|  |  |
| --- | --- |
| **Roll No.:** A059 | **Name:** Chinmay Parikh |
| **Prg/Yr/Sem:** B.Tech(I.T.)/4th /7 | **Batch:** A3 |
| **Date of Experiment:** 18/10/2014 | **Date of Submission:** 2/11/2014 |

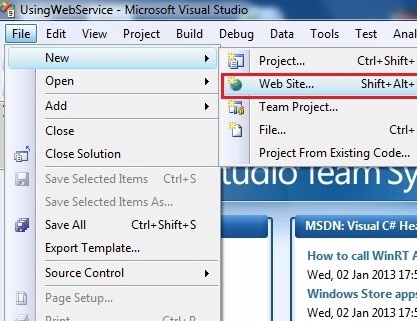
1. **Design Scenario**

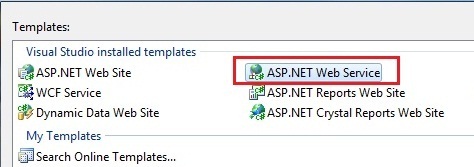
**Ans:** Input three subject marks and display the average of the three subjects marks as the web servce.

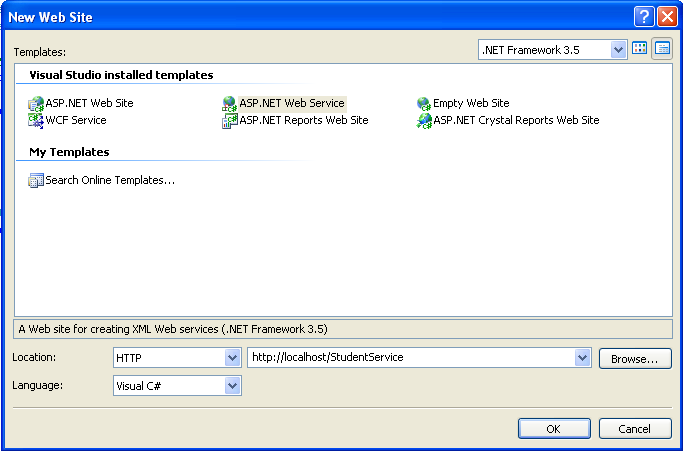
1. **Detailed Steps for completing the Practical**

**Ans: Steps:** Let’s start with the creation of the WebService.

1) Open Visual Studio 2008 Client. Click on New>WebSite>ASP.NET Web Service. Give the path where you want to place the code. Refer to the below screenshots.







2) Select location as HTTP and give service name as StudentService language Visual c# , click on ok as shown in figure above.

3) Visual Studio will open a default Service.cs file in which you can write your methods. Write the below lines of code which is highlighted.

using System;

using System.Linq;

using System.Web;

using System.Web.Services;

using System.Web.Services.Protocols;

using System.Xml.Linq;

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

// 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 Service : System.Web.Services.WebService

{

public Service () {

//Uncomment the following line if using designed components

//InitializeComponent();

}

[WebMethod]

public String Result(float m1, float m2, float m3)

{

float res= (m1+m2+m3)/3;

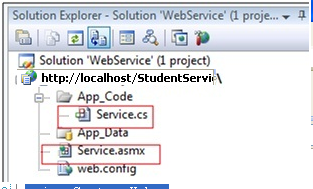
return "The result is " + res;

}

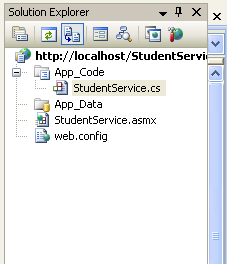
}

4) Rename the Service.cs and Service.asmx file in solution window as StudentService.cs and StudentService.asmx.

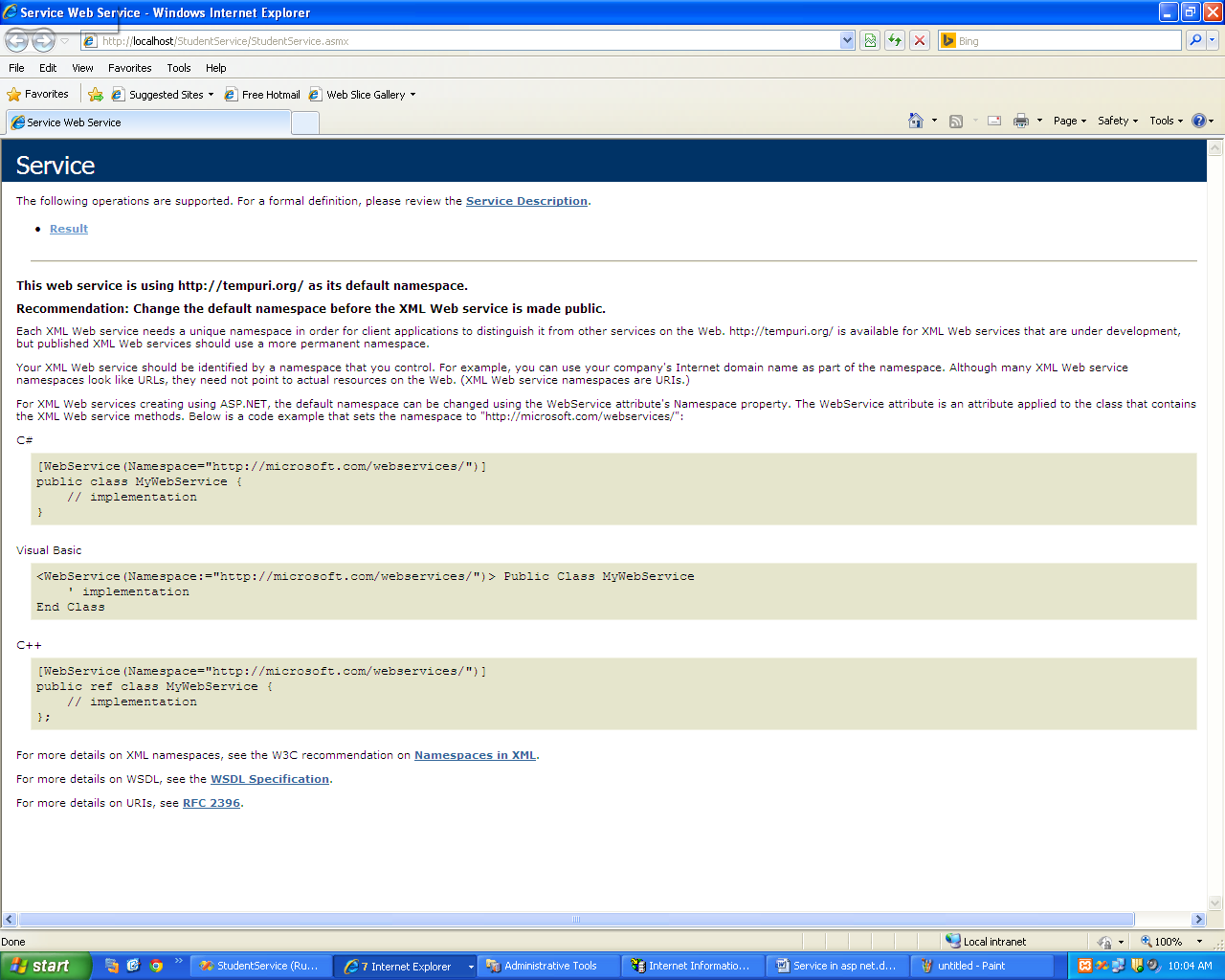
Build the solution and view the Service.asmx file in the browser.



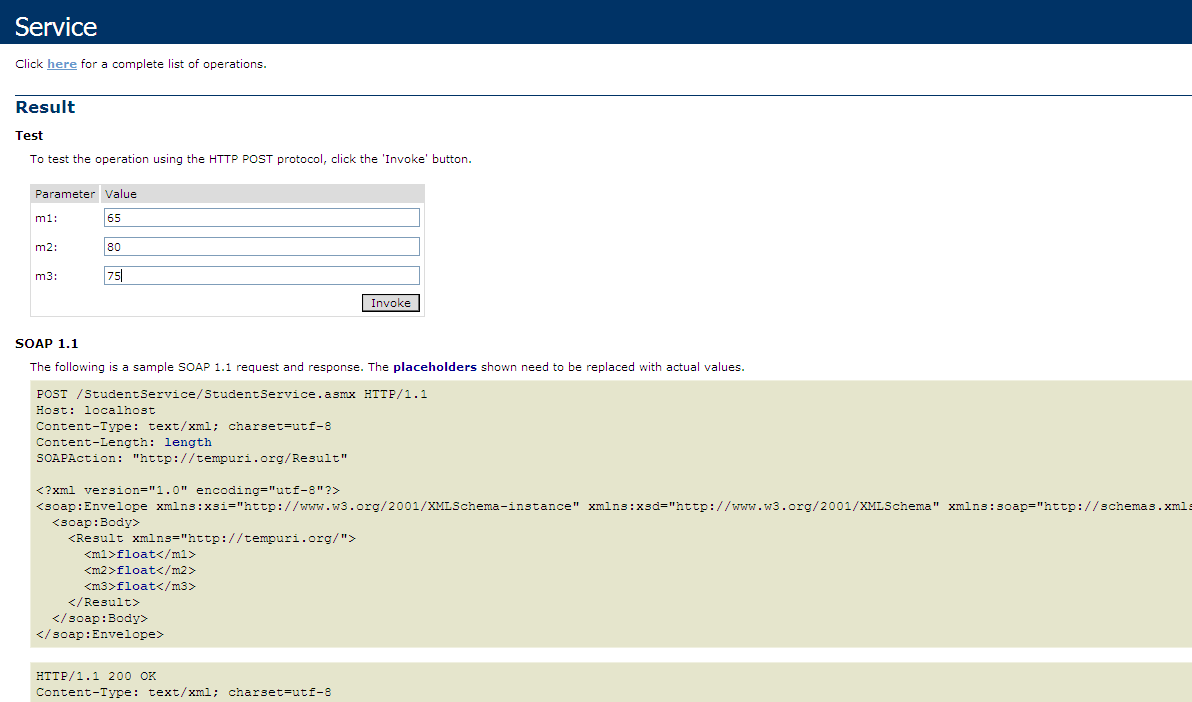
Now it will look like



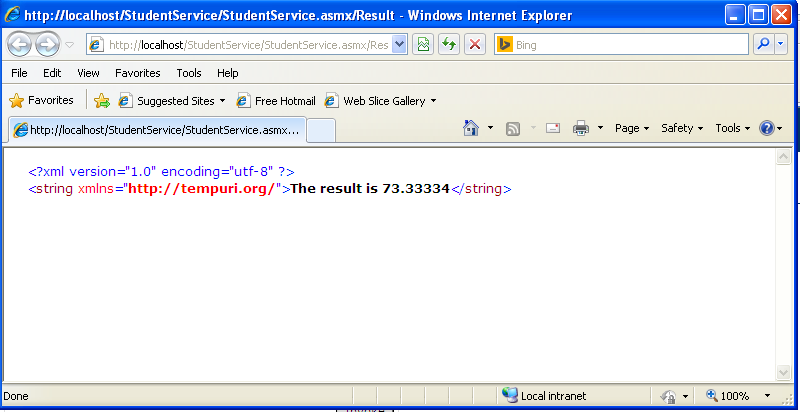
5) Build the solution and view the StudentService.asmx file in the browser. You can run the StudentSerive by play button on menu bar or press F5 key.



6) Click on Result link and add values for marks and click on invoke to test the service



Result will be



So here it is, a very basic WebService has been coded and created, which will help us to calculate the result of the student.

Next, we will see how to call and use the Web Service in times of need.

Well, there may be 3 methods by which a Web Service is called, depending upon the scenarios.

1. **By HTTP POST method.**

**HTTP POST method**: This method can be used directly by calling the **.asmx** file from the client.

The most favorable method to execute an HTTP POST method is to have the codes inside an HTML file. Let’s get started to implement it.

1. **By creating a Proxy Class.**

Will be discussed in detail.

**3)     By using XMLHTTP Request object over SOAP**

The XMLHttpRequest specification defines an API that provides scripted client functionality for transferring data between a client and a server.

It is used to send [HTTP](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) or [HTTPS](http://en.wikipedia.org/wiki/HTTP_Secure) requests directly to a [web server](http://en.wikipedia.org/wiki/Web_server) and load the server response data directly back into the script.

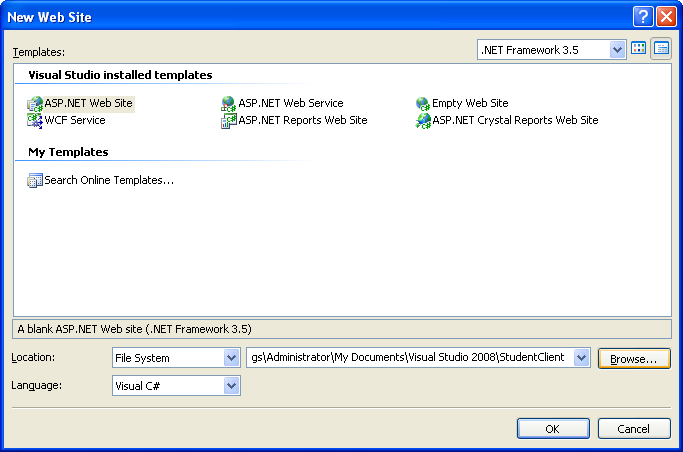
In this implementation we will use an HTML form and a JavaScript function to demonstrate the full potential of this method.

**Let’s consider method 2 i.e. to call a webservice by creating a Proxy class. (Creating a Client for the service)**

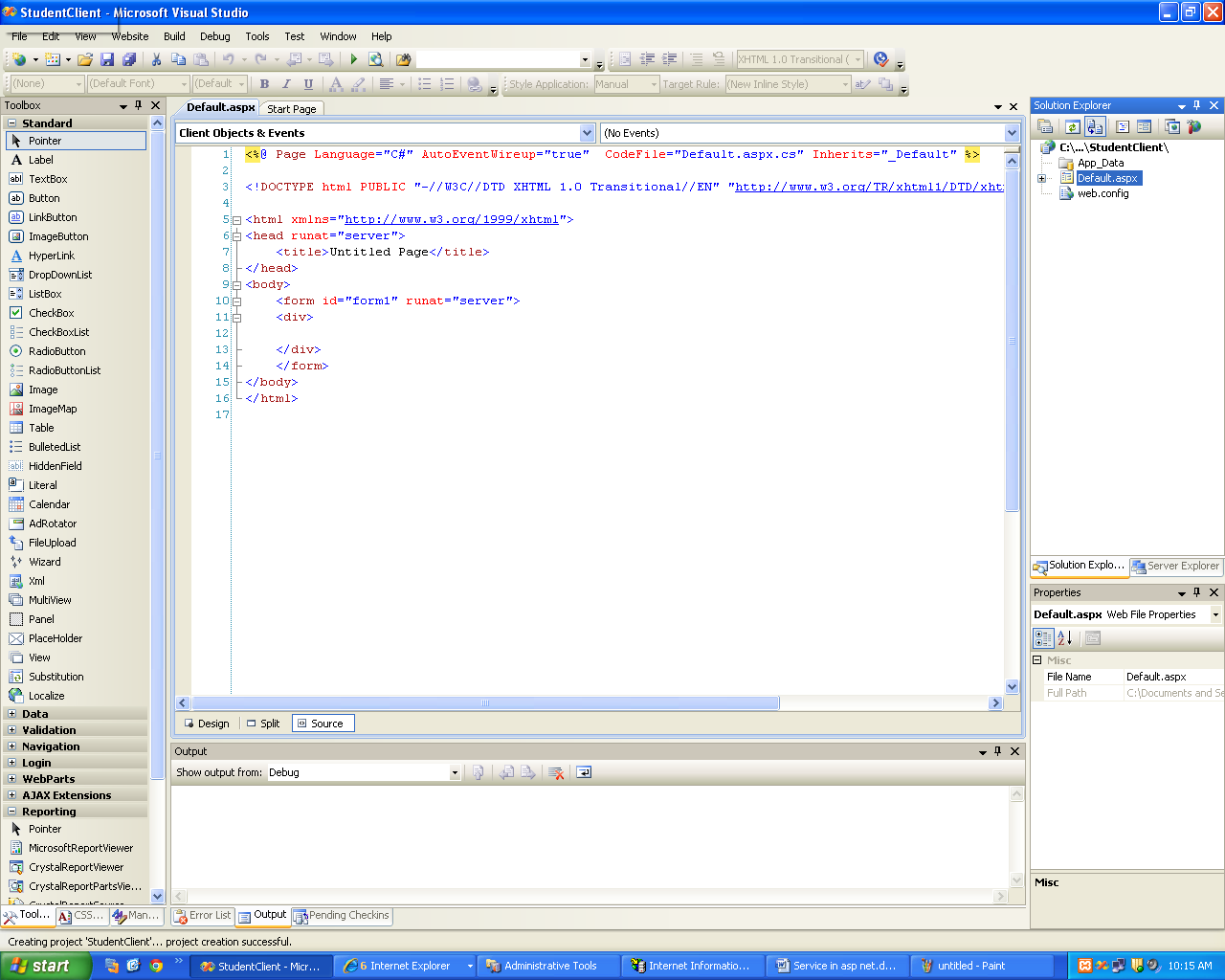
**Creation of a Proxy Class**: This method is widely used by developers. This method actually makes use of the full potential of a Web Service. Let’s implement this and see its wonders.

We need to create one ASP.NET website to see and implement this method.

Open Visual Studio once again and click on New>Website>ASP.NET Website. Give the path where you want to place the code as shown if figure below. Click on OK.



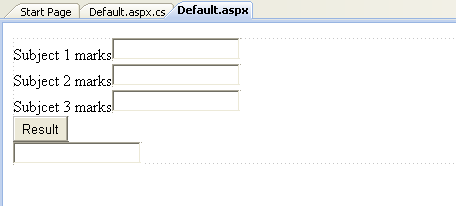
The screen will look like this



This web site creation will open a Default.aspx page, its aspx.cs page, an App\_Code folder and few more stuffs.

Open the Default.aspx page and design it as per below screenshot:

It’s basically 3 ASP label controls, 4 ASP text box controls and 1 ASP button bound under 1 div.



The code for default page will be

<%@ 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>Untitled Page</title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:Label ID="lbsub1" runat="server" Text="Subject 1 marks"></asp:Label>

<asp:TextBox ID="txtsub1" runat="server"></asp:TextBox>

<br />

<asp:Label ID="lbsub2" runat="server" Text="Subject 2 marks"></asp:Label>

<asp:TextBox ID="txtsub2" runat="server"></asp:TextBox>

<br />

<asp:Label ID="lbsub3" runat="server" Text="Subjcet 3 marks"></asp:Label>

<asp:TextBox ID="txtsub3" runat="server"></asp:TextBox>

<br />

<asp:Button ID="Btresult" runat="server" Text="Result"

onclick="Btresult\_Click" />

<br />

<asp:TextBox ID="txtResult" runat="server"></asp:TextBox>

</div>

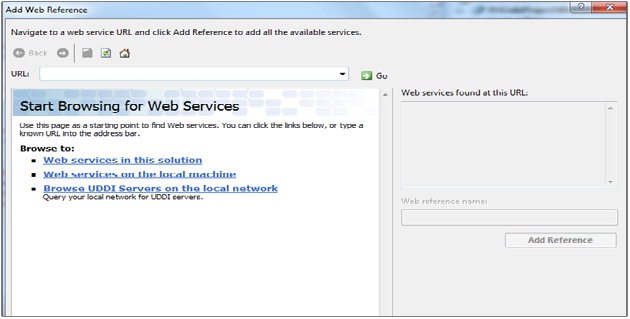
</form>

</body>

</html>

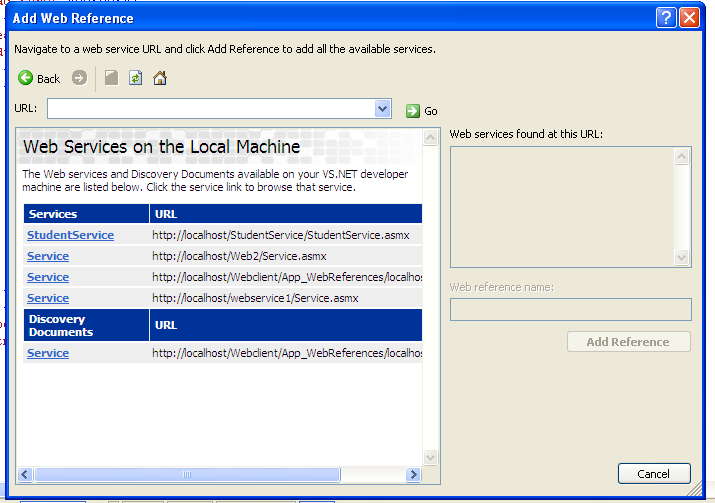
Now the most important aspect of this very method comes into play. Addition of the previously created StudentService.

Right-click the project and click on “Add Web Reference”. The click will open a dialog box similar to the below screenshot.

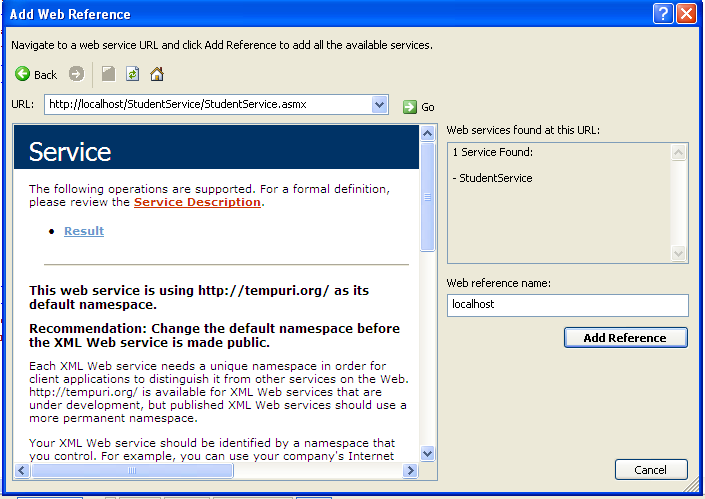


Now, we need to catch the Web Service here either by adding the URL or by browsing through the various options given.

Click on Web service on the local machine. All the available services will be listed down.

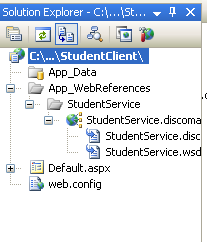


Select StudentService.



Successful addition of the Web Service will yield “Web services found at this URL”. You would see the methods that you have actually created. You might want to change the name of the web reference from “localhost” to something StudentService. Click on Add Reference and there you go. You have successfully added a Web Reference to your ASP.NET website.

The solution explorer will be something like below:



Now, once the Web Reference is added, let’s write some code in Code-Behind to make use of the web service.

Open **Default.aspx.cs** file and on Button Click event, add the following highlighted code:

using System;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Btresult\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(txtsub1.Text) && !string.IsNullOrEmpty(txtsub2.Text) && !string.IsNullOrEmpty(txtsub3.Text))

{

StudentService.Service obj = new StudentService.Service();

float m1 = Convert.ToInt32(txtsub1.Text);

float m2 = Convert.ToInt32(txtsub2.Text);

float m3 = Convert.ToInt32(txtsub3.Text);

txtResult.Text = Convert.ToString(obj.Result(m1,m2,m3));

}

}

}

Now, let see explain the above code further. The bold line of code is what this method is all about, Creation of proxy class.

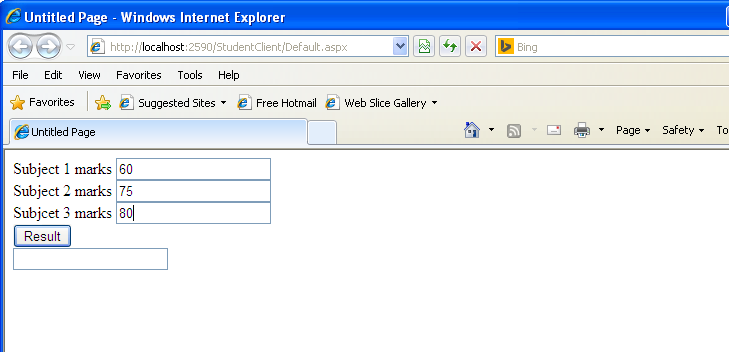
We are making an object of the Web Service added and then are using that object to call the individual methods of the WebService. Rest of the code is self-explanatory, which will calculate the result for the student.

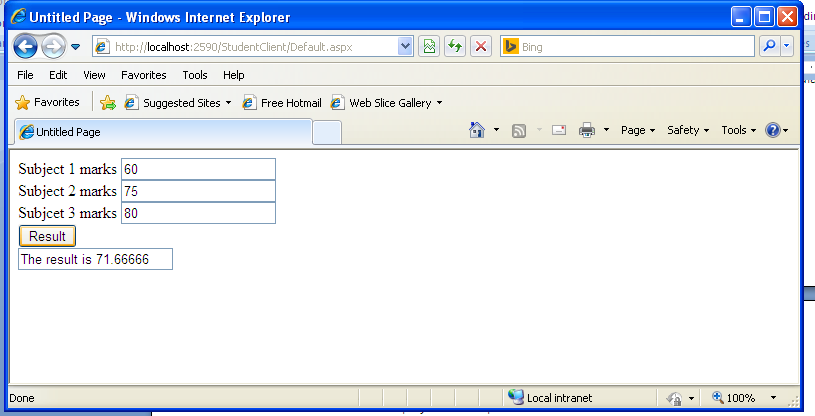
Build the solution and view the Default.aspx file in the browser.

Enter some values for marks and click on the button.

You will see the result displayed in the placed textbox.

Similar screenshots are below.



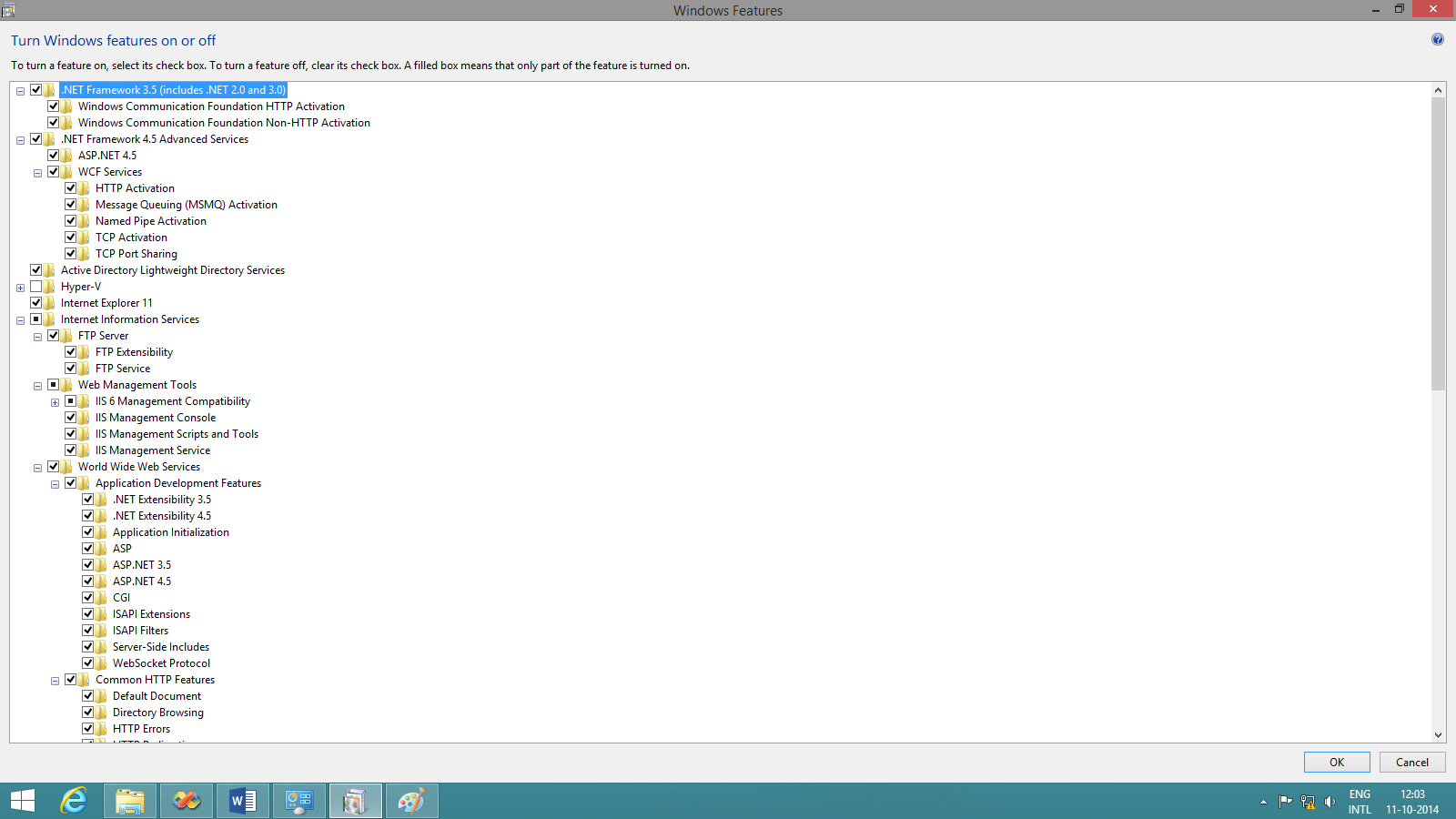


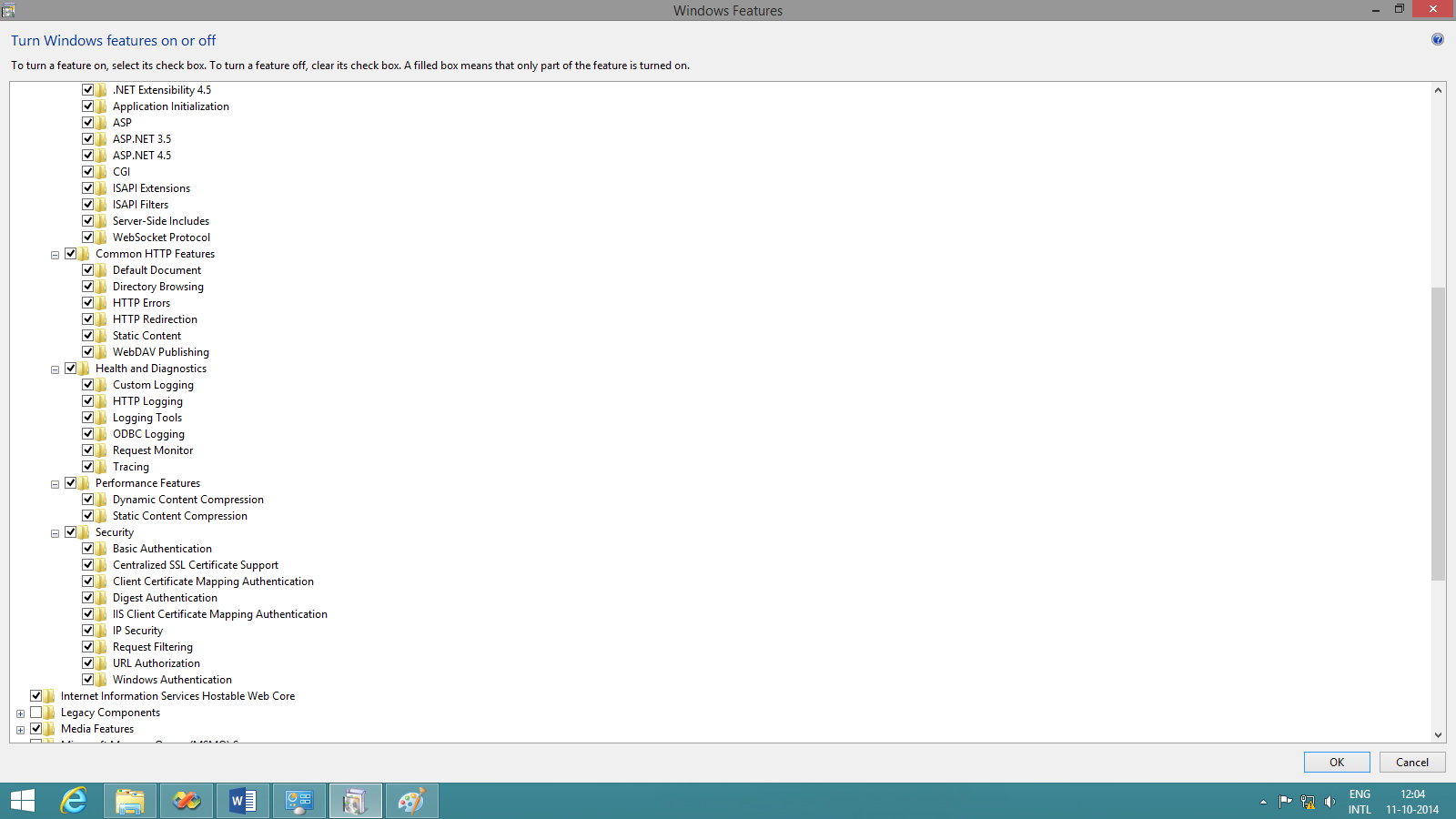
**Note –**

Work as Admin Login

For Visual Studio 2008 to start “right click” and select “run as admin”.

Screen shots of Control Panel (Select Control Panel->Programs->Turn On and Off Windows features”





Create the Project Web Service Application

Then Build the Project

To run the project press Ctrl-F5

This executes the service.

Now create the client.

1. **Output Screen Shots**

WSDL-

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

<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/" xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://tempuri.org/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

<wsdl:types>

<s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">

<s:element name="Result">

<s:complexType>

<s:sequence>

<s:element minOccurs="1" maxOccurs="1" name="m1" type="s:float" />

<s:element minOccurs="1" maxOccurs="1" name="m2" type="s:float" />

<s:element minOccurs="1" maxOccurs="1" name="m3" type="s:float" />

</s:sequence>

</s:complexType>

</s:element>

<s:element name="ResultResponse">

<s:complexType>

<s:sequence>

<s:element minOccurs="0" maxOccurs="1" name="ResultResult" type="s:string" />

</s:sequence>

</s:complexType>

</s:element>

</s:schema>

</wsdl:types>

<wsdl:message name="ResultSoapIn">

<wsdl:part name="parameters" element="tns:Result" />

</wsdl:message>

<wsdl:message name="ResultSoapOut">

<wsdl:part name="parameters" element="tns:ResultResponse" />

</wsdl:message>

<wsdl:portType name="ServiceSoap">

<wsdl:operation name="Result">

<wsdl:input message="tns:ResultSoapIn" />

<wsdl:output message="tns:ResultSoapOut" />

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">

<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />

<wsdl:operation name="Result">

<soap:operation soapAction="http://tempuri.org/Result" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">

<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />

<wsdl:operation name="Result">

<soap12:operation soapAction="http://tempuri.org/Result" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="Service">

<wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">

<soap:address location="http://localhost/StudentService/StudentService.asmx" />

</wsdl:port>

<wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">

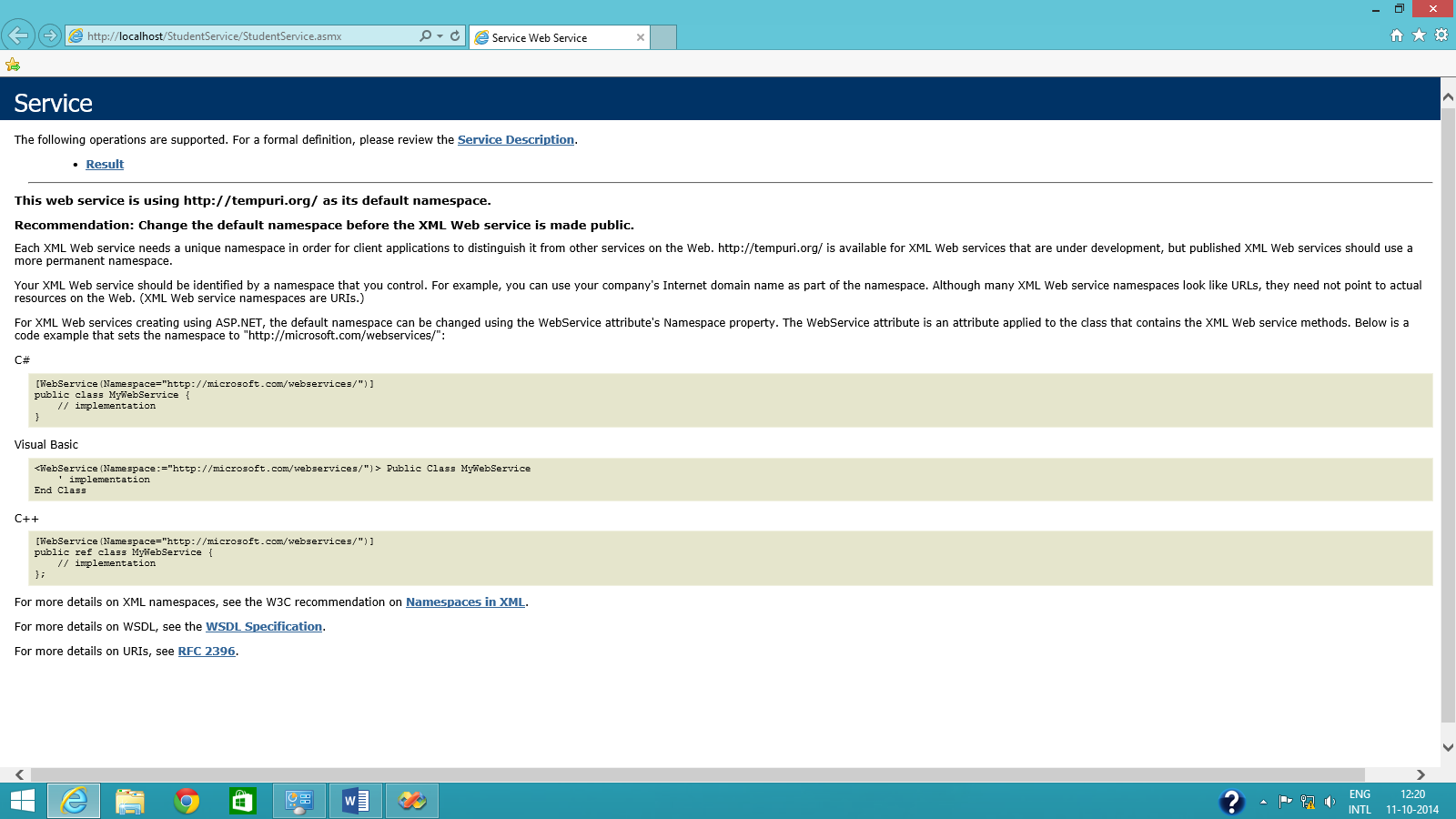
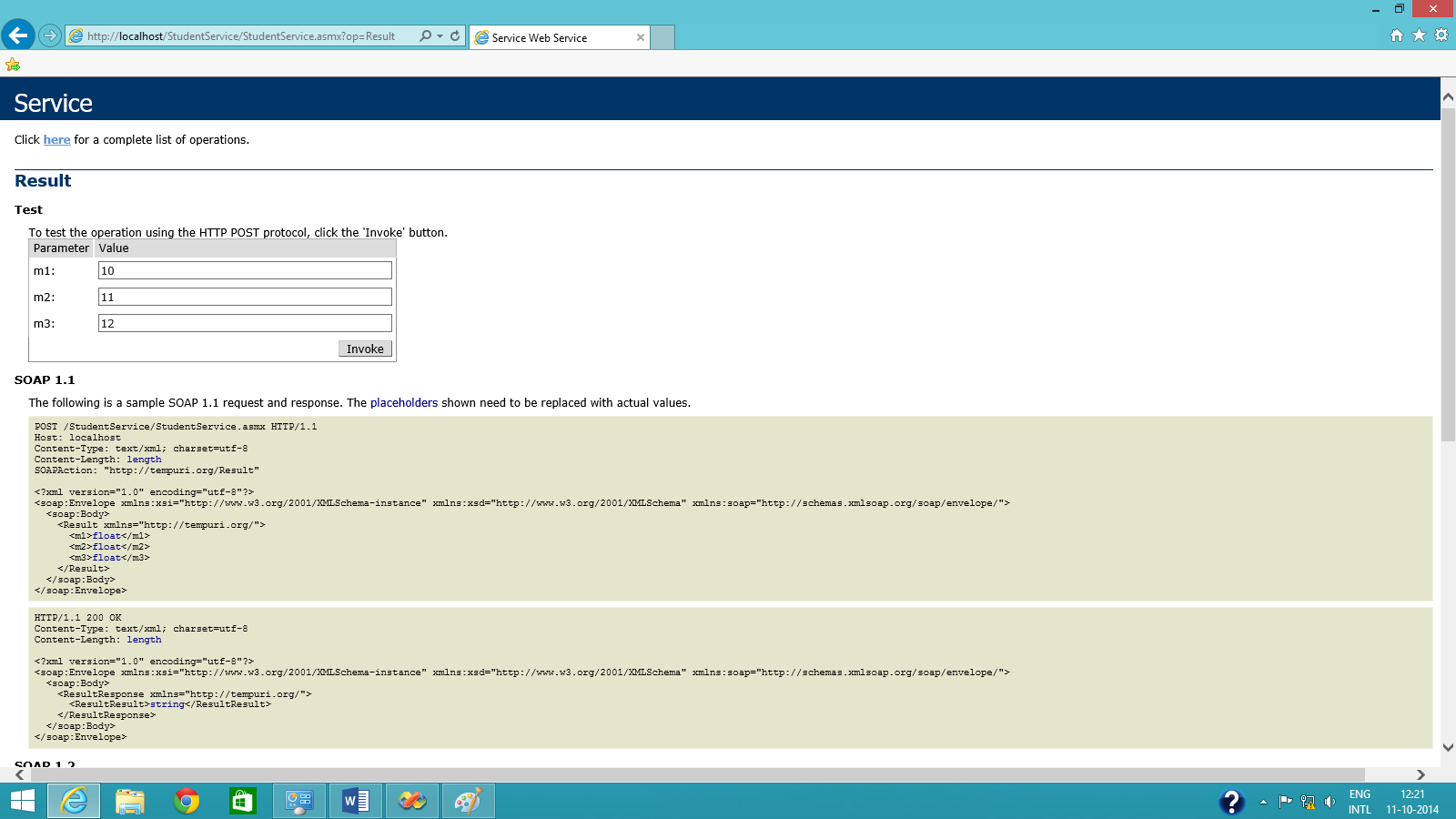
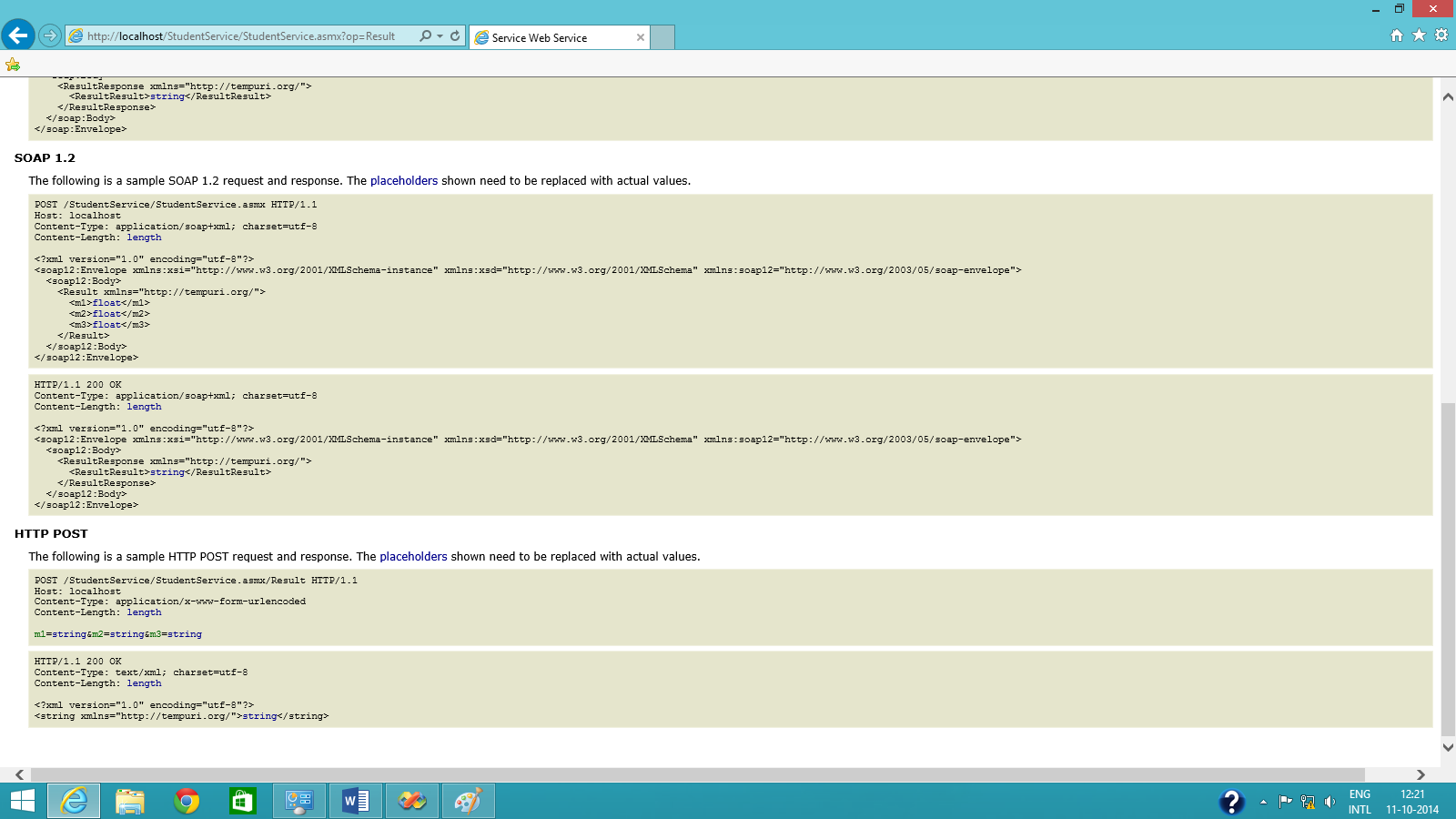
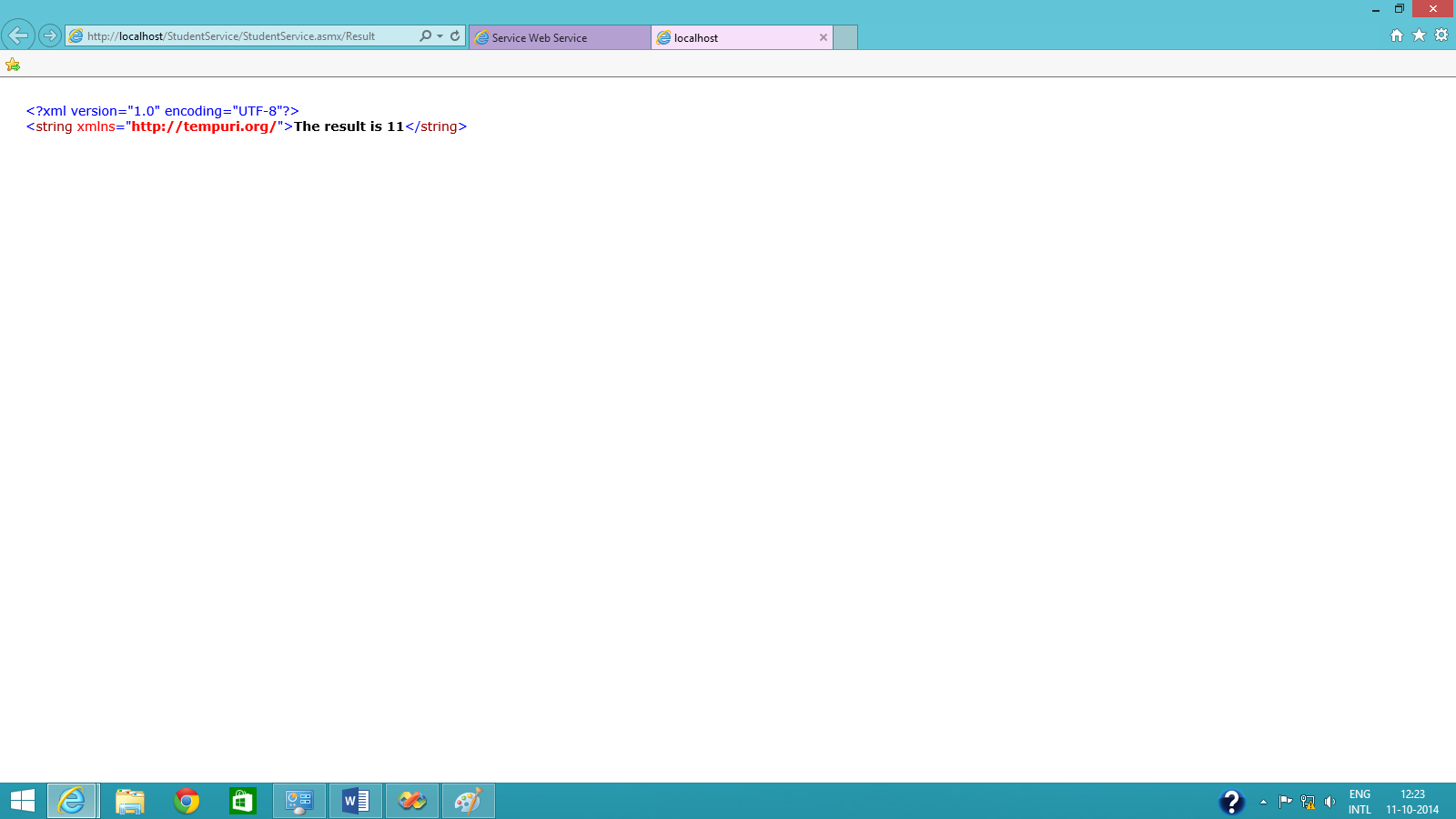
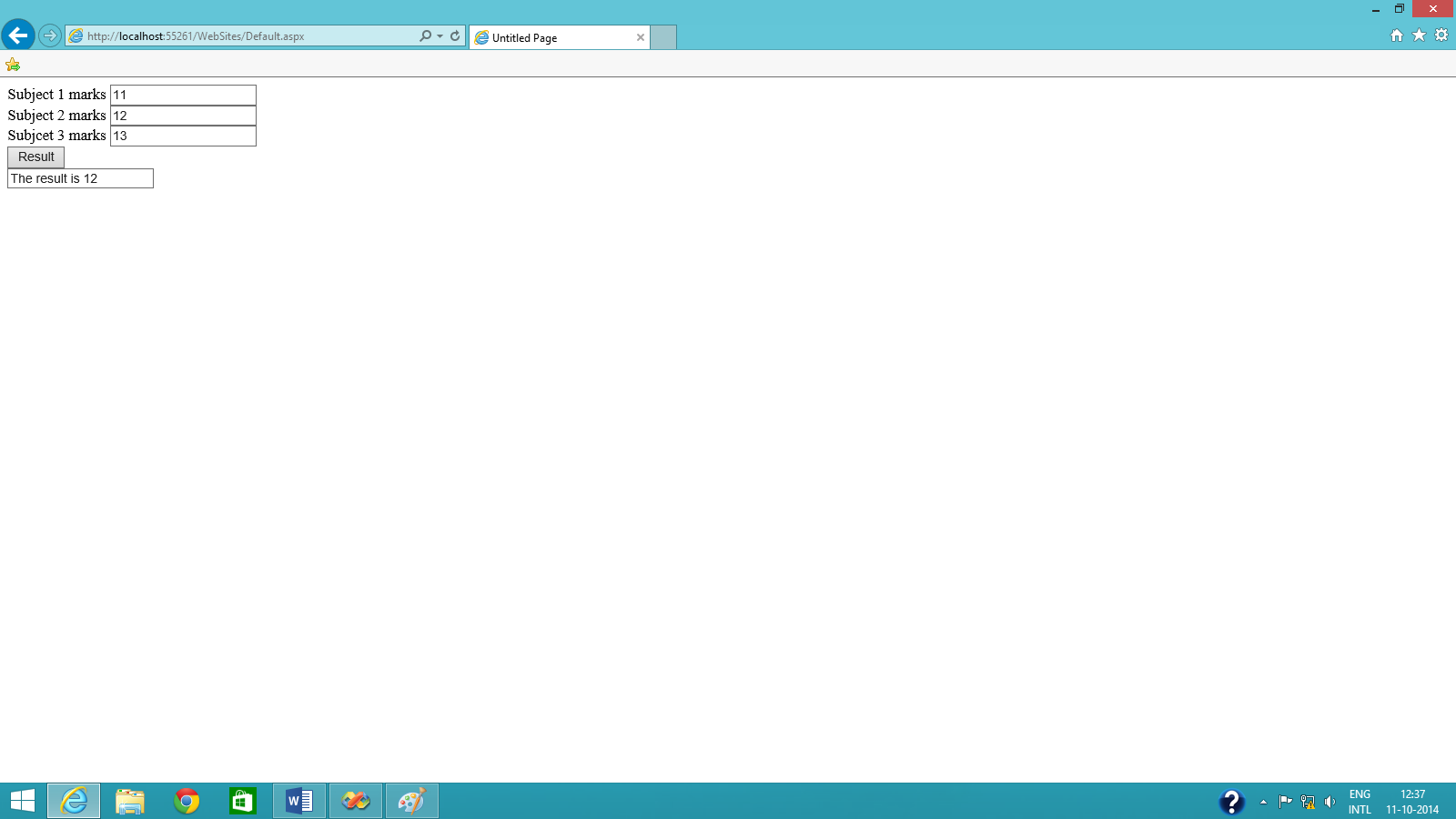
<soap12:address location="http://localhost/StudentService/StudentService.asmx" />

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

**OUTPUT-**

1. **Conclusion:** Thus we have studied about how to ASP.NET using Visual Studio and implemented the experiment. We inserted marks of three students and calculated the average of it.