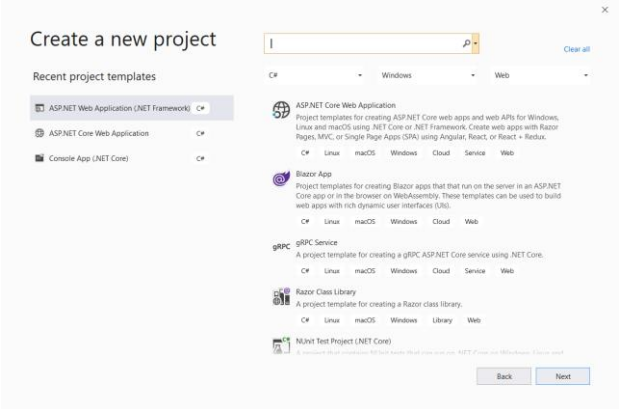
Date: 19/08/2020

Practical No . 1

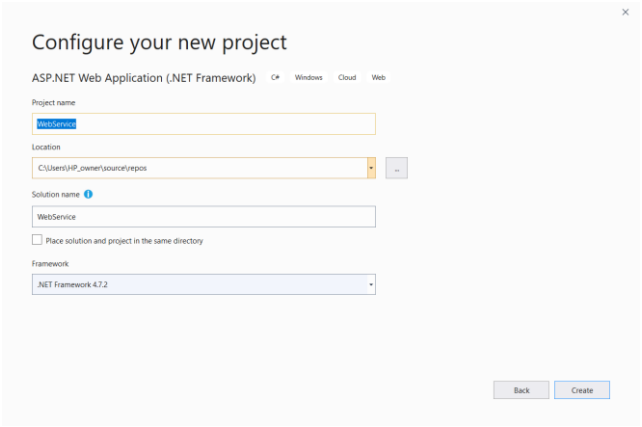
Aim : *Write a program to implement to create a simple web service that converts the temperature from Farhrenheit to Celcius and vice a versa.*

Steps for **Creating Web Service :**

1 . Create New Project -> Select ASP.NET Web Application(.NET FrameWork) and click NEXT.



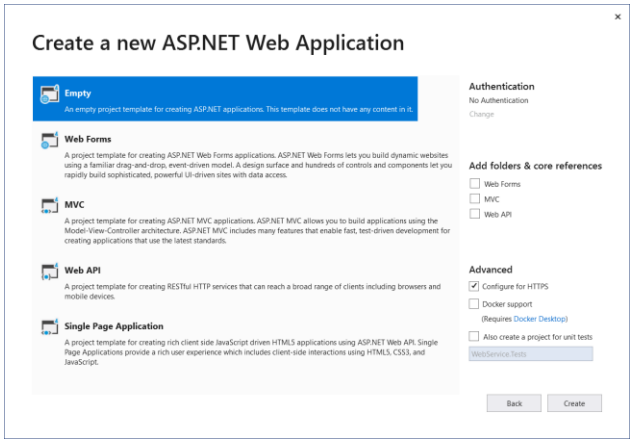
2 . Now,the window will open where you will be ask to give the project a name and also asked to select .NET Framework. After this process done click on CREATE button.



3 . Now you will ask to select ASP.NET Web Application Templates.Select “EMPTY” template.

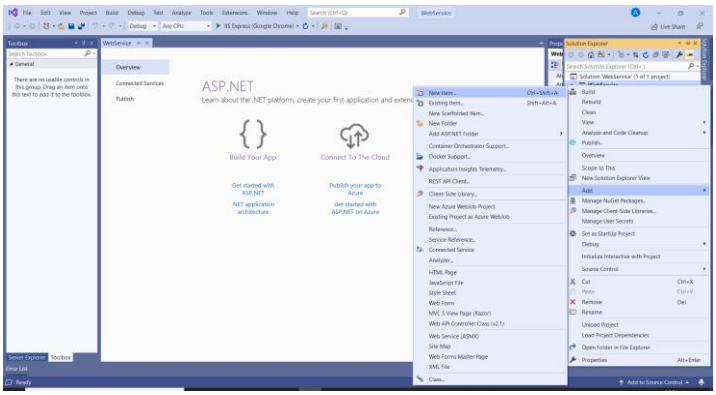
And also unchecked the “Configure for HTTPS” checkbox which at the bottom right of the window.And after it is done simply click on CREATE.

(NOTE -Due to the Configure for Https system doesn’t allow our web service to run on browser properly)



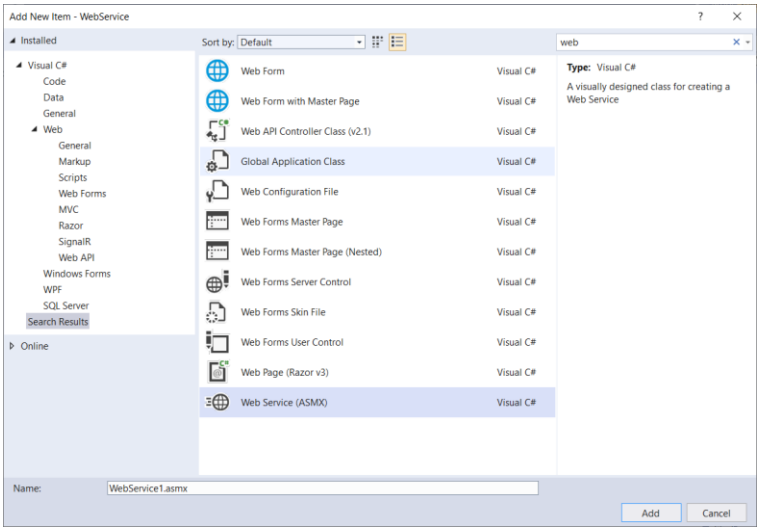
4 . Now our ASP.NET Web Application Empty project is created. And for creating a Web Service . We must go to “Solution Explorer” tab . Then Right click on our project .

Go to “Add” option and then select “New Item”.(Add --> New item).

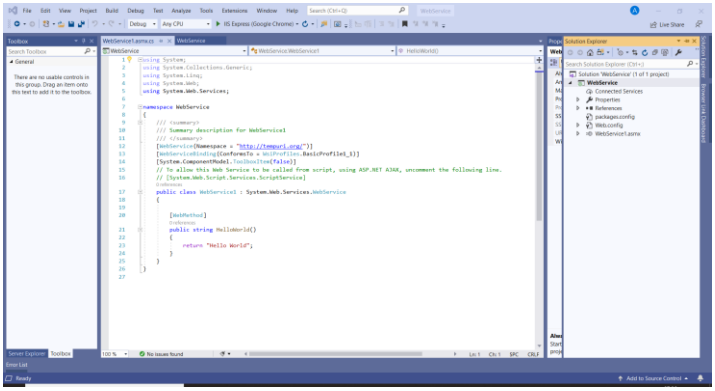


5.After selecting New Item a new window will pop up. Search for Web Service (You can use search bar to search WebService).

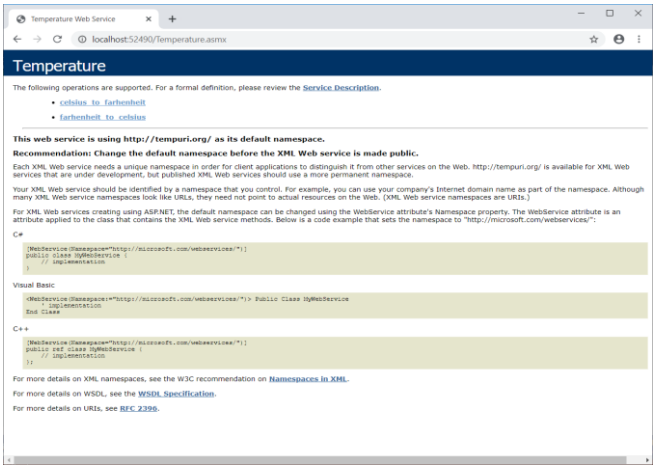
After that Select WebService(with ASMX extension) give it some name and add it to your project .



6.Now a WebService.asmx.cs file will be added to your project.It will have by default a Hello World Web Service program in it .



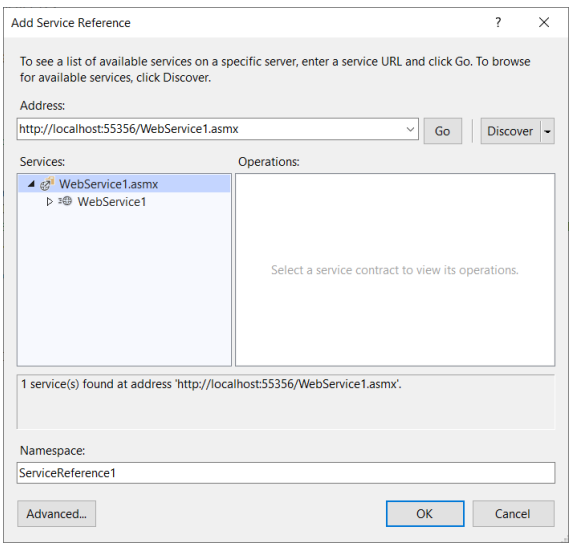
7.Now add your own WebService program in it . And run the webservice (asmx.cs extension file) file on the browser.



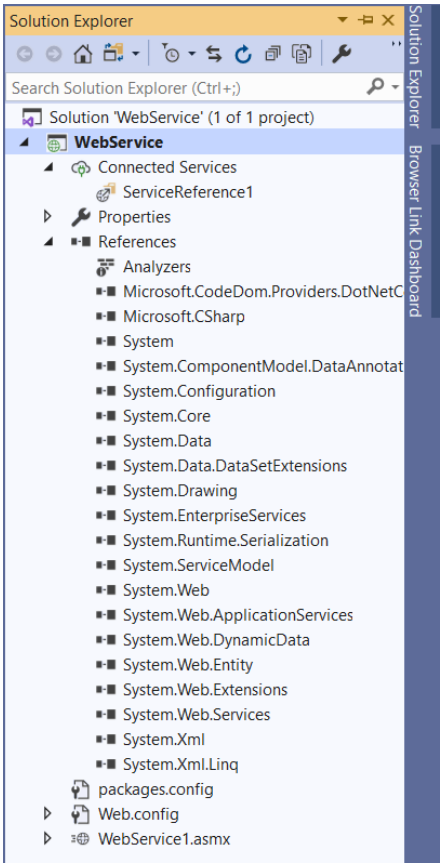
8.Now after running WebService properly . We should add a service reference.For doing so,we should go to project right click on it.

Then go to Add → Service Reference.

A new window will pop up asking for reference link and Namespace. For reference link click on “Discover” Button . And click on OK.



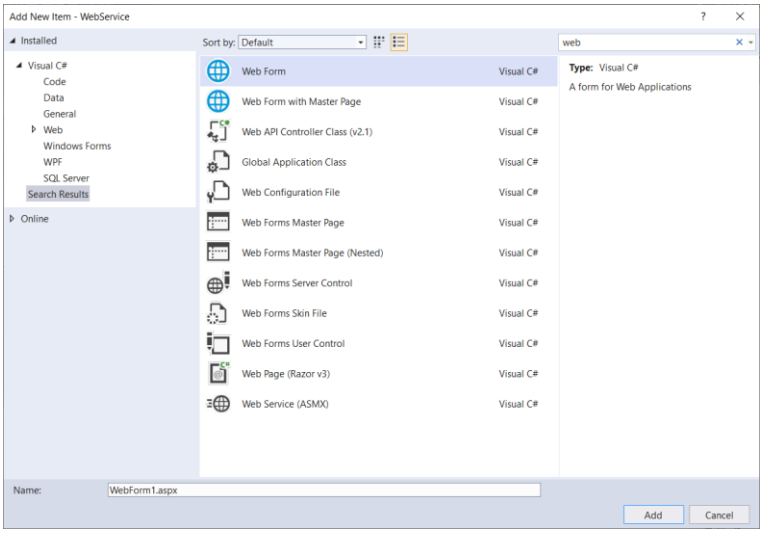
9.Now the service reference is added to our project.



10.Now we required a client to use our WebService.For that again right click on project .

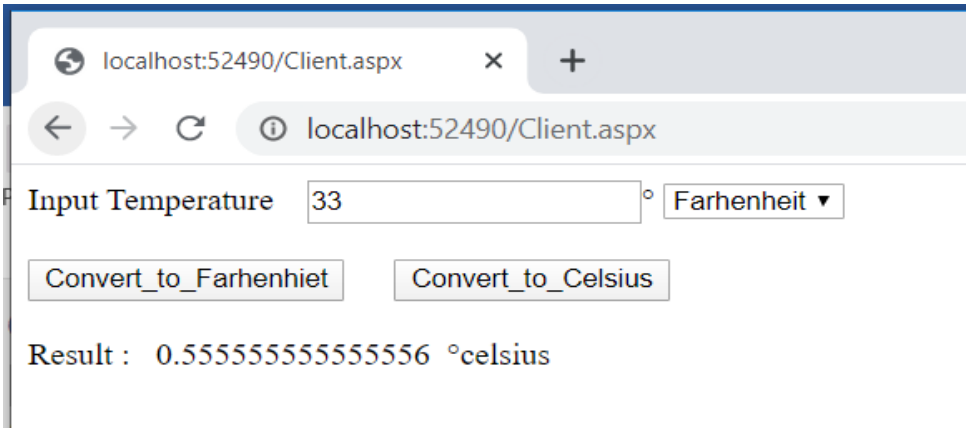
Then go to Add → New Item → Web Form.

Give the WebForm a appropriate name and add it to our project .



11.After adding web form to our project add the widgets like label ,button etc in our web form and design it properly . Then double click on button and a webform.aspx.cs file will get opened .Now program and call the WebService using Service reference object in our webform.aspx.cs(client).

And then run the Webform.aspx and our webservice will be ready to use by our client.



Code

Temperature.asmx.cs file :

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

namespace WS_TempConv
{
    /// <summary>
    /// Summary description for Temperature
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class Temperature : System.Web.Services.WebService
    {

        [WebMethod]
        public double celsius_to_farhenheit(double celsius)
        {
            return ((celsius * 9 / 5) + 32);
        }
        [WebMethod]
        public double farhenheit_to_celsius(double farhenheit)
        {
            return ((farhenheit - 32) * 5 / 9);

        }
    }
}
```

Client.aspx file :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Client.aspx.cs"
Inherits="WS_TempConv.Client" %>

<!DOCTYPE html>

<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="Input Temperature"></asp:Label>
              
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>&deg;
            <asp:DropDownList ID="DropDownList1" runat="server">

                <asp:ListItem>Celsius</asp:ListItem>
                 <asp:ListItem>Farhenheit</asp:ListItem>

            </asp:DropDownList>
            <br/><br/>

            <asp:Button ID="Button1" runat="server" Text="Convert_to_Farhenhiet"
OnClick="Button1_Click"/>
                
            <asp:Button ID="Button2" runat="server" Text="Convert_to_Celsius"
OnClick="Button2_Click" />
            <br/><br/>

            <asp:Label ID="Label3" runat="server" Text="Result :"></asp:Label>
             
            <asp:Label ID="Label2" runat="server" Text=""></asp:Label>
             <asp:Label ID="Label4" runat="server" Text=""></asp:Label>

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

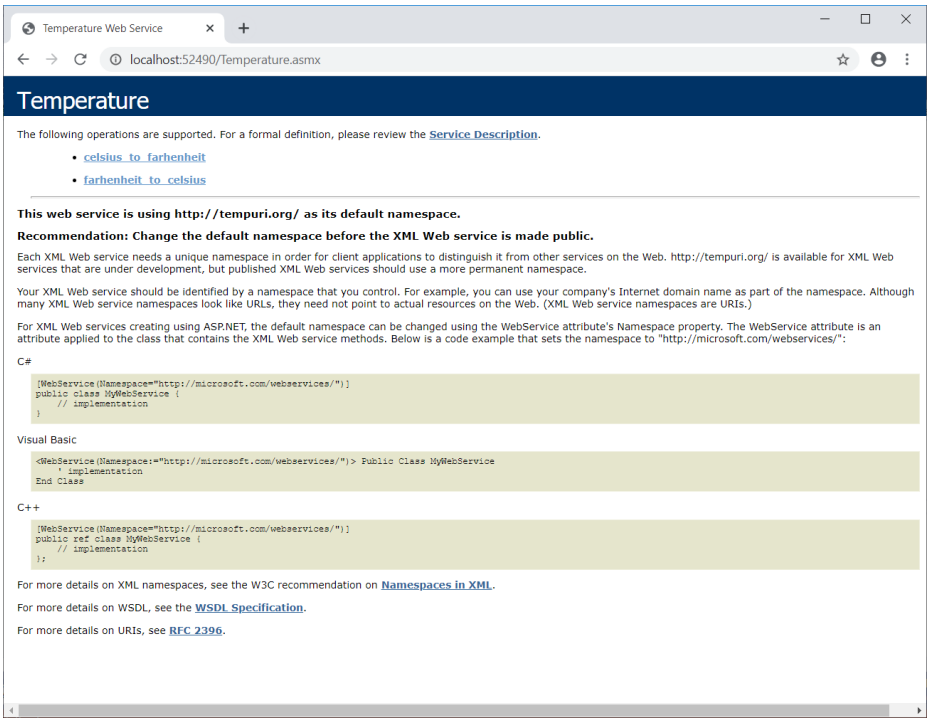
Client.aspx.cs file :

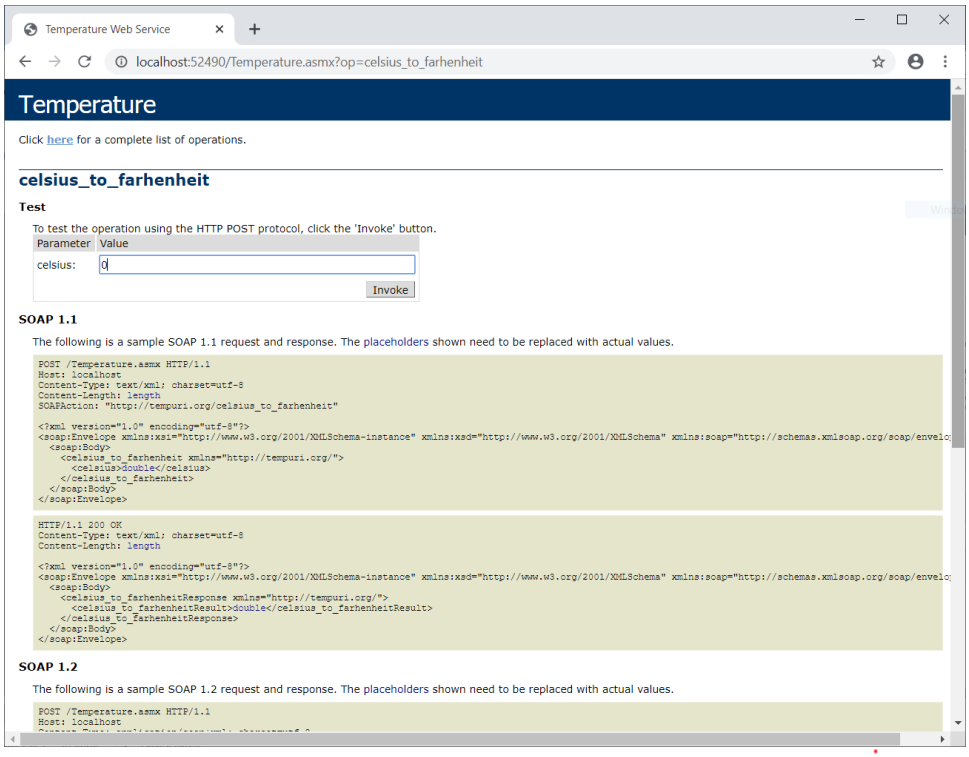
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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


      }

      protected void Button1_Click(object sender, EventArgs e)
      {

         double result;
         try
         {
            ServiceReference1.TemperatureSoap client = new
ServiceReference1.TemperatureSoapClient();

            result = client.celsius_to_farhenheit(Convert.ToDouble(TextBox1.Text));

            if (DropDownList1.SelectedValue.Equals("Celsius"))
            {
               Label2.Text = result.ToString();
               Label4.Text = "&deg;Farhenheit";
            }
            if (DropDownList1.SelectedValue.Equals("Farhenheit"))
            {
               Label2.Text = "Already in Farhenheit";
               Label4.Text = "";
            }
         }
         catch (System.FormatException)
         {
            Label2.Text = "Invalid Inputs";
            Label4.Text = "";
         }

      }
```

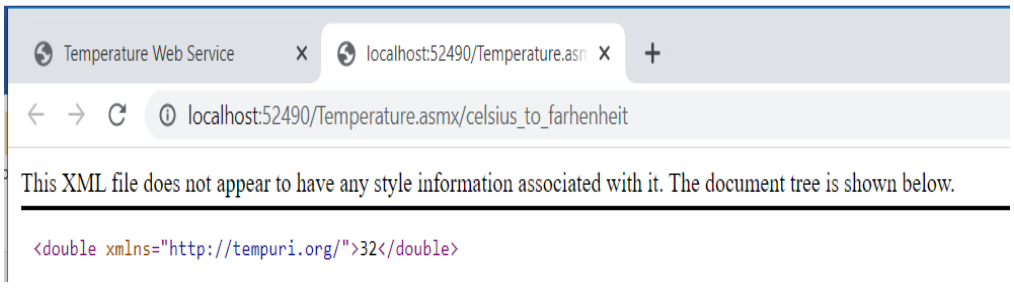
```
    protected void Button2_Click(object sender, EventArgs e)
    {
        double result;
        try
        {
            ServiceReference1.TemperatureSoapClient client = new
ServiceReference1.TemperatureSoapClient();
            result = client.farhenheit_to_celsius((Convert.ToDouble(TextBox1.Text)));
            if (DropDownList1.SelectedValue.Equals("Celsius"))
            {
                Label2.Text = "Already in Celsius";
                Label4.Text = "";
            }
            if (DropDownList1.SelectedValue.Equals("Farhenheit"))
            {
                Label2.Text = result.ToString();
                Label4.Text = "&deg;celsius";
            }
        }
        catch (System.FormatException)
        {
            Label2.Text = "Invalid Inputs";
            Label4.Text = "";

        }
    }
  }
}
```

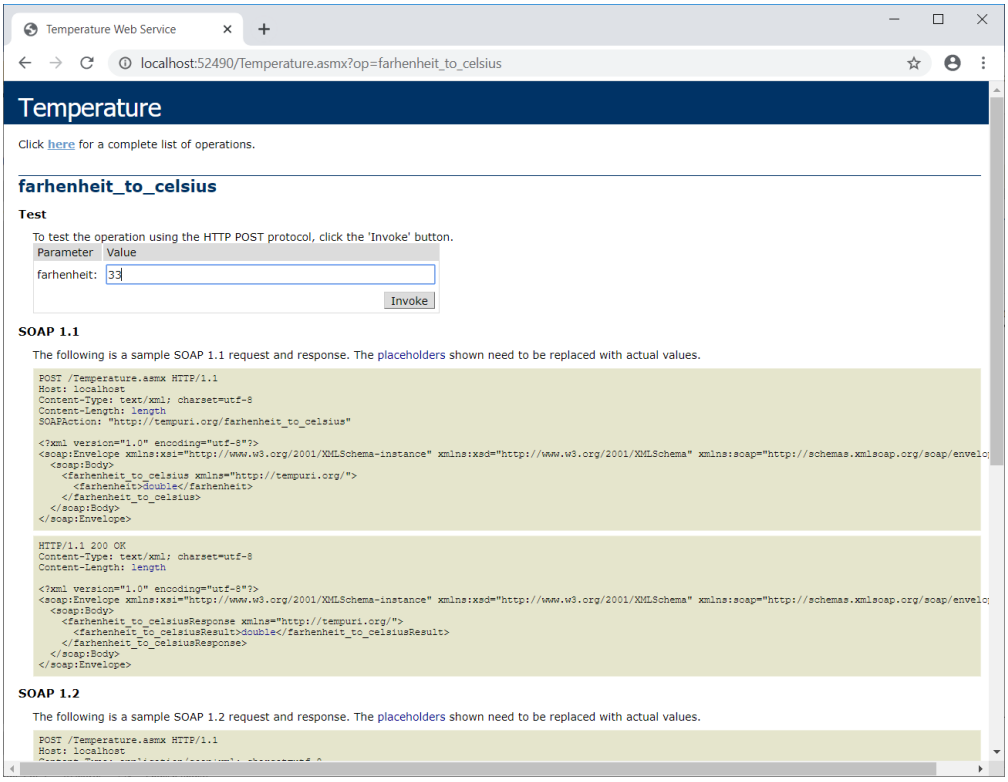
OUTPUT

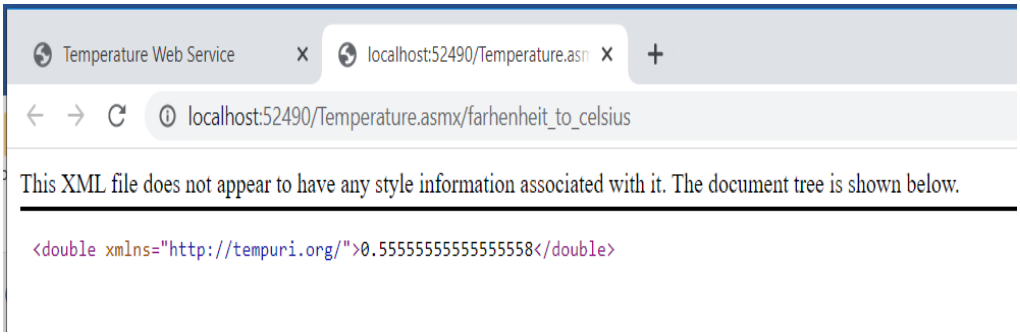
(WebService)



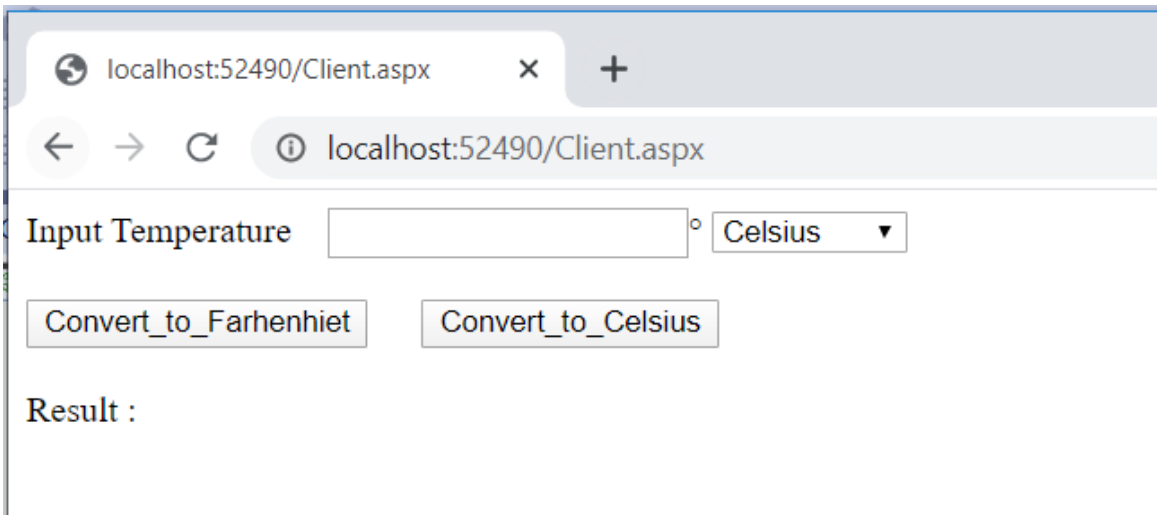


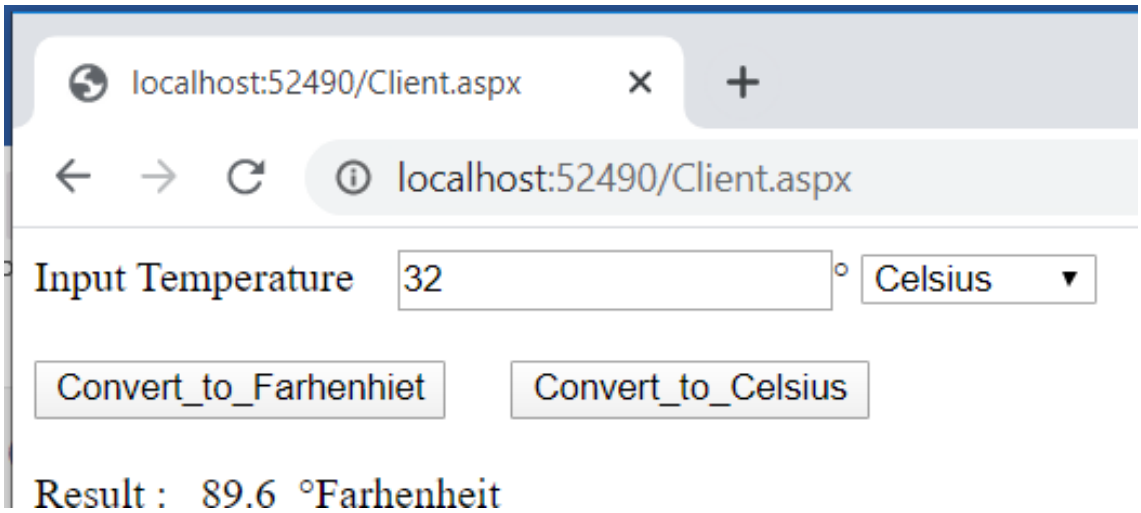


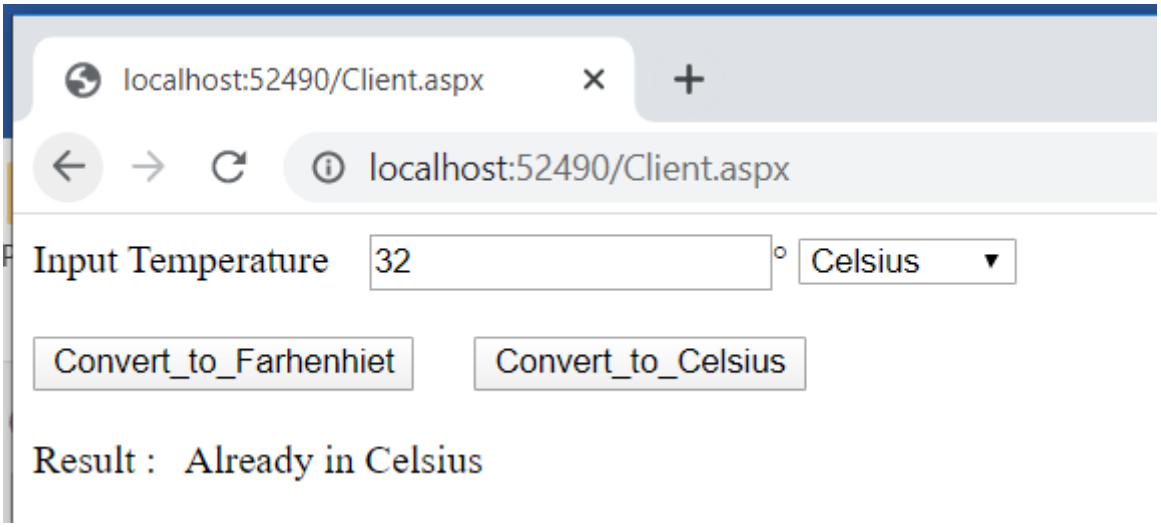

Temperature Web Service
localhost:52490/Temperature.asmx?op=farhenheit_to_celsius
Temperature
Click here for a complete list of operations.
farhenheit_to_celsius
Test
To test the operation using the HTTP POST protocol, click the 'Invoke' button.
Parameter Value
farhenheit: 33
Invoke
SOAP 1.1
The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.
POST /Temperature.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/farhenheit_to_celsius"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelo
<soap:Body>
<farhenheit_to_celsius xmlns="http://tempuri.org/">
<farhenheit>double</farhenheit>
</farhenheit_to_celsius>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelo
<soap:Body>
<farhenheit_to_celsiusResponse xmlns="http://tempuri.org/">
<farhenheit_to_celsiusResult>double</farhenheit_to_celsiusResult>
</farhenheit_to_celsiusResponse>
</soap:Body>
</soap:Envelope>
SOAP 1.2
The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.
POST /Temperature.asmx HTTP/1.1
Host: localhost

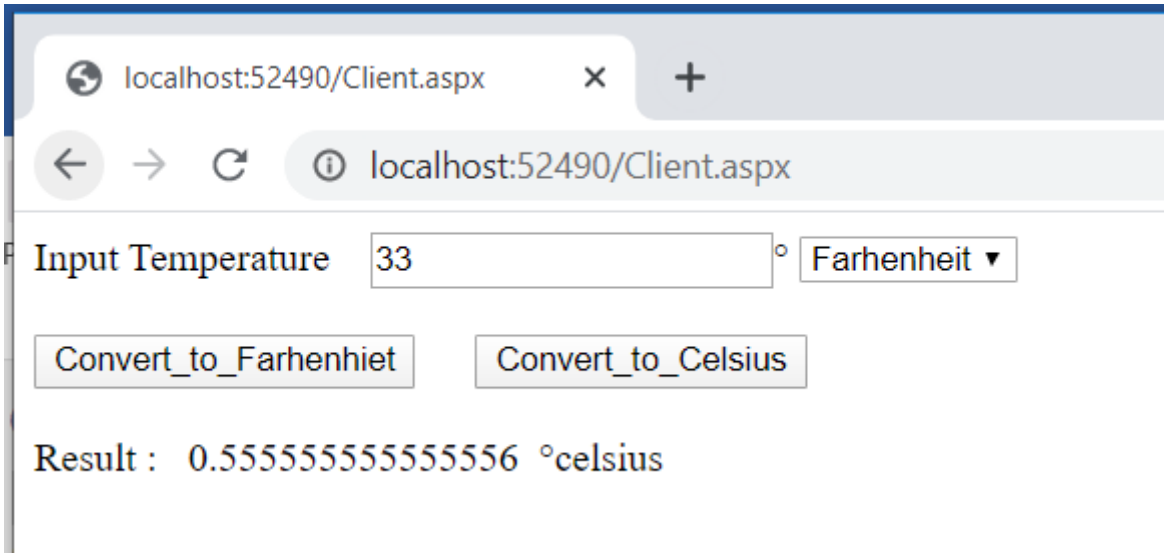

Temperature Web Service
localhost:52490/Temperature.asm
localhost:52490/Temperature.asmx/farhenheit_to_celsius
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<double xmlns="http://tempuri.org/">0.55555555555555558</double>

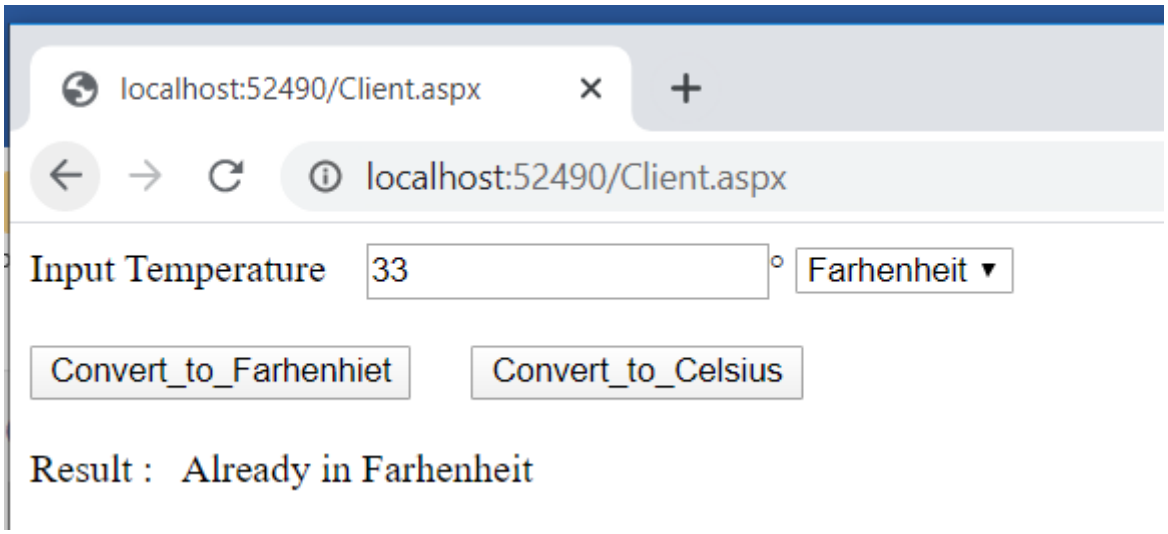
(Client)

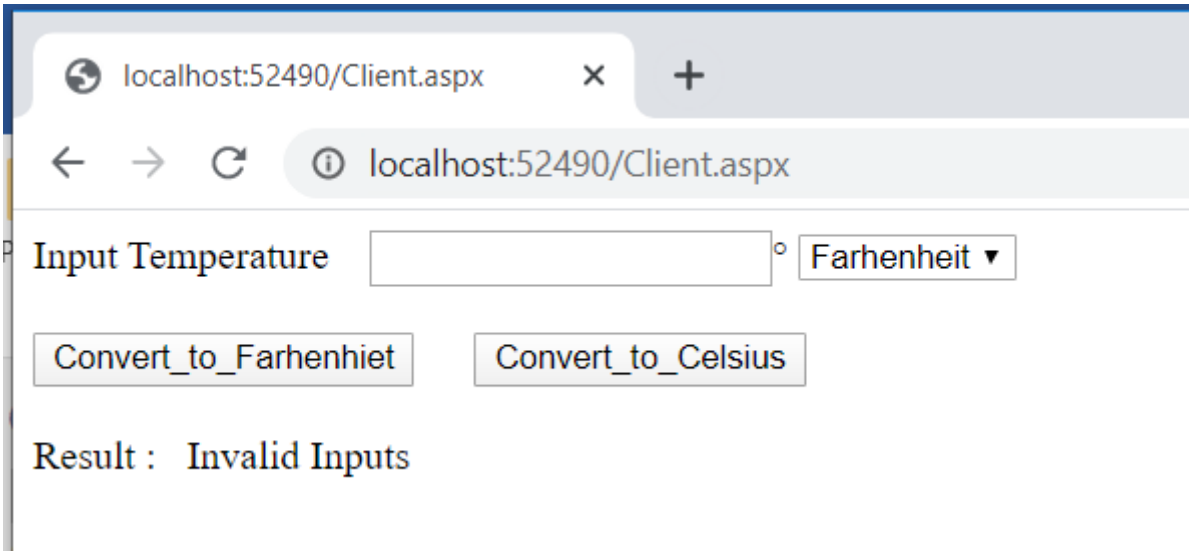

localhost:52490/Client.aspx
localhost:52490/Client.aspx
Input Temperature
°
Celsius
Convert_to_Farhenhiet
Convert_to_Celsius
Result :


localhost:52490/Client.aspx
localhost:52490/Client.aspx
Input Temperature
32
°
Celsius
Convert_to_Farhenhiet
Convert_to_Celsius
Result : 89.6 °Farhenheit


localhost:52490/Client.aspx
localhost:52490/Client.aspx
Input Temperature
32
°
Celsius
Convert_to_Farhenhiet
Convert_to_Celsius
Result : Already in Celsius


localhost:52490/Client.aspx
localhost:52490/Client.aspx
Input Temperature
33
°
Farhenheit
Convert_to_Farhenhiet
Convert_to_Celsius
Result : 0.555555555555556 °celsius


localhost:52490/Client.aspx
localhost:52490/Client.aspx
Input Temperature
33
°
Farhenheit
Convert_to_Farhenhiet
Convert_to_Celsius
Result : Already in Farhenheit


localhost:52490/Client.aspx
localhost:52490/Client.aspx
Input Temperature
°
Farhenheit
Convert_to_Farhenhiet
Convert_to_Celsius
Result : Invalid Inputs