 'O' || ch== 'u' || ch =='U')
{
Console.WriteLine("{0} is a vowel !" , ch);
}
else
{
Console.WriteLine("{0} IS NOT A VOWEL !",ch);
}
Console.ReadLine();
}
```

}.

}

**Output:**


file:///F:/Farzin_AWP_2018/vowel/vowel/bin/Debug/vowel.EXE
Enter any Character
d
d IS NOT A VOWEL !

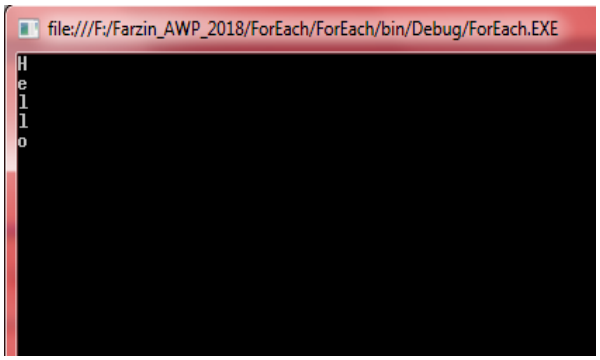
iv). Use of foreach loop with arrays:

The **foreach loop** in C# executes a block of code on **each** element in an array or a collection of items. When executing **foreach loop** it traversing items in a collection or an array. The **foreach loop** is useful for traversing **each** items in an array or a collection of items and displayed one by one.

The major **difference between** the for and **foreach loop** in c# we understand by its working: The for **loop**: The for **loop's** variable always be integer only. The For **Loop** executes the **statement** or block of statements repeatedly until specified expression evaluates to false.

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

namespace ForEach
{
class Program
{
static void Main(string[] args)
{
char[] myArray = {'H','e','l','l','o'};

foreach(char ch in myArray)
{
Console.WriteLine(ch);
}
Console.ReadLine();
}} }
```

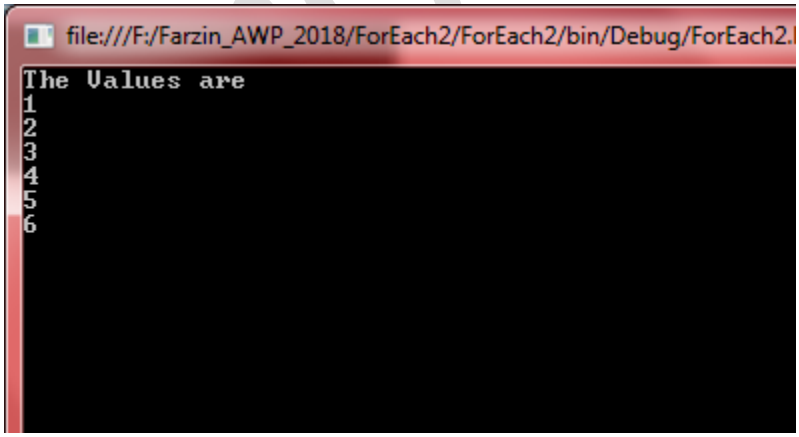
**Output:**




Example 2 : Foreach loop

```
static void Main(string[] args)
{
int[] num = { 1, 2, 3, 4, 5, 6 };
Console.WriteLine("The Values are");
foreach (int i in num)
{
Console.WriteLine(i);
}
Console.ReadLine();
}
```

**Output:**

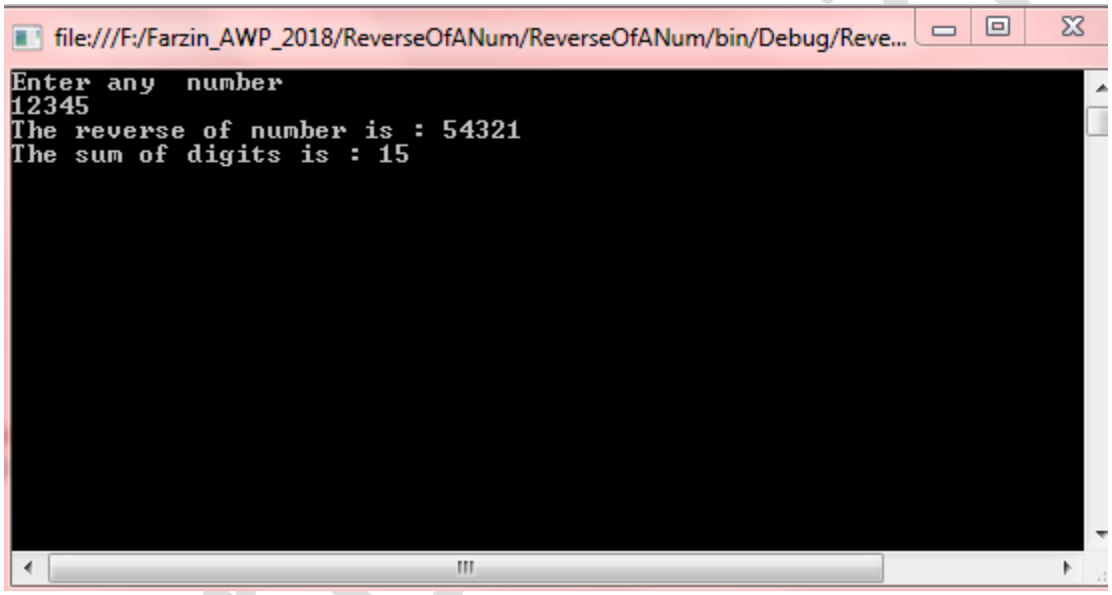


v). Reverse a number and find sum of digits of a number.

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

namespace Reverse_no
{
class Program
{
static void Main(string[] args)
{
int n, m;
int sum = 0;
int rev = 0;
Console.WriteLine("Enter any  number");
n = Convert.ToInt32(Console.ReadLine());
while (n > 0)
{
m = n % 10;
sum += m;
rev = rev * 10 + m;
n = n / 10;
}
```

```
Console.WriteLine("The reverse of number is : {0}", rev);

Console.WriteLine("The sum of digits is : {0}", sum);

Console.ReadLine();

}

}

}
```

**Output:**



## Practical 2 : Working with Object Oriented C# and ASP.NET

### Practical No: 2(A)

**Aim:** Create simple application to perform following operations

i. Finding factorial Value

ii. Money Conversion

iii. Quadratic Equation

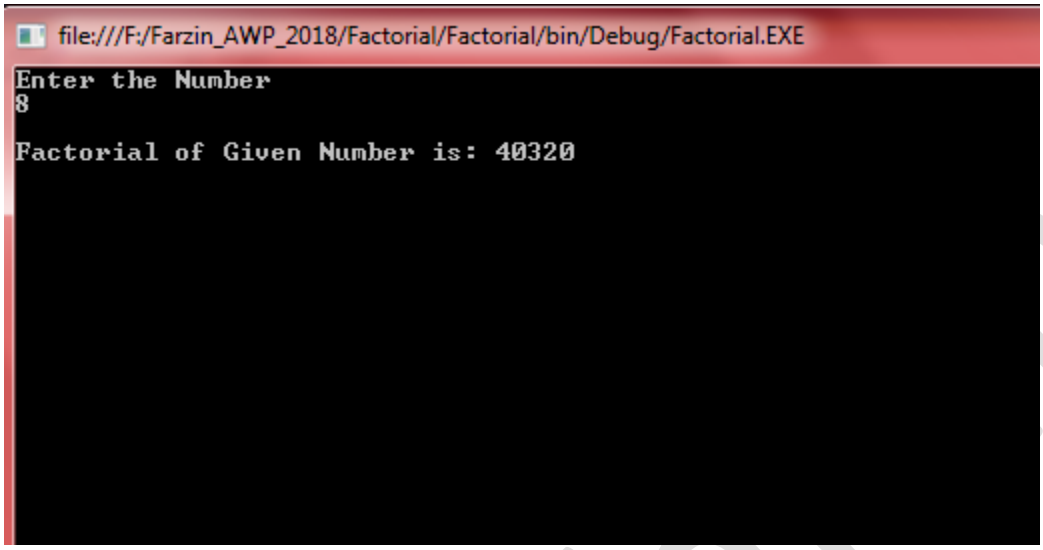
iv. Temperature Conversion.

### Source Code:

(i) Factorial of a Number.

```
namespace Factorial
{
class Program
{
static void Main(string[] args)
{
int i, number, fact;
Console.WriteLine("Enter the Number");
number = int.Parse(Console.ReadLine());
fact = 1 ;
 for (i=1 ; i < = number ; i++ )
{
fact = fact * i;
}
Console.WriteLine("\nFactorial of Given Number is: " + fact);
Console.ReadLine();
}
}
}
```

**Output:**



(ii) Money Conversion:

*Default.aspx:*

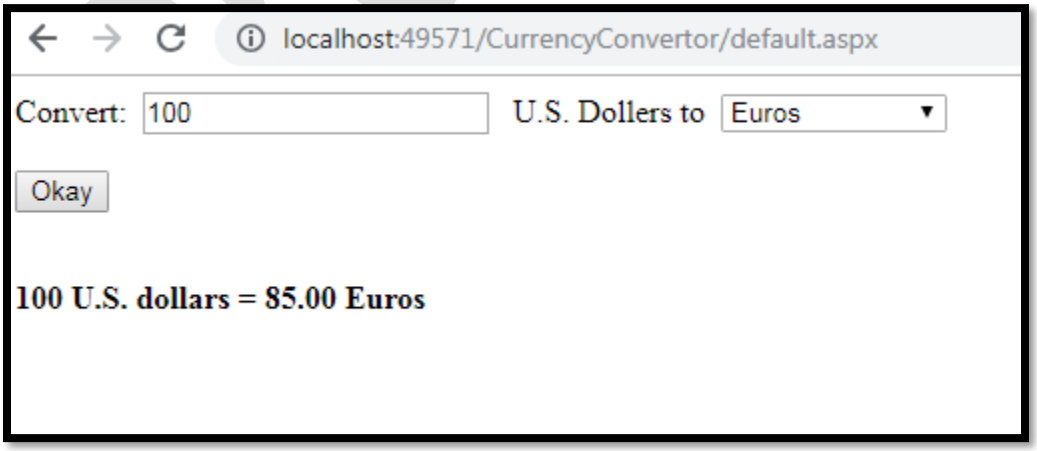
```
<%@ Page Language = "C#" AutoEventWireup = "true" CodeFile = "CurrencyConverter.aspx.cs" Inherits = "CurrencyConverter" %>
<!DOCTYPE html>
<html xmlns = " http://www.w3.org/1999/xhtml ">
<head>  <title > Currency Converter</title>  </head>
 <body>
 <form runat = "server">
  <div>  Convert:  
  <input type = "text" ID = "US" runat = "server" />
    U.S. dollars to  
      <select ID = "Currency" runat = "server" />  <br /> < br />
 <input type = "submit" value = "OK" ID = "Convert"  OnServerClick = "Convert_ServerClick" runat = "server" />  <br /> < br />
  <p style = "font-weight: bold" ID = "Result" runat = "server" > </p>
 </div>  </form>  </body>
</html>
```

Default.aspx.cs :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
if (this.IsPostBack == false)
{
Currency.Items.Add(new ListItem("Euros","0.85"));
Currency.Items.Add(new ListItem("Japnese Yen", "110.33"));
Currency.Items.Add(new ListItem("Canadia Dollar", "1.2"));

}
}
```

```
protected void Convert_ServerClick(object sender, EventArgs e)
{
decimal oldAmount;
bool success = Decimal.TryParse(US.Value, out oldAmount);

if (success)
{
ListItem item = Currency.Items[Currency.SelectedIndex];
decimal newAmout = oldAmount * Decimal.Parse(item.Value);
Result.InnerText = oldAmount.ToString() + " U.S. dollars = ";
Result.InnerText += newAmout.ToString() + " " + item.Text;
}
}
}
```

**Output:**

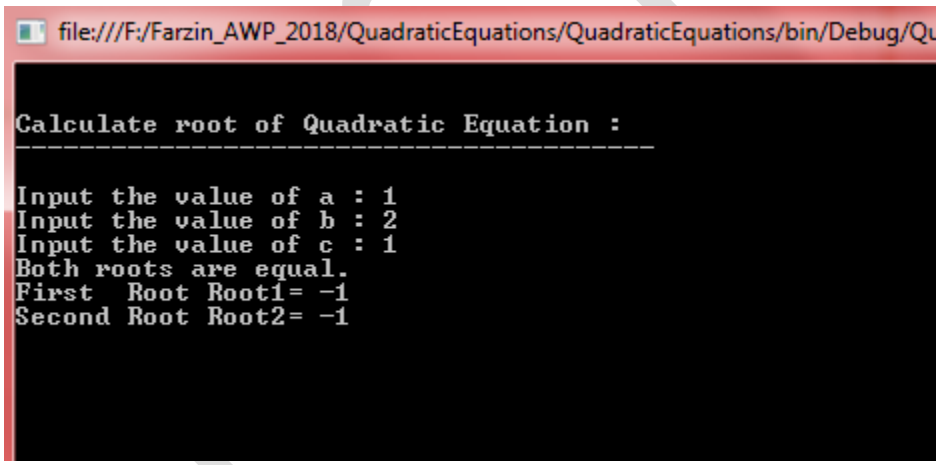


(iii) Quadratic Equation

```
namespace QuadraticEquations
{
class Program
{
static void Main(string[] args)
{
int a,b,c;

double d, x1,x2;
Console.Write("\n\n");
Console.Write("Calculate root of Quadratic Equation :\n");
Console.Write("----------------------------------------");
Console.Write("\n\n");

Console.Write("Input the value of a : ");
a = Convert.ToInt32(Console.ReadLine());
Console.Write("Input the value of b : ");
b = Convert.ToInt32(Console.ReadLine());
Console.Write("Input the value of c : ");
c = Convert.ToInt32(Console.ReadLine());

d=b*b-4*a*c;
if(d==0)
{
Console.Write("Both roots are equal.\n");
x1=-b/(2.0*a);
x2=x1;
Console.Write("First  Root Root1= {0}\n",x1);
Console.Write("Second Root Root2= {0}\n",x2);
}
else if(d>0)
{
Console.Write("Both roots are real and diff-2\n");

x1=(-b+Math.Sqrt(d))/(2*a);
x2=(-b-Math.Sqrt(d))/(2*a);

Console.Write("First  Root Root1= {0}\n",x1);
```

```
Console.Write("Second Root root2= {0}\n",x2);
}
else
Console.Write("Root are imeainary;\nNo Solution. \n\n");
Console.ReadLine();
}
}
}
```

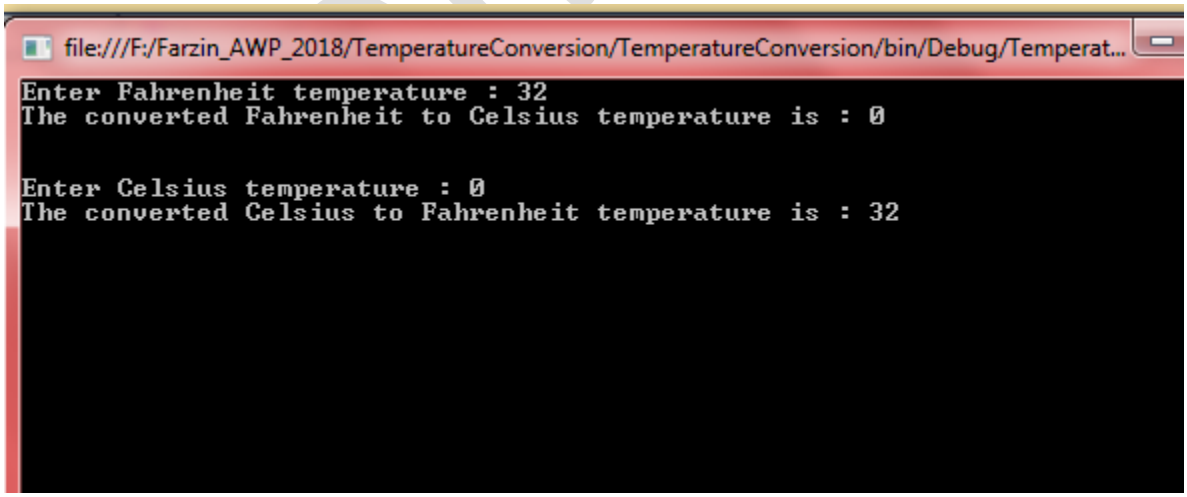
**Output:**



(iv) Temperature Conversion

```
namespace TemperatureConversion
{
class Program
{
static void Main(string[] args)
{
Console.Write("Enter Fahrenheit temperature : ");
double fahrenheit = Convert.ToDouble(Console.ReadLine());
double celsius = (fahrenheit - 32) * 5 / 9;
Console.WriteLine("The converted Fahrenheit to Celsius temperature is : " + celsius);
Console.WriteLine("\n");

Console.Write("Enter Celsius temperature : ");
celsius = Convert.ToDouble(Console.ReadLine());
fahrenheit = (celsius*9) /5 +32;
Console.WriteLine("The converted Celsius to Fahrenheit temperature is : " + fahrenheit  );
Console.ReadLine();
}
}
}
```

**Output:**



**Practical No: 2(B)**

**Aim:** Create simple application to demonstrate use of following concepts :

i. Function Overloading

ii. Inheritance (all types)

iii. Constructor overloading

iv. Interfaces.

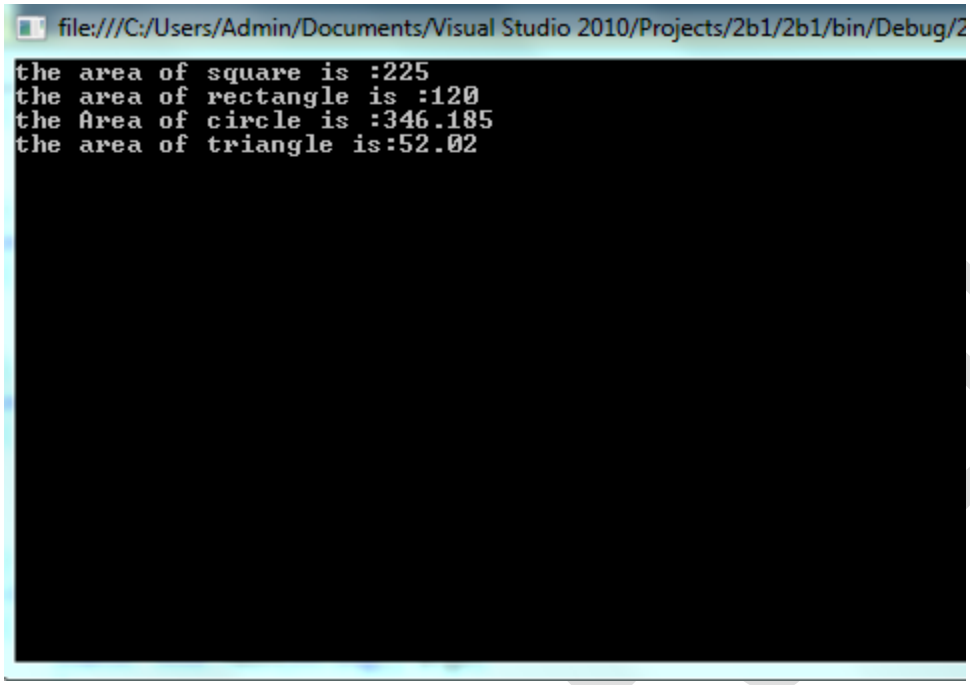
**Source Code:**

(i) Function Overloading

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

namespace function_overload
{
public class Shape
{
public void Area(int side)
{
int SqArea = side * side;
Console.WriteLine("the area of square is :" + SqArea);
}
public void Area(int l, int b)
{
int RectArea = l * b;
```

```
Console.WriteLine("the area of rectangle is :" + RectArea);
}
public void Area(double r)
{
double CircArea = 3.14 * r * r;
Console.WriteLine("the Area of circle is :" + CircArea);
}
public double Area(double b, double h)
{
double TriArea = (b * h) / 2;
Console.WriteLine("the area of triangle is:" + TriArea);
return TriArea;
}
class Program
{
static void Main(string[] args)
{
Shape s = new Shape();
s.Area(15);
s.Area(12, 10);
s.Area(10.5);
s.Area(10.2, 10.2);
Console.ReadLine();
}
}
}
}
```

**Output:**

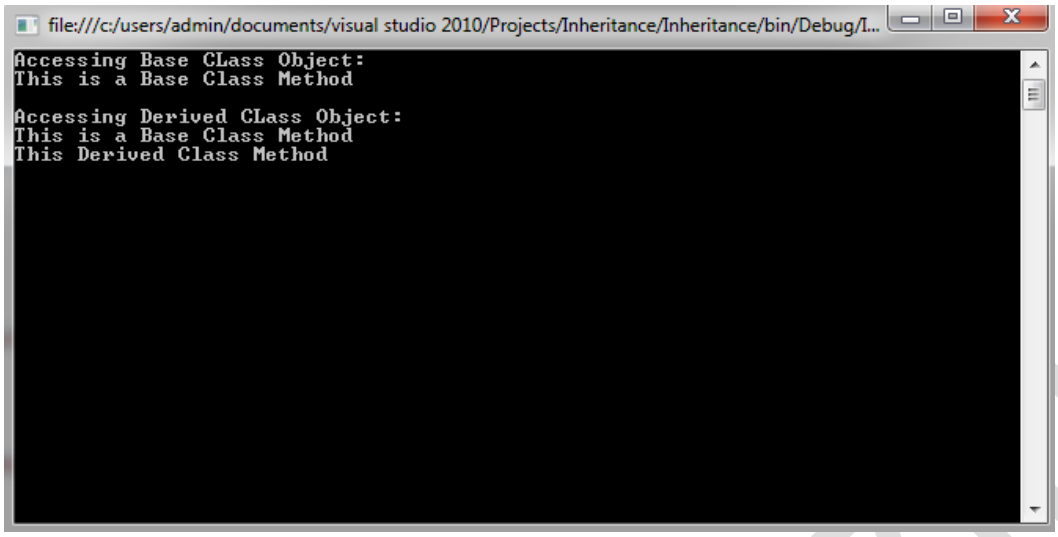


(ii) Inheritance (all types):

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

namespace Inheritance
{
public class BaseClass
{
public int DataMember;
public void BaseClassMethod()
{
Console.WriteLine("This is a Base Class Method");
}
}
public class DerivedClass : BaseClass
{
public void DerivedClassMethod()
{
Console.WriteLine("This Derived Class Method ");
}
class Program
{
```

```
static void Main(string[] args)
{
Console.WriteLine("Accessing Base CLass Object:");
BaseClass b  = new BaseClass ();
b.DataMember=1;
b.BaseClassMethod();
Console.WriteLine(" ");
Console.WriteLine("Accessing Derived CLass Object:");
DerivedClass d = new DerivedClass ();
d.DataMember=2 ;
d.BaseClassMethod();
d.DerivedClassMethod();
Console.ReadLine();
}
}
}
}
```

**Output:**

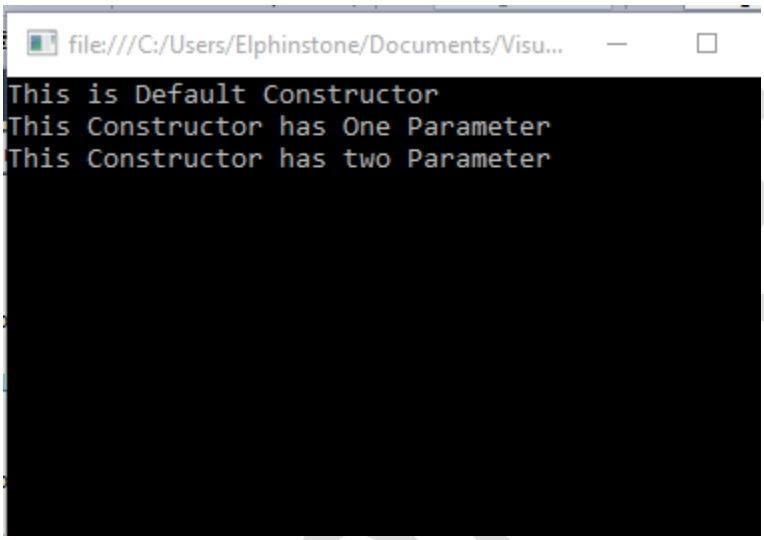

file:///c:/users/admin/documents/visual studio 2010/Projects/Inheritance/Inheritance/bin/Debug/I...
Accessing Base CLass Object:
This is a Base Class Method
Accessing Derived CLass Object:
This is a Base Class Method
This Derived Class Method

(iii) Constructor overloading :

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

namespace Constructor_Overloading
{
class Program
{
public Program()
{
Console.WriteLine("This is Default Constructor");
}

public Program(int a)
{
Console.WriteLine("This Constructor has One Parameter");
}

public Program(int b, double c)
{
Console.WriteLine("This Constructor has two Parameter");

}
static void Main(string[] args)
{
```

```
Program p = new Program();
Program p1= new Program(10);
Program p2= new Program(1,1.001);
Console.ReadLine();
}
}
}
```

**Output:**

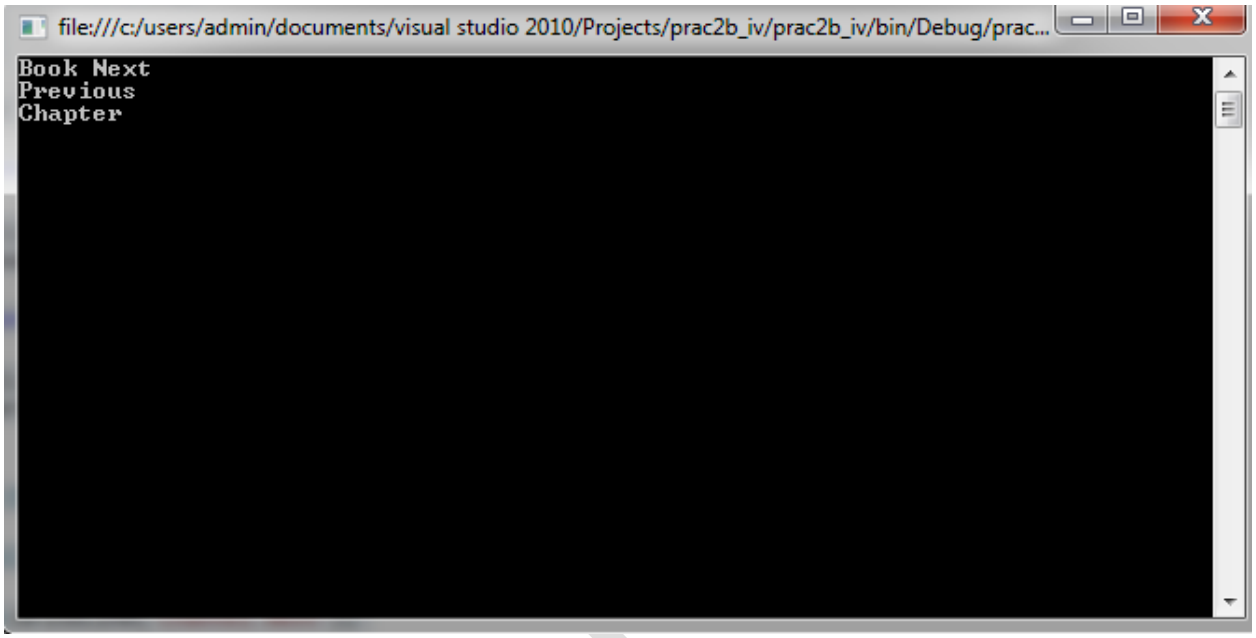


(iv) Interfaces :

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

namespace prac2b4
{
public interface Channel
{
void Next();
void Previous();
}
public interface Book
{
void Next();
void Chapter();
}
class Program : Channel,Book
{
void Channel.Next()
{
Console.WriteLine("Channel Next");
}
```

```
void Book.Next()
{
Console.WriteLine("Book Next");
}
public void Previous()
{
Console.WriteLine("Previous");
}
public void Chapter()
{
Console.WriteLine("Chapter");
}
static void Main(string[] args)
{
Program p = new Program();
((Channel)p).Next();
((Book)p).Next();
p.Previous();
p.Chapter();
Console.ReadLine();
}
}
}
```

**Output:**


file:///c:/users/admin/documents/visual studio 2010/Projects/prac2b_iv/prac2b_iv/bin/Debug/prac...
Book Next
Previous
Chapter

**Practical No: 2(C)**

**Aim:** Create simple application to demonstrate use of following concepts:

i. Using Delegates and events

ii. Exception handling

**Source Code:**

(i) Using Delegates and events:

**Default.aspx**

```
<html>
<body>
   <form id="form1" runat="server">
   <div>;
      <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label><br /><br />
            <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label><br /><br />
   </div>
   </form>
</body>
</html>
```

**Default.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{

   public delegate string dele();

   public static string display1()
   {
      string s1 = "Farzin Qureshi";
      return s1;
   }
```

```
    public static string display2()
    {
        string s2 = "Advanced Web Programming";
        return s2;
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        dele d1 = new dele(display1);
        d1();

        dele d2 = new dele(display2);
        d2();

        Label1.Text = d1();
        Label2.Text = d2();
    }
}
```

**Output:**

Farzin Qureshi

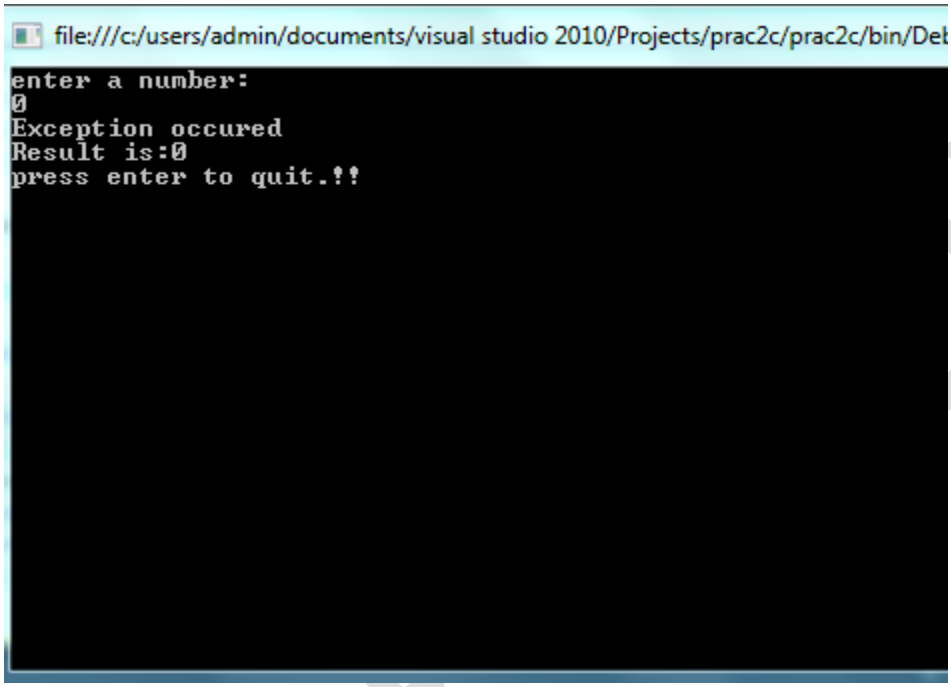
Advanced Web Programming

(ii) Exception handling

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace prac2c
{
class Program
{
static void Main(string[] args)
{
int num;
Console.WriteLine("enter a number:");
num = int.Parse(Console.ReadLine());
int div = 0;
try
{
div = 100 / num;
}
catch (DivideByZeroException)
{
Console.WriteLine("Exception occured");
}
Console.WriteLine("Result is:" +div);
Console.WriteLine("press enter to quit.!!");
Console.ReadLine();
}
```

```
}
}
```

**Output:**



**Practical 3(A)**

**Aim:** Create a simple web page with various server controls to demonstrate setting and use of their properties.(Example : AutoPostBack)

**Source Code:**

**Default.aspx:**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

FIRST NAME :<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /> <br />
LAST NAME :<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br /> <br />
CLASS :<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox><br /> <br />
ADDRESS :<asp:TextBox ID="TextBox4" runat="server"></asp:TextBox><br /> <br />
MOBILE NO :<asp:TextBox ID="TextBox5" runat="server"></asp:TextBox><br /> <br />
SUBJECT :<br /> <br />

<asp:CheckBox ID="CheckBox1" runat="server" Text="Asp.Net" />
<br /> <br />
<asp:CheckBox ID="CheckBox2" runat="server" Text="Linux" /><br /> <br />
<asp:CheckBox ID="CheckBox3" runat="server" Text="Java" /><br /> <br />
<asp:Button ID="Button1" runat="server" Text="DISPLAY"
onclick="Button1_Click" />    
        <asp:Button ID="Button2" runat="server" Text="CANCEL" onclick="Button2_Click" />
        <br /> <br />

<asp:Label ID="Label1" runat="server" Text=""></asp:Label>

</div>
</form>
```

```
</body>
</html>
```

**Default.aspx:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{

}
protected void Button1_Click(object sender, EventArgs e)
{
Label1.Text += "First Name :" + TextBox1.Text + "<br />";
Label1.Text += "Last Name : " + TextBox2.Text + "<br />";
Label1.Text += "Class : " + TextBox3.Text + "<br />";
Label1.Text += "Address : " + TextBox4.Text + "<br />";
Label1.Text += "Mobile No : " + TextBox5.Text + "<br />";
Label1.Text += "Subject :  <br />";
if (CheckBox1.Checked == true)
Label1.Text += CheckBox1.Text  + "<br />";
if (CheckBox2.Checked == true)
Label1.Text += CheckBox2.Text + "<br />";
if (CheckBox3.Checked == true)
Label1.Text += CheckBox3.Text + "<br />";

}

protected void Button2_Click(object sender, EventArgs e)
{
TextBox1.Text = "";
TextBox2.Text = "";
TextBox3.Text = "";
TextBox4.Text = "";
TextBox5.Text = "";
CheckBox1.Checked = false;
CheckBox2.Checked = false;
CheckBox3.Checked = false;
Label1.Text = "";
```

```
}
}
```

**Output :**


FIRST NAME :farzin
LAST NAME :qureshi
CLASS :TyBScIT
ADDRESS :Mumbai
MOBILE NO :8567942879
SUBJECT :
Asp.Net
Linux
Java
DISPLAY
CANCEL
First Name :farzin
Last Name : qureshi
Class : TyBScIT
Address : Mumbai
Mobile No : 8567942879
Subject :
Asp.Net
Linux

**Practical No: 4 Working with form Controls**

**Prac 4 A :**

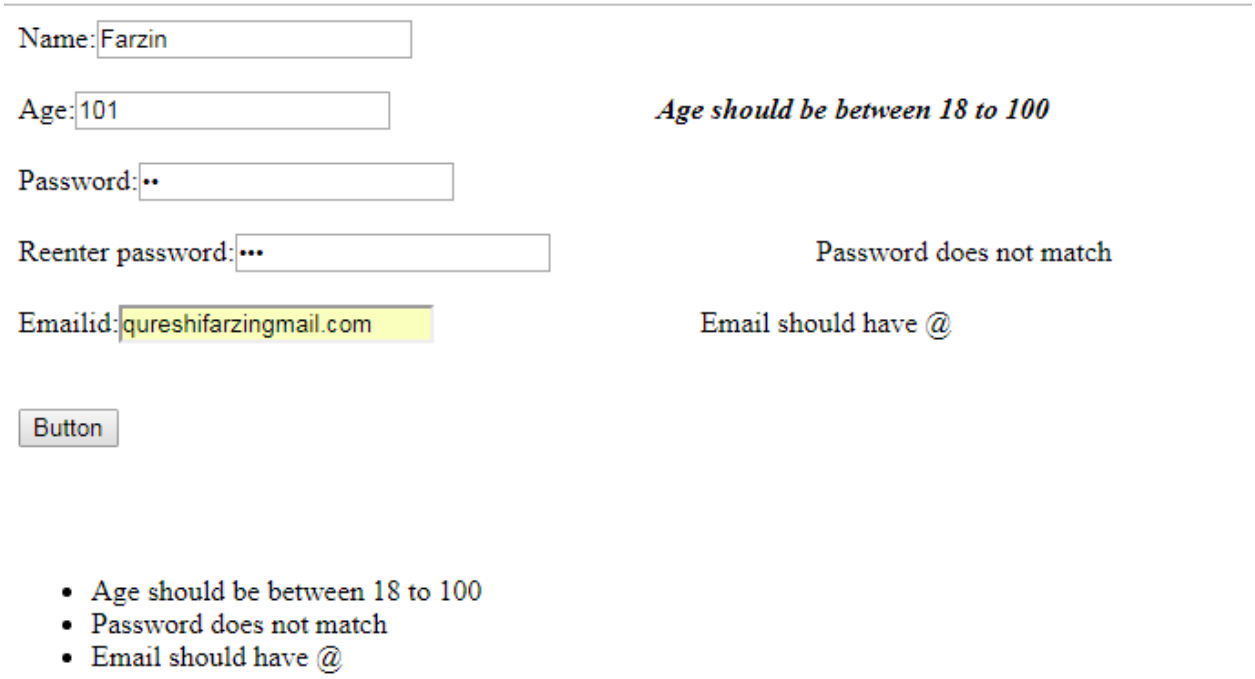
**Aim : Create a Registration form to demonstrate use of various Validation Controls**

**Deafult.aspx:**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   </head>
   <body>
   <form id="form1" runat="server">
   <div>


  Name:<asp:TextBox ID="TextBox1" runat="server" ></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ErrorMessage="
Field cannot be blank" ControlToValidate="TextBox1"></asp:RequiredFieldValidator>
<br /><br />

  Age:<asp:TextBox ID="TextBox2" runat="server" ></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ErrorMessage="Field cannot be blank"
ControlToValidate="TextBox2"></asp:RequiredFieldValidator>

<asp:RangeValidator ID="RangeValidator1" runat="server" ErrorMessage="Age should be
between 18 to 100" Type="Integer" Font-Italic="True" Font-Bold="True"
ControlToValidate="TextBox2" MaximumValue="100"
MinimumValue="18"></asp:RangeValidator>
<br /> <br />

  Password:<asp:TextBox ID="TextBox3" runat="server"
TextMode="Password"></asp:TextBox>
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
ErrorMessage="Field cannot be blank"
ControlToValidate="TextBox3"></asp:RequiredFieldValidator>

<br /><br />

 Reenter password:<asp:TextBox ID="TextBox4" runat="server"
TextMode="Password"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
ErrorMessage="Field cannot be blank"
ControlToValidate="TextBox4"></asp:RequiredFieldValidator>

<asp:CompareValidator ID="CompareValidator1" runat="server" ErrorMessage="Password
does not match" ControlToValidate="TextBox4"
ControlToCompare="TextBox3"></asp:CompareValidator><br /><br />

 Emailid:<asp:TextBox ID="TextBox5" runat="server" ></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server"
ErrorMessage="Field cannot be blank"
ControlToValidate="TextBox5"></asp:RequiredFieldValidator>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Email should have @ " ControlToValidate="TextBox5"
ValidationExpression=".+@.+"></asp:RegularExpressionValidator><br /><br /><br />

<asp:Button ID="Button1" runat="server" Text="Button" onclick="Button1_Click" /><br />

<asp:Label ID="Label1" runat="server" Text=""></asp:Label> <br /><br />

<asp:ValidationSummary ID="ValidationSummary1" runat="server" />


 </div>
</form>
</body>
</html>
```

**Output:**


Name:Farzin
Age:101
Age should be between 18 to 100
Password:••
Reenter password:•••
Password does not match
Emailid:qureshifarzingmail.com
Email should have @
Button
Age should be between 18 to 100
Password does not match
Email should have @

**Prac 4B:**

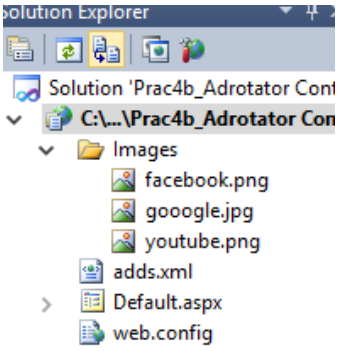
**Aim:** Create Web Form to demonstrate use of Adrotator Control.

**Source Code:**

*Default.aspx:*

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="Default.aspx.cs"Inherits="_Default"%>

<!DOCTYPEhtmlPUBLIC"-//W3C//DTD XHTML 1.0 Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
<title></title>
</head>
<body>
<formid="form1"runat="server">
<div>

<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
<br />
<asp:Timer ID="Timer1" interval="2000" runat="server">
</asp:Timer>
<br />
<asp:UpdatePanel ID="UpdatePanel1" runat="server">

<Triggers>
<asp:AsyncPostBackTrigger ControlID="Timer1"  EventName="Tick" />
</Triggers>

<ContentTemplate>
<asp:AdRotator ID="Adrotator1" runat="server" AdvertisementFile="~/adds.xml"
Height="200px" Width="200px"  />
</ContentTemplate>

</asp:UpdatePanel>

</div>
</form>
</body>
</html>
```

Create a folder named Images in your website and paste images of Google , Facebook and Youtube.




***adds.xml***

```
<?xml version="1.0" encoding="utf-8" ?>

<Advertisements>
<Ad>
<ImageUrl>~/Images/gooogle.jpg</ImageUrl>
<NavigateUrl>http://www.google.com</NavigateUrl>
<AlternateText>
Pls visit google.com
</AlternateText>
<Impressions>20</Impressions>
<Keyword>Google</Keyword>
</Ad>
<Ad>
<ImageUrl>~/Images/facebook.png</ImageUrl>
<NavigateUrl>http://www.facebook.com</NavigateUrl>
<AlternateText>Pls visit facebook.com</AlternateText>
<Impressions>20</Impressions>
<Keyword>Facebook</Keyword>
</Ad>
<Ad>
<ImageUrl>~/Images/youtube.png</ImageUrl>
<NavigateUrl>http://www.youtube.com</NavigateUrl>
<AlternateText>Pls visit youtube.com</AlternateText>
<Impressions>20</Impressions>
<Keyword>Youtube</Keyword>
</Ad>
</Advertisements>
```

**Output:**


**Practical No: 4(C)**

**Aim:** Create Web Form to demonstrate use User Controls.

**Source Code:**

*Default.aspx:*

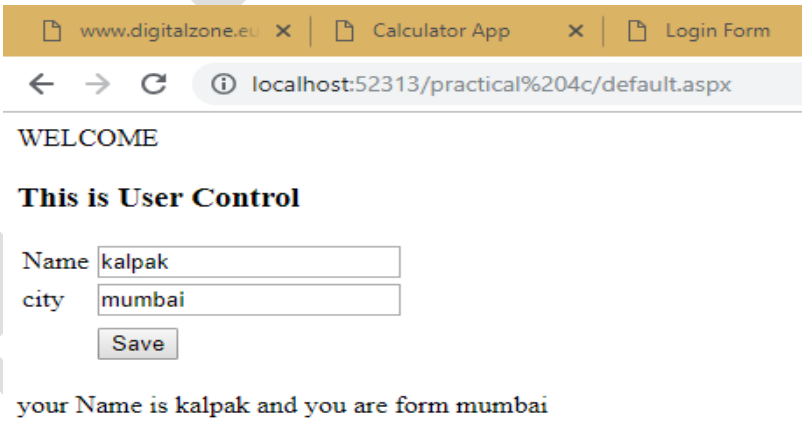
```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="Default.aspx.cs"Inherits="_Default"%>

<%@RegisterSrc="~/WebUserControl.ascx"TagPrefix="uc"TagName="Student"%>

<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
<title></title>
</head>
<body>
<form id="form1"runat="server">
<div>

<asp:LabelID="Label1" runat="server" Text="WELCOME"></asp:Label>
<uc:StudentID="studentcontrol" runat="server"/>

</div>
</form>
</body>
</html>
```

***WebUserControl.ascx:***

```
<%@ControlLanguage="C#"AutoEventWireup="true"CodeFile="WebUserControl.ascx.cs"Inherits="WebUserControl"%>
<h3>This is User Control</h3>
<table>
<tr>
<td>Name</td>
<td>
<asp:TextBoxID="txtName" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>city</td>
<td><asp:TextBoxID="txtCity" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>
<asp:ButtonID="txtsave"  runat="server"  Text="Save"  OnClick="txtsave_click"/>
</td>
</tr>
</table><br/>
<asp:LabelID="Lebel1"  runat="server"  ForeColor="Black"  Text=" "></asp:Label>
```

***WebUserControl.ascx.cs:***

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;

publicpartialclassWebUserControl : System.Web.UI.UserControl
{
protectedvoidPage_Load(object sender, EventArgs e)
{
}

protectedvoidtxtsave_click(object sender, EventArgs e)
{
Label1.Text = "your Name is " + txtName.Text + " and you are form " + txtCity.Text;
}
}
```



**Pract 5b:**

**Aim: Create a web application to demonstrate use of Master Page with applying Style and themes for page beautification.**

➔ Create a Master page

**MasterPage.master**

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
<link href="StyleSheet.css" rel="Stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>
<asp:ContentPlaceHolder ID="ContentPlaceHolder2" runat="server">
</asp:ContentPlaceHolder>

</div>
</form>
</body>
</html>
```

➔ Create a Web Form

**Default2.aspx**

```
<%@ Page Theme="Theme1" Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>



<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<asp:Label ID="Label1" runat="server" Text="AWP"></asp:Label><br /><br />
<asp:Label ID="Label2" runat="server" Text="JAVA"></asp:Label><br /><br />
<asp:Label ID="Label3" runat="server" Text=""></asp:Label>
<asp:Button ID="Button1" runat="server" Text="Button" onclick="Button1_Click" /><br /><br
/>

</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" Runat="Server">
</asp:Content>
```

**Default2.aspx.cs**

```
public partial class Default2 : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{

}
protected void Button1_Click(object sender, EventArgs e)
{
Label1.Text = "AWP";
Label2.Text = "Practicals";
Label3.Text = "Author : FQ ";
}
}
```

➔ Create an exertnal Style Sheet

**Stylesheet.css**

```
body
{
background-color:pink;
font-family:Cooper Black;
font-size:18px;

}
h1
{
text-align:center;
text-decoration:underline;
}
h2
{
text-align:right;
font-size:larger;
font-family:Arabic Transparent;
}

h3
{
text-align:left;
font-family:Chiller;
font-size:medium;
}
```

Right click on your website -> Add ASP.NET Folder -> Themes

In App_themes folder -> Right click -> add new item -> SkinFile.(give name SkinFile.skin)

**SkinFile.skin**

<%--
Default skin template. The following skins are provided as examples only.

1. Named control skin. The SkinId should be uniquely defined because
duplicate SkinId's per control type are not allowed in the same theme.

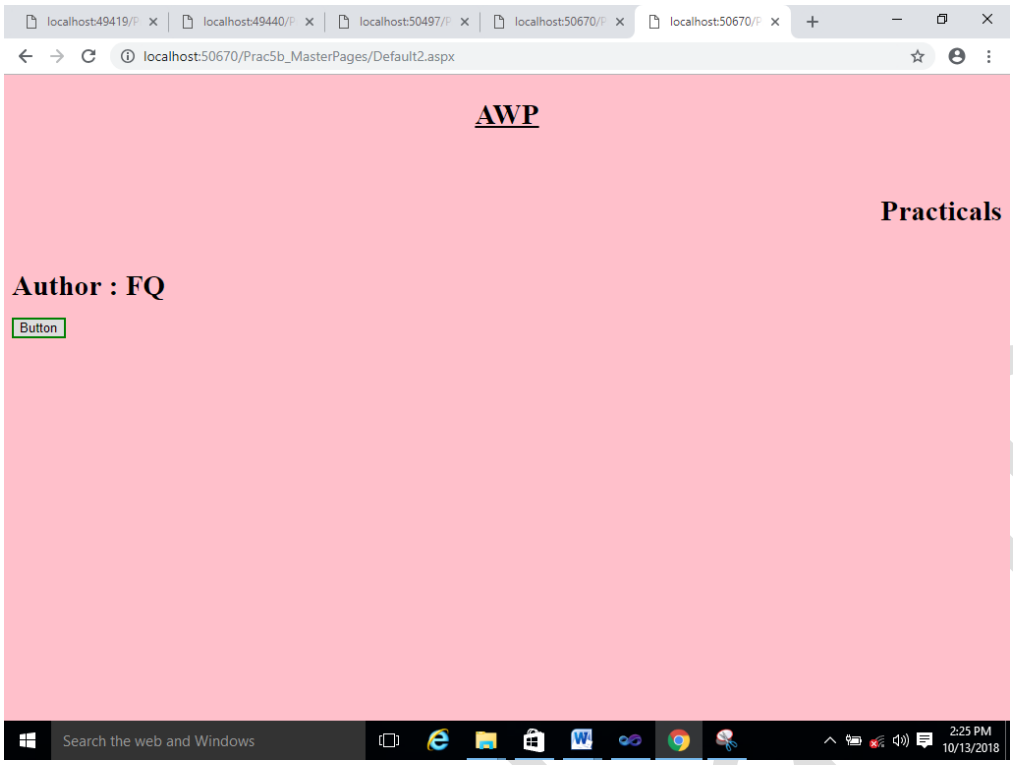
<asp:GridView runat="server" SkinId="gridviewSkin" BackColor="White" >
<AlternatingRowStyle BackColor="Blue" />
</asp:GridView>

2. Default skin. The SkinId is not defined. Only one default
control skin per control type is allowed in the same theme.

<asp:Image runat="server" ImageUrl="~/images/image1.jpg" />
--%>

<asp:Label runat="server" Text="Label" ForeColor="black" Font-size="22pt" Font-Names="Verdena"></asp:Label>

<asp:Button runat="server" Text="Button" Borderstyle="Solid" Bordercolor="green"/>


localhost:49419/P
localhost:49440/P
localhost:50497/F
localhost:50670/F
localhost:50670/F
localhost:50670/Prac5b_MasterPages/Default2.aspx
AWP
Practicals
Author : FQ
Button
Search the web and Windows
2:25 PM
10/13/2018

**Practical 5c**

**Aim : Create a web application to demonstrate various States of ASP.NET pages.**

1) **ViewState**

**Default.aspx**

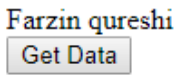
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<br />

<asp:Label ID="Label1" runat="server" Text="View State Data : "></asp:Label> <br />
<asp:Button ID="Button1" runat="server" Text="Get Data" onclick="Button1_Click" />
</div>
</form>
</body>
</html>
```

**Default.aspx.cs**

```
protected void Page_Load(object sender, EventArgs e)
{
if (!IsPostBack)
{
string str = "Farzin qureshi";
if (ViewState["name"] == null)
{
ViewState["name"] = str;
}
}
}
```

```
protected void Button1_Click(object sender, EventArgs e)
{
Label1.Text = ViewState["name"].ToString();
}
}
```

**Output:**



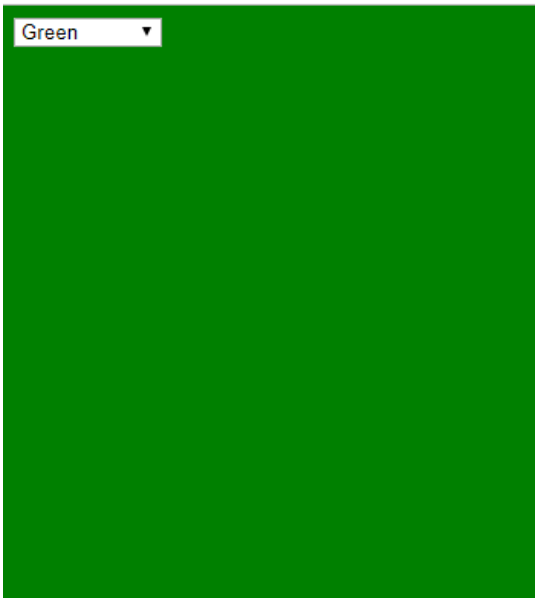

## 2) Cookies

**Cookies.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Cookies.aspx.cs" Inherits="Cookies" %>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body id="BodyTag" runat="server">
<form id="form1" runat="server">
<div>
<asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
onselectedindexchanged="DropDownList1_SelectedIndexChanged">

<asp:ListItem Value="White" Selected="True">Select Color</asp:ListItem>
<asp:ListItem Value="Red">Red</asp:ListItem>
<asp:ListItem Value="Green">Green</asp:ListItem>
<asp:ListItem Value="Blue">Blue</asp:ListItem>
<asp:ListItem>yellow</asp:ListItem>
</asp:DropDownList>


</div>
</form>
</body> </html>
```

**Cookies.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Cookies : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{

if (!IsPostBack)
{
if (Request.Cookies["BackgroundColor"] != null)
{
DropDownList1.SelectedValue = Request.Cookies["BackgroundColor"].Value;
BodyTag.Style["background-color"] = DropDownList1.SelectedValue;
}
}

}
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{

BodyTag.Style["background-color"] = DropDownList1.SelectedValue;
HttpCookie cookie = new HttpCookie("BackgroundColor");
cookie.Value = DropDownList1.SelectedValue;

cookie.Expires = DateTime.Now.AddMilliseconds(20);

//cookie.Expires = now.AddSeconds(30);
Response.Cookies.Add(cookie);

Response.SetCookie(cookie);

}
}
```




### 3) Query String:

**QueryString.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="QueryString.aspx.cs" Inherits="QueryString" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>


<br />

User ID : <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /><br />

Password : <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br /><br />


<asp:Button ID="Button1" runat="server" Text="Send Values"
onclick="Button1_Click" />
```

```
</div>
</form>
</body>
</html>
```

**QueryString.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class QueryString : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
}
protected void Button1_Click(object sender, EventArgs e)
{
Response.Redirect("Default2.aspx?UserId=" + TextBox1.Text + "&Password=" + TextBox2.Text);
}
}
```

**Default2.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
<form id="form1" runat="server">
<div>

<asp:Label ID="Label3" runat="server" Text="Query String Parameter values : "></asp:Label><br /> <br />

User ID : <asp:Label ID="Label1" runat="server" Text=""></asp:Label><br /><br />


Password : <asp:Label ID="Label2" runat="server" Text=""></asp:Label>
```

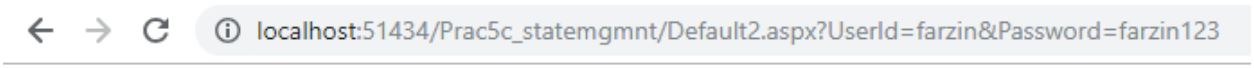
```
</div>
</form>
</body>
</html>
```

**Default2.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default2 : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
if (!IsPostBack)
{
Label1.Text = Request.QueryString["UserId"];
Label2.Text = Request.QueryString["Password"];
}
}
}
```



**Output :**


Query String Parameter values :

User ID : farzin

Password : farzin123

## 4) Session And Application State

**Global.asax**

```
<%@ Application Language="C#" %>

<script runat="server">

void Application_Start(object sender, EventArgs e)
{
// Code that runs on application startup
Application["OnlineUsers"] = 0;

}

void Application_End(object sender, EventArgs e)
{
//  Code that runs on application shutdown

}

void Application_Error(object sender, EventArgs e)
{
// Code that runs when an unhandled error occurs
}

void Session_Start(object sender, EventArgs e)
{
// Code that runs when a new session is started
Application.Lock();
Application["OnlineUsers"] = (int)Application["OnlineUsers"] + 1;
Application.UnLock();
}

void Session_End(object sender, EventArgs e)
{
// Code that runs when a session ends.
// Note: The Session_End event is raised only when the sessionstate mode
// is set to InProc in the Web.config file. If session mode is set to StateServer

Application.Lock();
Application["OnlineUsers"] = (int)Application["OnlineUsers"]-1;
Application.UnLock();
}

</script>
```

**Web.config:**

```
<configuration>

<system.web>
<sessionState mode="InProc"/>
<compilation debug="true" targetFramework="4.0" />

</system.web>

</configuration>
```

**SessionApplication.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="SessionAppication.aspx.cs" Inherits="SessionAppication" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
Visitors Count : <%=Application["OnlineUsers"].ToString() %>
</div>
</form>
</body>
</html>
```

Output:

Visitors Count : 1

## Practical 6 : Working with Database

### Practical 6a

**Aim: Create a web application bind data in a multiline textbox by querying in another textbox.**

**Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:TextBox ID="TextBox1" runat="server" TextMode="MultiLine"></asp:TextBox><br />
<asp:Button ID="Button1" runat="server" Text="Execute query"
onclick="Button1_Click" /><br />

<asp:TextBox ID="TextBox2" runat="server" TextMode="MultiLine" Rows="20"
Columns="100" ></asp:TextBox><br />
</div>
</form>
</body>
</html>
```

**Default.aspx.cs**

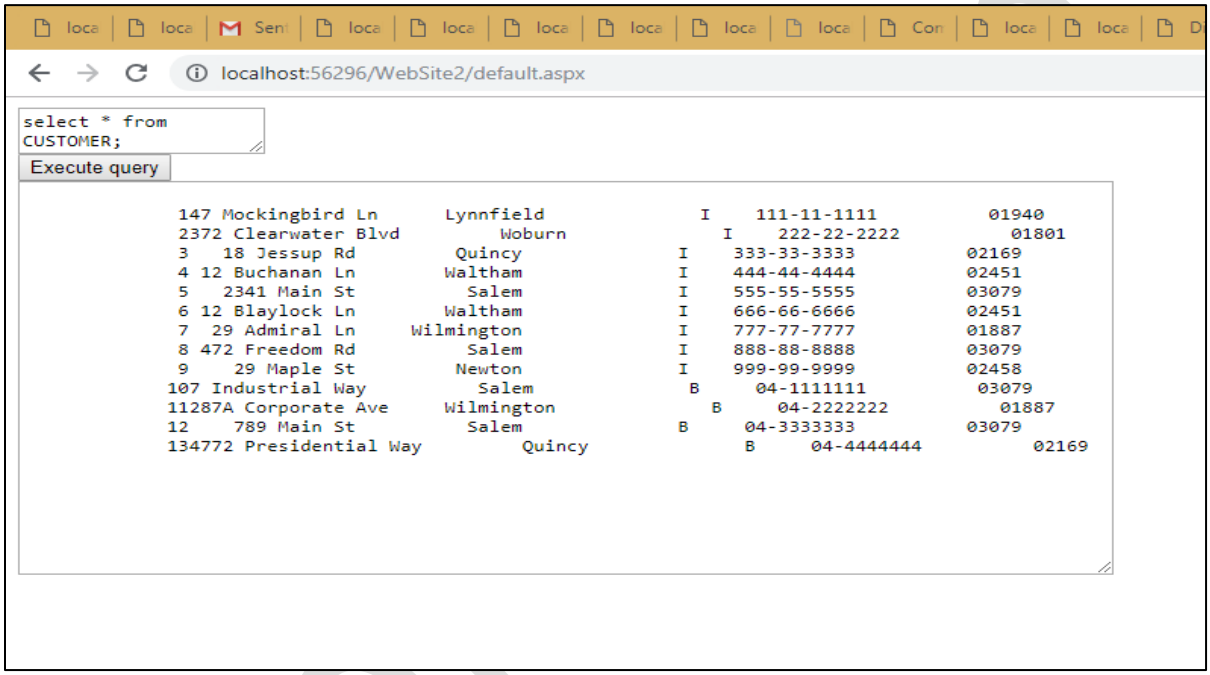
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Configuration;
using System.Data.SqlClient;
```

```
public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{

}
protected void Button1_Click(object sender, EventArgs e)
{
string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
SqlConnection con = new SqlConnection(connStr);
con.Open();

SqlCommand cmd = new SqlCommand(TextBox1.Text, con);
SqlDataReader reader = cmd.ExecuteReader();
TextBox2.Text = "";

while (reader.Read())
{
TextBox2.Text += Environment.NewLine;

for (int i = 0; i < reader.FieldCount - 1;i++ )
{
TextBox2.Text += reader[i].ToString().PadLeft(15);
}
}

reader.Close();
con.Close();


}
}
```

**Web.config**

```
<?xml version="1.0"?>

<configuration>
<system.web>
<compilation debug="false" targetFramework="4.0" />
</system.web>
```

```
<connectionStrings>
<add name="connStr" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\MITIN\Documents\Visual Studio 2010\WebSites\WebSite2\App_Data\Database.mdf;Integrated Security=True;User Instance=True"/>
</connectionStrings>

</configuration>
```



**Practical 6b**

**Aim: create an application to display records by using Database.**

**Default.apsx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Text="customer details" Font-Size="X-Large"></asp:Label><br />
<asp:Label ID="Label2" runat="server" Text=" "></asp:Label><br /><br />

<asp:Button ID="Button1" runat="server" Text="Button" onclick="Button1_Click" />
</div>
</form>
</body>
</html>
```

**Default.aspx.cs**

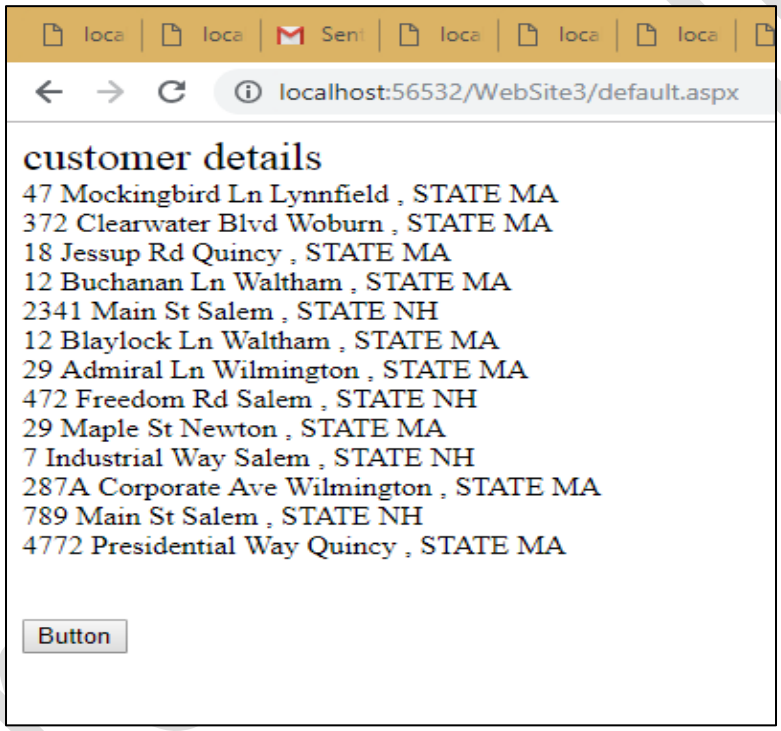
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Configuration;
using System.Data.SqlClient;

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
```

```
}
protected void Button1_Click(object sender, EventArgs e)
{
string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
SqlConnection con = new SqlConnection(connStr);
SqlCommand cmd = new SqlCommand("Select ADDRESS , CITY , STATE from CUSTOMER", con);
con.Open();


SqlDataReader reader = cmd.ExecuteReader();
while (reader.Read())
{
Label2.Text += reader["ADDRESS"].ToString() + " " + reader["city"].ToString() + " , STATE "
+reader["STATE"].ToString() + "<br>";
}
reader.Close();
con.Close();
}
}
```

**Web.config**

```
<?xml version="1.0"?>

<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->

<configuration>

<system.web>
<compilation debug="false" targetFramework="4.0" />
</system.web>

<connectionStrings>
<add name="connStr" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\MITIN\Documents\Visual Studio 2010\WebSites\WebSite3\App_Data\Database.mdf;Integrated Security=True;User Instance=True"/>
</connectionStrings>

</configuration>
```

**Output :**

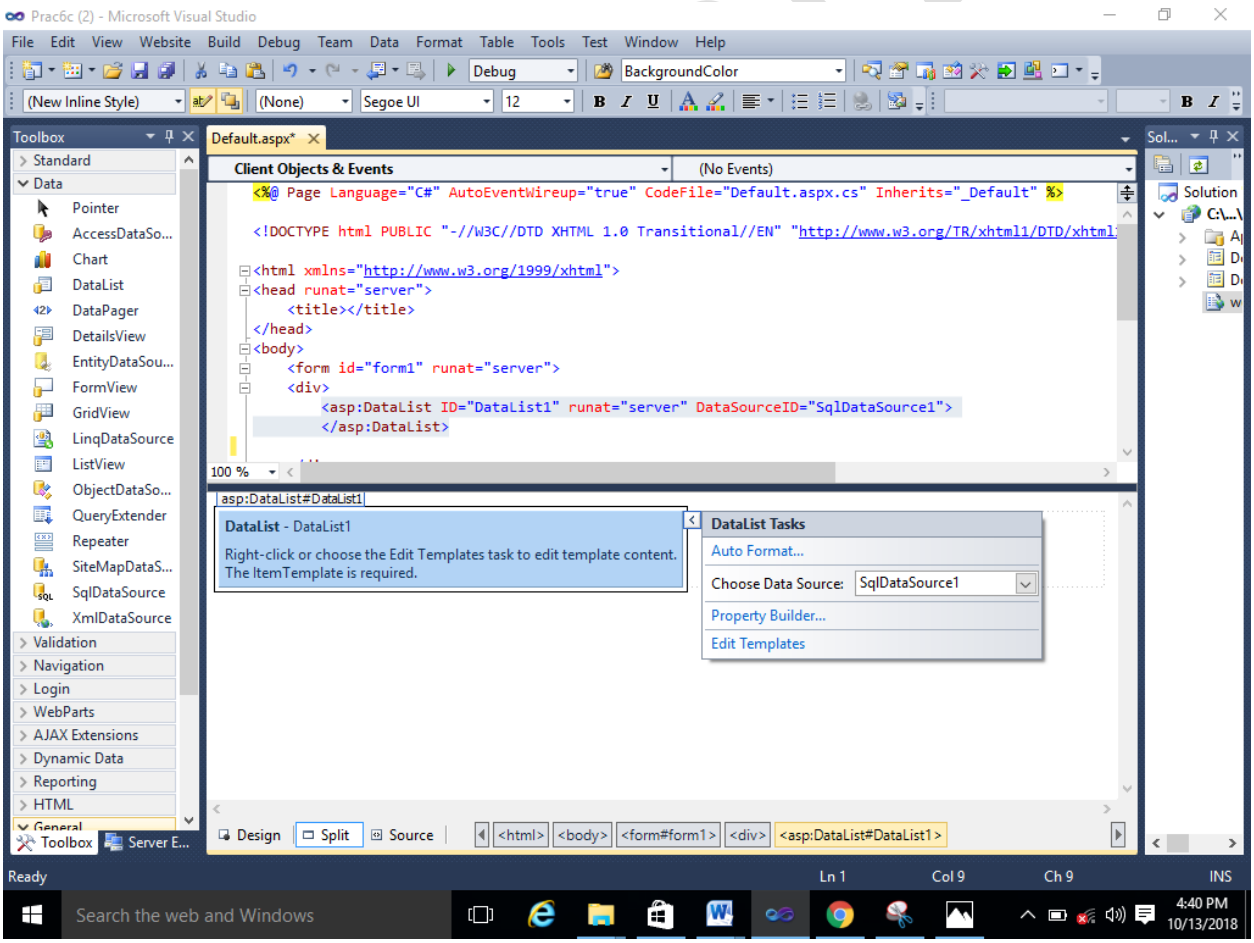

loca
loca
Sent
loca
loca
loca
localhost:56532/WebSite3/default.aspx
customer details
47 Mockingbird Ln Lynnfield , STATE MA
372 Clearwater Blvd Woburn , STATE MA
18 Jessup Rd Quincy , STATE MA
12 Buchanan Ln Waltham , STATE MA
2341 Main St Salem , STATE NH
12 Blaylock Ln Waltham , STATE MA
29 Admiral Ln Wilmington , STATE MA
472 Freedom Rd Salem , STATE NH
29 Maple St Newton , STATE MA
7 Industrial Way Salem , STATE NH
287A Corporate Ave Wilmington , STATE MA
789 Main St Salem , STATE NH
4772 Presidential Way Quincy , STATE MA
Button

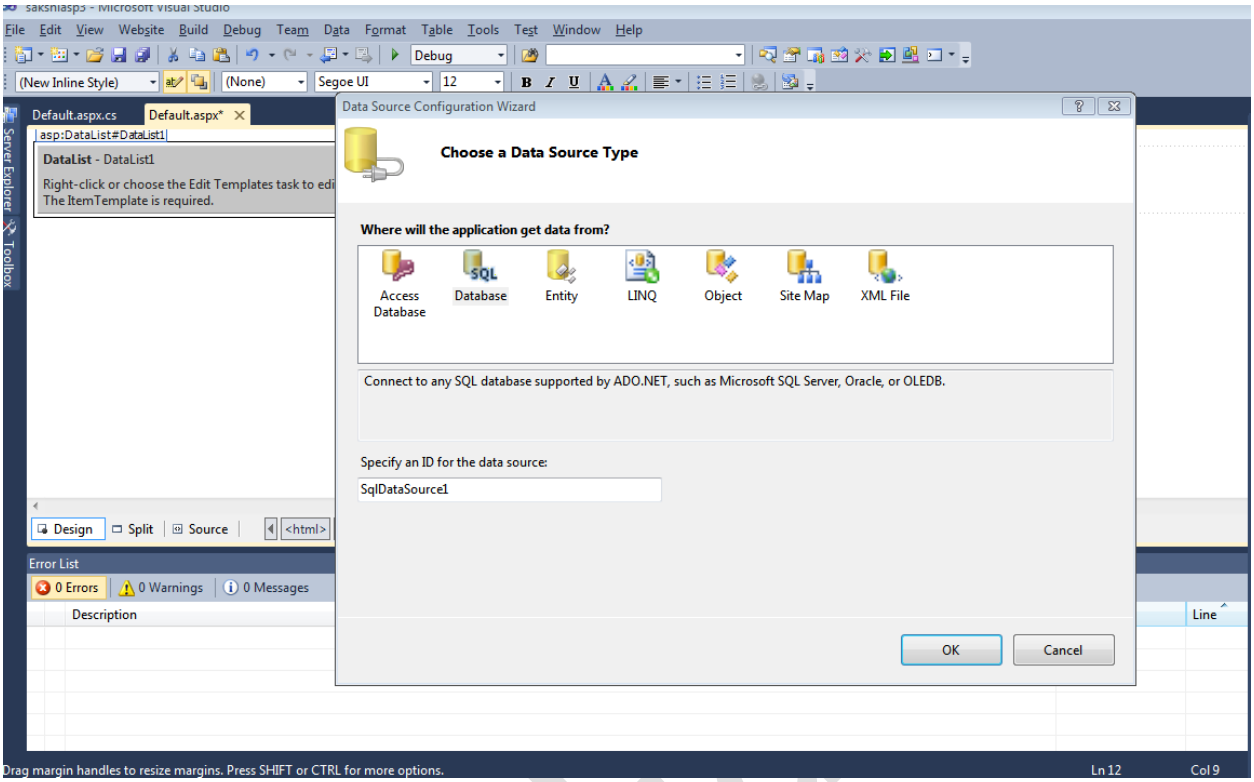
## Practical 6c

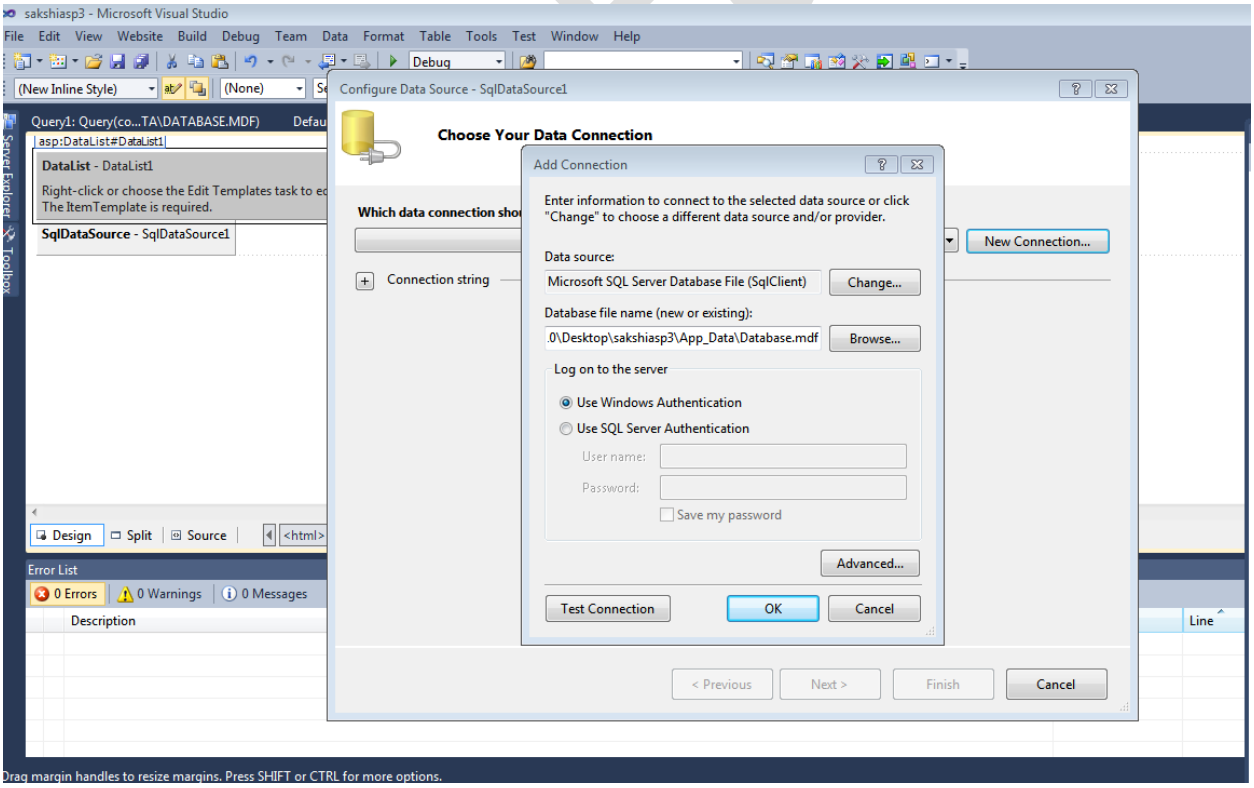
**Aim: Demonstrate the use of DataList link Control**

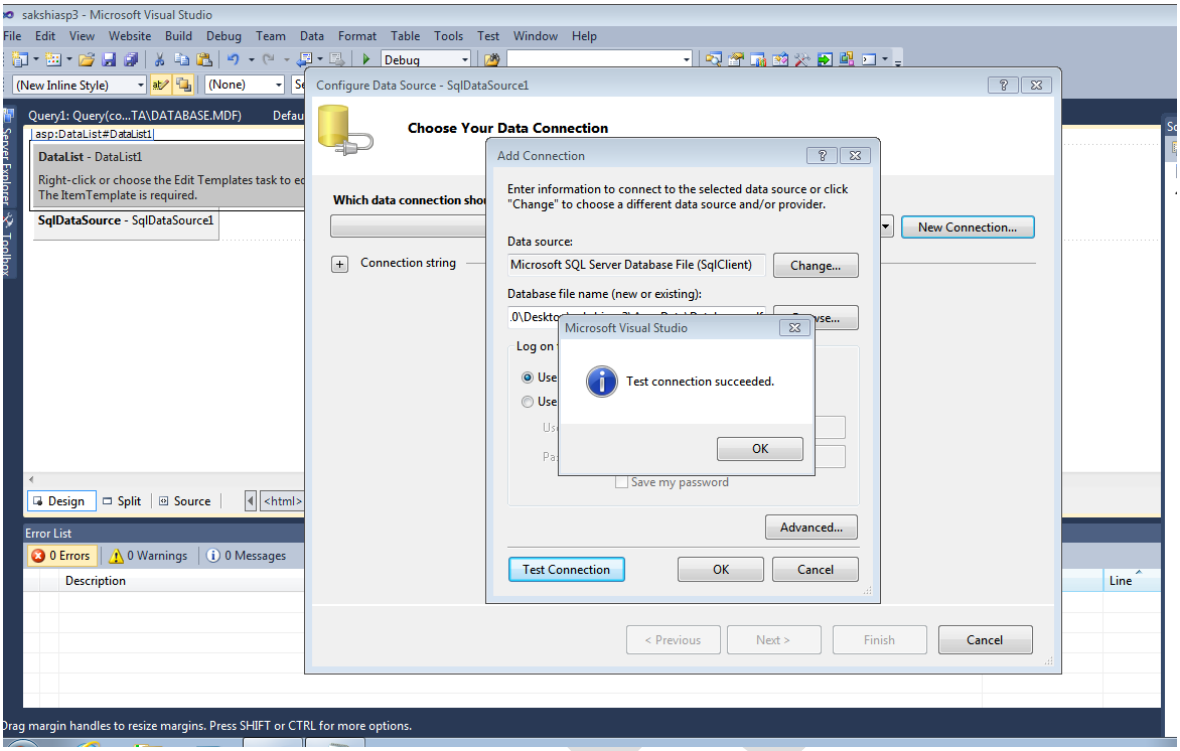
**Steps:**

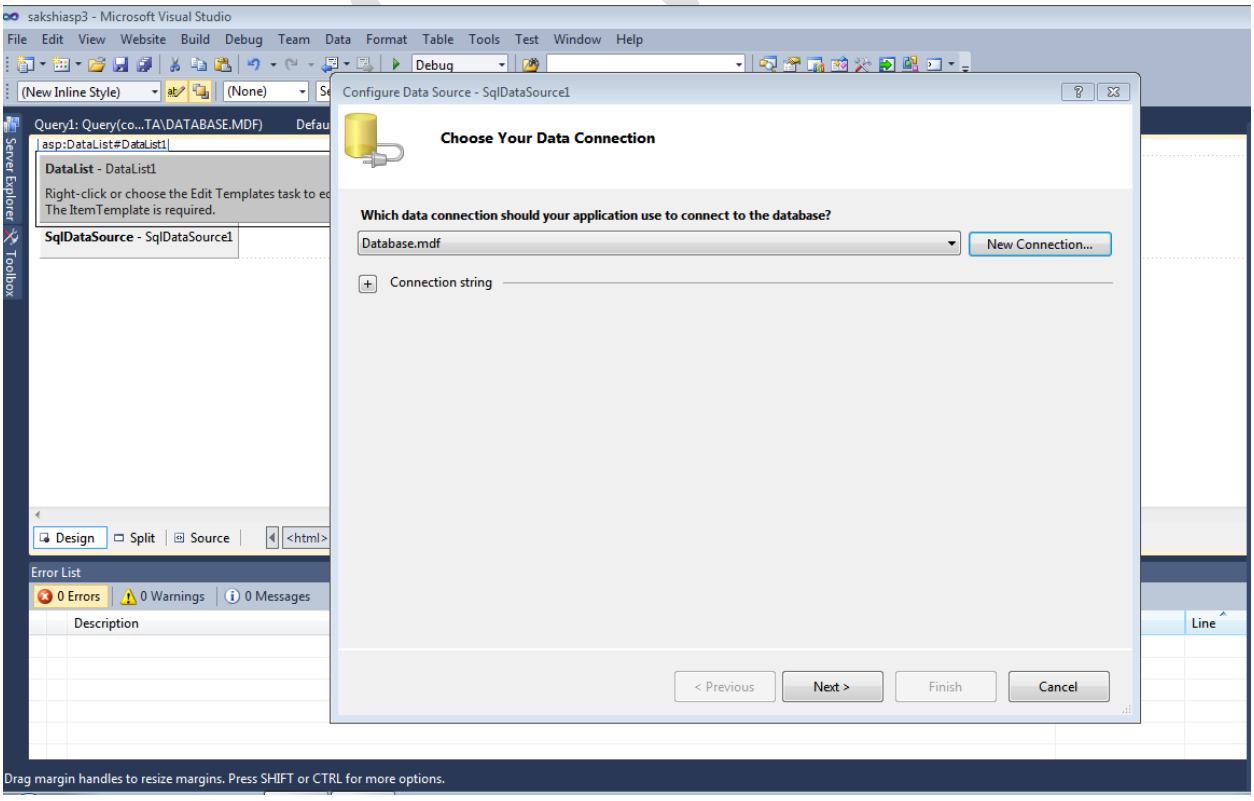
- ➔ **Do all the Database related steps done in practicals 6a and 6b.**
- ➔ **Write connection string in web.config file**
- ➔ **Add a web form**
- ➔ **Drag and drop a datalist from Data control from Toolbox**
- ➔ **Do the following steps.**

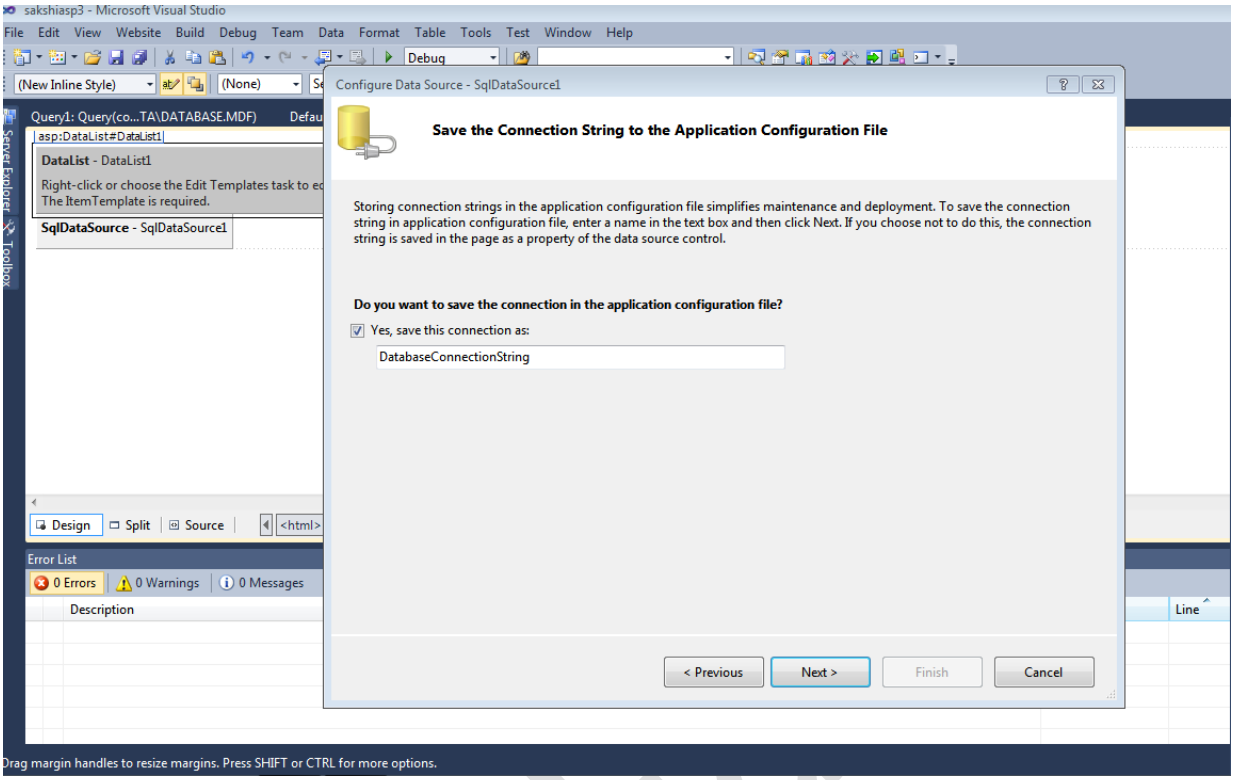


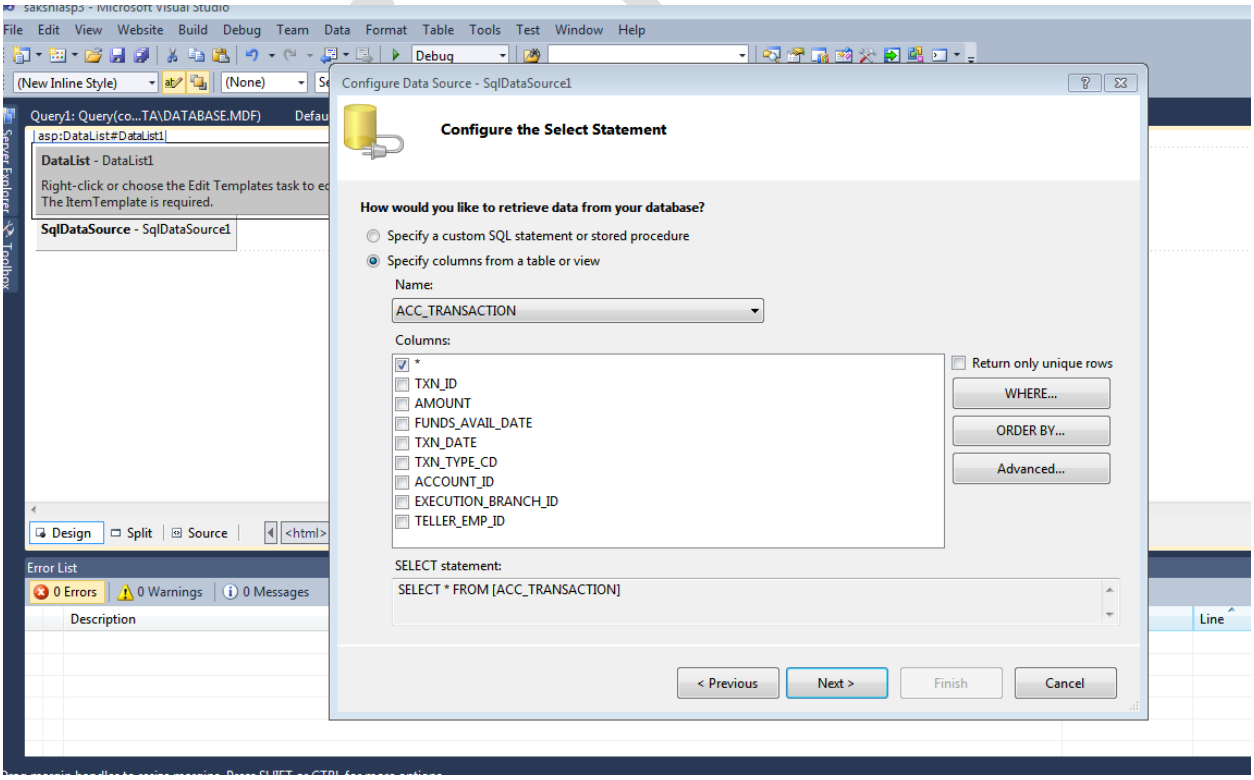

Data Source Configuration Wizard
Choose a Data Source Type
Where will the application get data from?
Access Database
Database
Entity
LINQ
Object
Site Map
XML File
Connect to any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.
Specify an ID for the data source:
SqlDataSource1
OK
Cancel

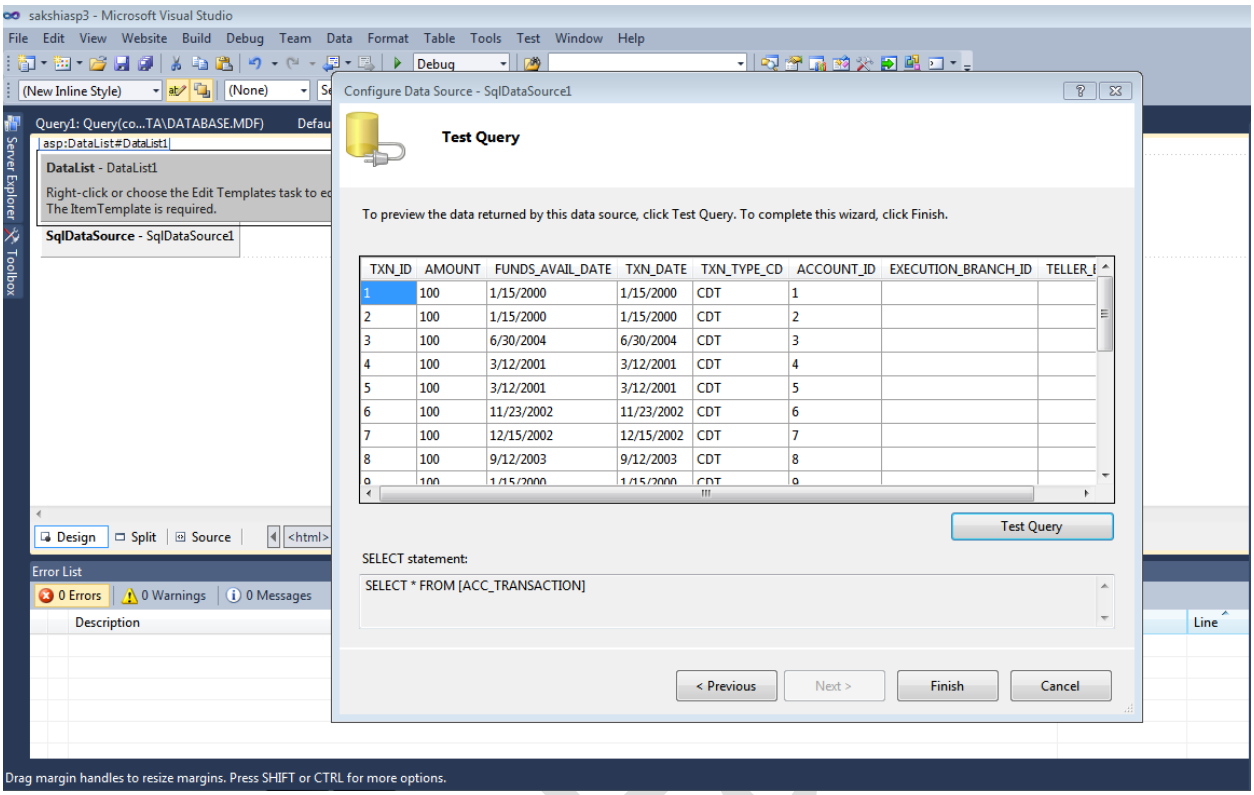

Configure Data Source - SqlDataSource1
Choose Your Data Connection
Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server Database File (SqlClient)
Change...
Database file name (new or existing):
.0\Desktop\sakshiasp3\App_Data\Database.mdf
Browse...
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
User name:
Password:
Save my password
Advanced...
Test Connection
OK
Cancel
New Connection...
Connection string
< Previous
Next >
Finish
Cancel

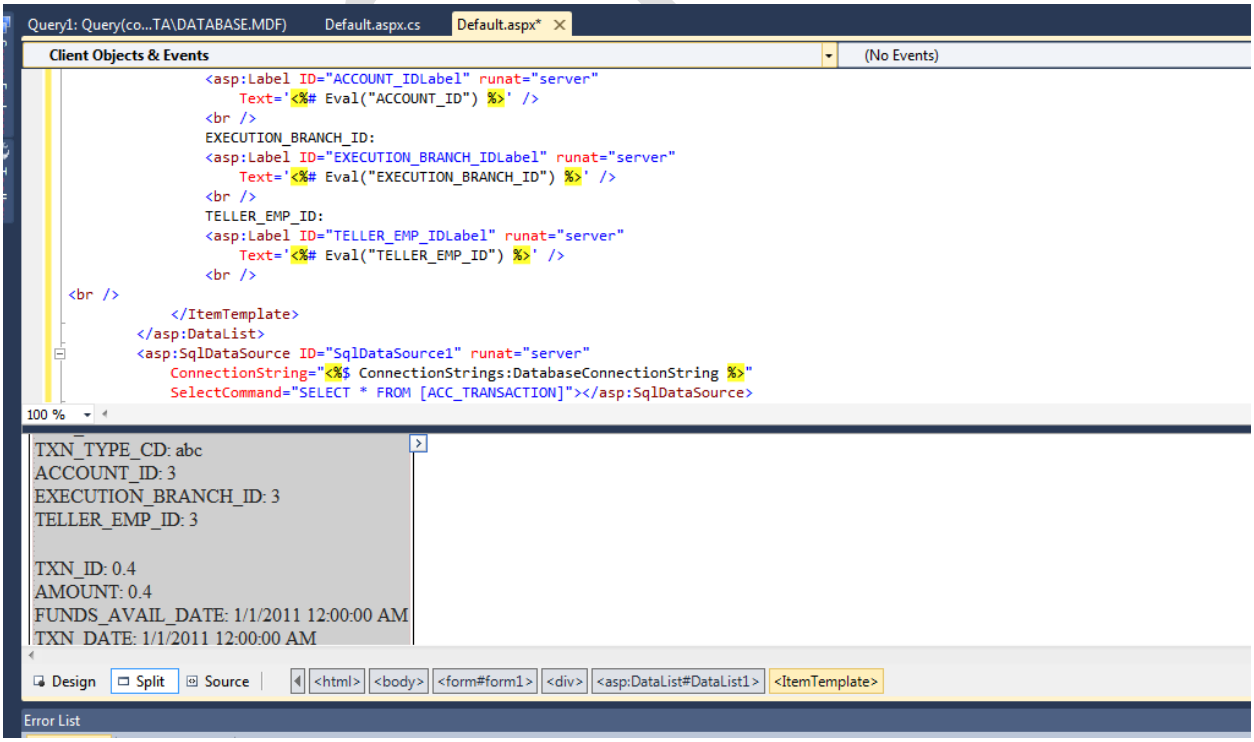

sakshiasp3 - Microsoft Visual Studio
Configure Data Source - SqlDataSource1
Choose Your Data Connection
Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server Database File (SqlClient)
Change...
Database file name (new or existing):
Microsoft Visual Studio
Test connection succeeded.
OK
Save my password
Advanced...
Test Connection
OK
Cancel
New Connection...
Connection string
< Previous
Next >
Finish
Cancel


sakshiasp3 - Microsoft Visual Studio
Configure Data Source - SqlDataSource1
Choose Your Data Connection
Which data connection should your application use to connect to the database?
Database.mdf
New Connection...
Connection string
< Previous
Next >
Finish
Cancel


sakshiasp3 - Microsoft Visual Studio
File Edit View Website Build Debug Team Data Format Table Tools Test Window Help
Configure Data Source - SqlDataSource1
Save the Connection String to the Application Configuration File
Storing connection strings in the application configuration file simplifies maintenance and deployment. To save the connection string in application configuration file, enter a name in the text box and then click Next. If you choose not to do this, the connection string is saved in the page as a property of the data source control.
Do you want to save the connection in the application configuration file?
Yes, save this connection as:
DatabaseConnectionString
< Previous
Next >
Finish
Cancel


Configure Data Source - SqlDataSource1
Configure the Select Statement
How would you like to retrieve data from your database?
Specify a custom SQL statement or stored procedure
Specify columns from a table or view
Name:
ACC_TRANSACTION
Columns:
*
TXN_ID
AMOUNT
FUNDS_AVAIL_DATE
TXN_DATE
TXN_TYPE_CD
ACCOUNT_ID
EXECUTION_BRANCH_ID
TELLER_EMP_ID
Return only unique rows
WHERE...
ORDER BY...
Advanced...
SELECT statement:
SELECT * FROM [ACC_TRANSACTION]
< Previous
Next >
Finish
Cancel


sakshiasp3 - Microsoft Visual Studio
File Edit View Website Build Debug Team Data Format Table Tools Test Window Help
Configure Data Source - SqlDataSource1
Test Query
To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.
TXN_ID AMOUNT FUNDS_AVAIL_DATE TXN_DATE TXN_TYPE_CD ACCOUNT_ID EXECUTION_BRANCH_ID TELLER_I
1 100 1/15/2000 1/15/2000 CDT 1
2 100 1/15/2000 1/15/2000 CDT 2
3 100 6/30/2004 6/30/2004 CDT 3
4 100 3/12/2001 3/12/2001 CDT 4
5 100 3/12/2001 3/12/2001 CDT 5
6 100 11/23/2002 11/23/2002 CDT 6
7 100 12/15/2002 12/15/2002 CDT 7
8 100 9/12/2003 9/12/2003 CDT 8
Test Query
SELECT statement:
SELECT * FROM [ACC_TRANSACTION]
< Previous
Next >
Finish
Cancel
Error List
0 Errors
0 Warnings
0 Messages
Description
Line
Drag margin handles to resize margins. Press SHIFT or CTRL for more options.


Query1: Query(co...TA\DATABASE.MDF)
Default.aspx.cs
Default.aspx*
Client Objects & Events
(No Events)
<asp:Label ID="ACCOUNT_IDLabel" runat="server"
Text='<%# Eval("ACCOUNT_ID") %>' />
<br />
EXECUTION_BRANCH_ID:
<asp:Label ID="EXECUTION_BRANCH_IDLabel" runat="server"
Text='<%# Eval("EXECUTION_BRANCH_ID") %>' />
<br />
TELLER_EMP_ID:
<asp:Label ID="TELLER_EMP_IDLabel" runat="server"
Text='<%# Eval("TELLER_EMP_ID") %>' />
<br />
<br />
</ItemTemplate>
</asp:DataList>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:DatabaseConnectionString %>"
SelectCommand="SELECT * FROM [ACC_TRANSACTION]"></asp:SqlDataSource>
100 %
TXN_TYPE_CD: abc
ACCOUNT_ID: 3
EXECUTION_BRANCH_ID: 3
TELLER_EMP_ID: 3
TXN_ID: 0.4
AMOUNT: 0.4
FUNDS_AVAIL_DATE: 1/1/2011 12:00:00 AM
TXN_DATE: 1/1/2011 12:00:00 AM
Design
Split
Source
<html>
<body>
<form#form1>

<asp:DataList#DataList1>
<ItemTemplate>
Error List

**Output :**


TXN_ID: 1
AMOUNT: 100
FUNDS_AVAIL_DATE: 1/15/2000 12:00:00 AM
TXN_DATE: 1/15/2000 12:00:00 AM
TXN_TYPE_CD: CDT
ACCOUNT_ID: 1
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 2
AMOUNT: 100
FUNDS_AVAIL_DATE: 1/15/2000 12:00:00 AM
TXN_DATE: 1/15/2000 12:00:00 AM
TXN_TYPE_CD: CDT
ACCOUNT_ID: 2
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 3
AMOUNT: 100
FUNDS_AVAIL_DATE: 6/30/2004 12:00:00 AM
TXN_DATE: 6/30/2004 12:00:00 AM
TXN_TYPE_CD: CDT
ACCOUNT_ID: 3
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 4
AMOUNT: 100
FUNDS_AVAIL_DATE: 3/12/2001 12:00:00 AM
TXN_DATE: 3/12/2001 12:00:00 AM
TXN_TYPE_CD: CDT
ACCOUNT_ID: 4
EXECUTION_BRANCH_ID:

**Prac 7 Working with Databases**

7 a) Aim :Create a web application to display Databinding using Dropdow

n list Control.

- ➔ **Do all the Database related steps done in practicals 6a and 6b.**
- ➔ **Write connection string in web.config file**
- ➔ **Add a web form**

Default.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:DropDownList ID="DropDownList1" runat="server">
</asp:DropDownList>
<asp:Button ID="Button1" runat="server" Text="Button"
onclick="Button1_Click1" />
<asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

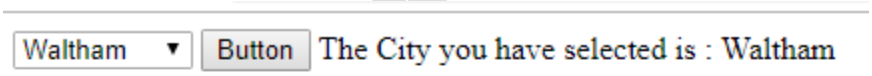
Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
if (IsPostBack == false)
{
string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
SqlConnection con = new SqlConnection(connStr);
SqlCommand cmd = new SqlCommand("Select City from Customer", con);
con.Open();
SqlDataReader reader = cmd.ExecuteReader();
DropDownList1.DataSource = reader;
DropDownList1.DataTextField = "City";
DropDownList1.DataBind();
reader.Close();
con.Close();
}

}

protected void Button1_Click1(object sender, EventArgs e)
{
Label1.Text = "The City you have selected is : " + DropDownList1.SelectedValue;

}
}
```

**Web.config**

```
<configuration>

<system.web>
<compilation debug="false" targetFramework="4.0" />
</system.web>
<connectionStrings>
<add name="connStr" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\Elphinstone\Desktop\AWP\AWP PRAC FINAL\AWP PRAC\Prac7aDataBinding\App_Data\Database.mdf;Integrated Security=True;User Instance=True" />

</connectionStrings>
</configuration>
```

Output:



Prac 7b)

Aim : Create a web application to display the title of the employee with the help of his first name.

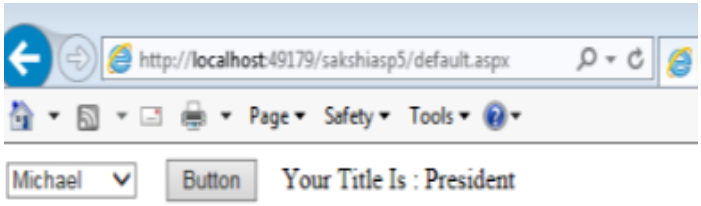
- ➔ **Do all the Database related steps done in practicals 6a and 6b.**
- ➔ **Write connection string in web.config file**
- ➔ **Add a web form**

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:DropDownList ID="DropDownList1" runat="server">
</asp:DropDownList>

<asp:Button ID="Button1" runat="server" Text="Button" onclick="Button1_Click" />
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
if (IsPostBack == false)
{
string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
SqlConnection con = new SqlConnection(connStr);
SqlCommand cmd = new SqlCommand("Select FIRST_NAME , TITLE from Employee", con);
con.Open();
SqlDataReader reader = cmd.ExecuteReader();
DropDownList1.DataSource = reader;
DropDownList1.DataTextField = "First_Name";
DropDownList1.DataValueField = "Title";
DropDownList1.DataBind();
reader.Close();
con.Close();
}
}
protected void Button1_Click(object sender, EventArgs e)
{
Label1.Text = "Your Title is : " + DropDownList1.SelectedValue;
}
}
```




**Prac 7 c**

Prac 7 c) **Create a web application to display data using Disconnected Data Access and Data binding using GridView**

- ➔ **Do all the Database related steps done in practicals 6a and 6b.**
- ➔ **Write connection string in web.config file**
- ➔ **Add a web form**

**Default.aspx:**

```
<html>
<head runat="server">
    <title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Button  ID="Button1" runat="server" Text="Show Disconnected Fetched Data"
          onclick="Button1_Click" /> <br /> <br />

<asp:GridView ID="GridView1" runat="server">
</asp:GridView>
 </div>
</form>
</body>
</html>
```

**Default.aspx.cs**

```
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

public partial class Default  : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {

   }
   protected void Button1_Click(object sender, EventArgs e)
   {
      string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
      SqlConnection con = new SqlConnection(connStr);
      SqlDataAdapter da = new SqlDataAdapter();
      DataSet ds = new DataSet();
      SqlCommand cmd = new SqlCommand("Select  *  from Customer",con);
      cmd.CommandType = CommandType.Text;
      da.SelectCommand = cmd;
      da.Fill(ds, "Product");
      GridView1.DataSource = ds.Tables[0];
      GridView1.DataBind();
   } }
```

Show Disconnected Fetched Data

| CUST_ID | ADDRESS | CITY | CUST_TYPE_CD | FED_ID | POSTAL_CODE | STATE |
|---|---|---|---|---|---|---|
| 1 | 47 Mockingbird Ln | Lynnfield | I | 111-11-1111 | 01940 | MA |
| 2 | 372 Clearwater Blvd | Woburn | I | 222-22-2222 | 01801 | MA |
| 3 | 18 Jessup Rd | Quincy | I | 333-33-3333 | 02169 | MA |
| 4 | 12 Buchanan Ln | Waltham | I | 444-44-4444 | 02451 | MA |
| 5 | 2341 Main St | Salem | I | 555-55-5555 | 03079 | NH |
| 6 | 12 Blaylock Ln | Waltham | I | 666-66-6666 | 02451 | MA |
| 7 | 29 Admiral Ln | Wilmington | I | 777-77-7777 | 01887 | MA |
| 8 | 472 Freedom Rd | Salem | I | 888-88-8888 | 03079 | NH |
| 9 | 29 Maple St | Newton | I | 999-99-9999 | 02458 | MA |
| 10 | 7 Industrial Way | Salem | B | 04-1111111 | 03079 | NH |
| 11 | 287A Corporate Ave | Wilmington | B | 04-2222222 | 01887 | MA |
| 12 | 789 Main St | Salem | B | 04-3333333 | 03079 | NH |
| 13 | 4772 Presidential Way | Quincy | B | 04-4444444 | 02169 | MA |

## Prac 10 Working with AJAX and XML

10 a)

Aim : Create a web application to demonstrate reading and writing operation in XML

**Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

<asp:Label ID="Label1" runat="server"Text="Label">
</asp:Label>  <br />  <br />
<asp:Button ID="Button1" runat="server" Text="XML Writer"
onclick="Button1_Click"  />
<br />
<br />
<br />
<asp:ListBox ID="ListBox1" runat="server" Rows="15"
style="width:200px;"></asp:ListBox>
<br />
<br />
<asp:Button ID="Button2" runat="server" Text="XML Reader" onclick="Button2_Click" />
</div>
</form>
</body>
</html>
```

**Default.aspx.cs**

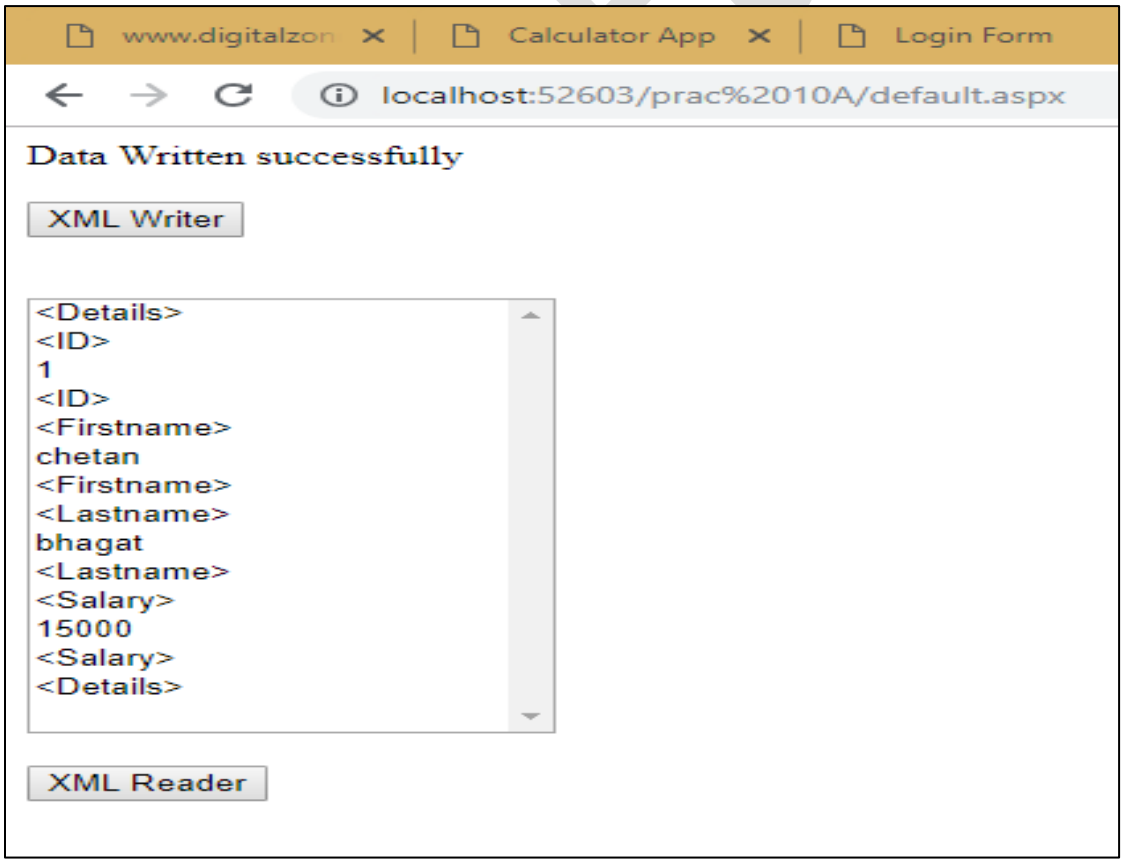
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Xml;
using System.Web.UI.WebControls;


public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{

}
protected void Button1_Click(object sender, EventArgs e)
{
XmlTextWriter writer=new XmlTextWriter("F:\\AWP\\prac 10A\\Demo.xml",null);
writer.WriteStartDocument();
writer.WriteStartElement("Details","");
writer.WriteElementString("ID","1");
writer.WriteElementString("Firstname","chetan");

writer.WriteElementString("Lastname","bhagat");
writer.WriteElementString("Salary","15000");
writer.WriteEndElement();
writer.WriteEndDocument();
writer.Close();
Label1.Text="Data Written successfully";

}
protected void Button2_Click(object sender, EventArgs e)
{
String xmlNode = "F:\\AWP\\prac 10A\\Demo.xml";
XmlReader xReader=XmlReader.Create(xmlNode);
while(xReader.Read())
{
switch(xReader.NodeType)
{
case XmlNodeType.Element:
ListBox1.Items.Add("<"+xReader.Name+">");
break;
```

```
case XmlNodeType.Text:
ListBox1.Items.Add(xReader.Value);
break;
case XmlNodeType.EndElement:
ListBox1.Items.Add("<"+xReader.Name+">");
break;

}
}
}
}
```

**Demo.xml**

```
<?xml
version="1.0"?><Details><ID>1</ID><Firstname>chetan</Firstname><Lastname>bhagat</Las
tname><Salary>15000</Salary></Details>
```

OUTPUT:



## Prac 10 b)

Aim: Create a web application to demonstrate from security and windows security with proper authentication and authorization properties.

**Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

</div>
<p>
UserName:<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
</p>
<p>
Password:<asp:TextBox ID="TextBox2" runat="server"
TextMode="Password"></asp:TextBox>
</p>
<p>
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Login" />
</p>
<p>
<asp:CheckBox ID="CheckBox1" runat="server" Text="Check here if this is not a public computer" />
</p>
<p>
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</p>
</form>
</body>
</html>
```

**Default.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
}
protected bool authenticate(String uname, String pass)
{
if (uname == "chetan")
{
if (pass == "joker123")
return true;
}
if (uname == "kalpak")
{
if (pass == "kpk123")
return true;
}
if (uname == "shashank")
{
if (pass == "svr123")
return true;
}
return false;
}

protected void Button1_Click(object sender, EventArgs e)
{
if (authenticate(TextBox1.Text, TextBox2.Text))
{
FormsAuthentication.RedirectFromLoginPage(TextBox1.Text, CheckBox1.Checked);
Session["Username"] = TextBox1.Text;
Response.Redirect("Default2.aspx");
}
else
{
Response.Write("Invalid username or password");
```

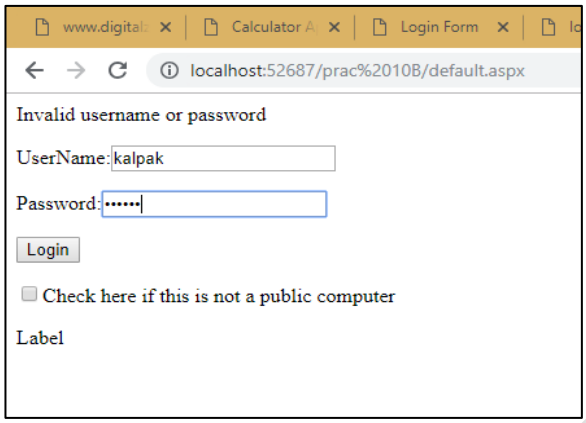
```
}
}
}
```

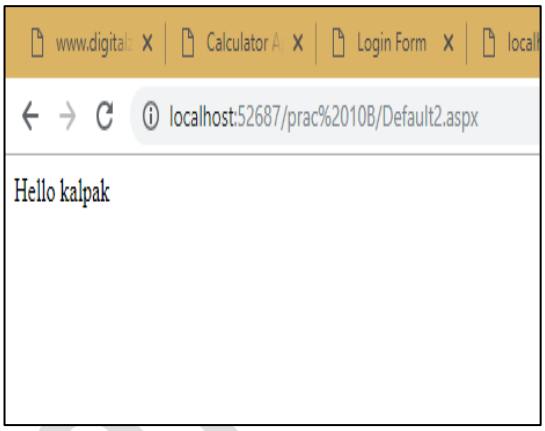
**Default2.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

**Default2.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default2 : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
if (Session["Username"] != null)
{
Label1.Text ="Hello " + Session["Username"].ToString();
}
}}
```


www.digital
Calculator A
Login Form
localhost:52687/prac%2010B/default.aspx
Invalid username or password
UserName: kalpak
Password:
Login
Check here if this is not a public computer
Label

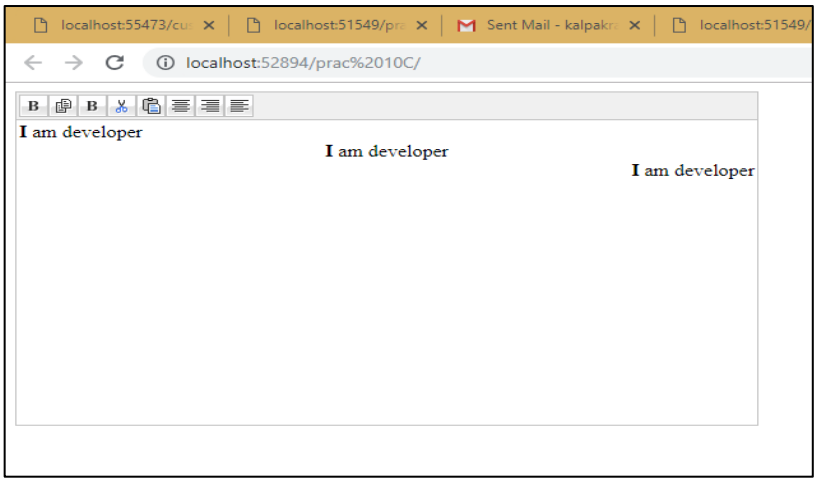

www.digital
Calculator A
Login Form
localhost:52687/prac%2010B/Default2.aspx
Hello kalpak

## Practical 10c

Aim: create a web application to demonstrate use of various Ajax Controls.

**Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="ajaxToolkit" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>

<asp:TextBox ID="TextBox1" runat="server" rows="20" Columns="80" TextMode="MultiLine"></asp:TextBox>
<ajaxToolkit:HtmlEditorExtender ID="HtmlEditorExtender1" runat="server" EnableSanitization="False" TargetControlID="TextBox1">
<Toolbar>
<ajaxToolkit:Bold />
<ajaxToolkit:Copy />
<ajaxToolKit:Bold />
<ajaxToolkit:Cut />
<ajaxToolkit:Paste />
<ajaxToolkit:JustifyCenter />
<ajaxToolKit:JustifyRight />
<ajaxToolKit:JustifyLeft />
</Toolbar>
</ajaxToolkit:HtmlEditorExtender>

</div>
</form>
</body>
</html>
```



# Pract 11

**Aim: Program to create and use DLL.**

Steps:

In Visual Studio -> file -> new -> Project -> select Class Library

**ClassLibrary1.cs**

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

namespace ClassLibrary1
{
public class Class1
{
public string UpperConvert(string text)
{
return text.ToUpper();
}
public string LowerConvert(string text)
{
return text.ToLower();
}
}
}
```

**Save your and click on Build Solution in Build Menu.**

**This code will generate .dll for your program (ClassLibrary1.dll)**

**Then click on file close solution**
**Create a new website and write the following codes in a web form**

**Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>

<body>
<form id="form1" runat="server">
<div>
<asp:Button ID="Button1" runat="server" Text="UPPER" style="height:26px" onclick="Button1_Click" />
<asp:Button ID="Button2" runat="server" Text="LOWER" onclick="Button2_Click" />
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

</div>
</form>
</body>
</html>
```

**Right Click on your website -> Add Reference Then browse to the ClassLibrary1.dll file you created.**

**Default.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ClassLibrary1;

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
```

```
}
protected void Button1_Click(object sender, EventArgs e)
{
Class1 t = new Class1();
TextBox1.Text = t.UpperConvert(TextBox1.Text);

}
protected void Button2_Click(object sender, EventArgs e)
{
Class1 t = new Class1();
TextBox1.Text = t.LowerConvert(TextBox1.Text);
}
}
```

Output :