|  |  |
| --- | --- |
| **Roll No.:** A059 | **Name:** Chinmay Parikh |
| **Prg/Yr/Sem:** B.Tech(I.T.)/4th /7 | **Batch:** A3 |
| **Date of Experiment:** 1/11/2014 | **Date of Submission:** 14/11/2014 |

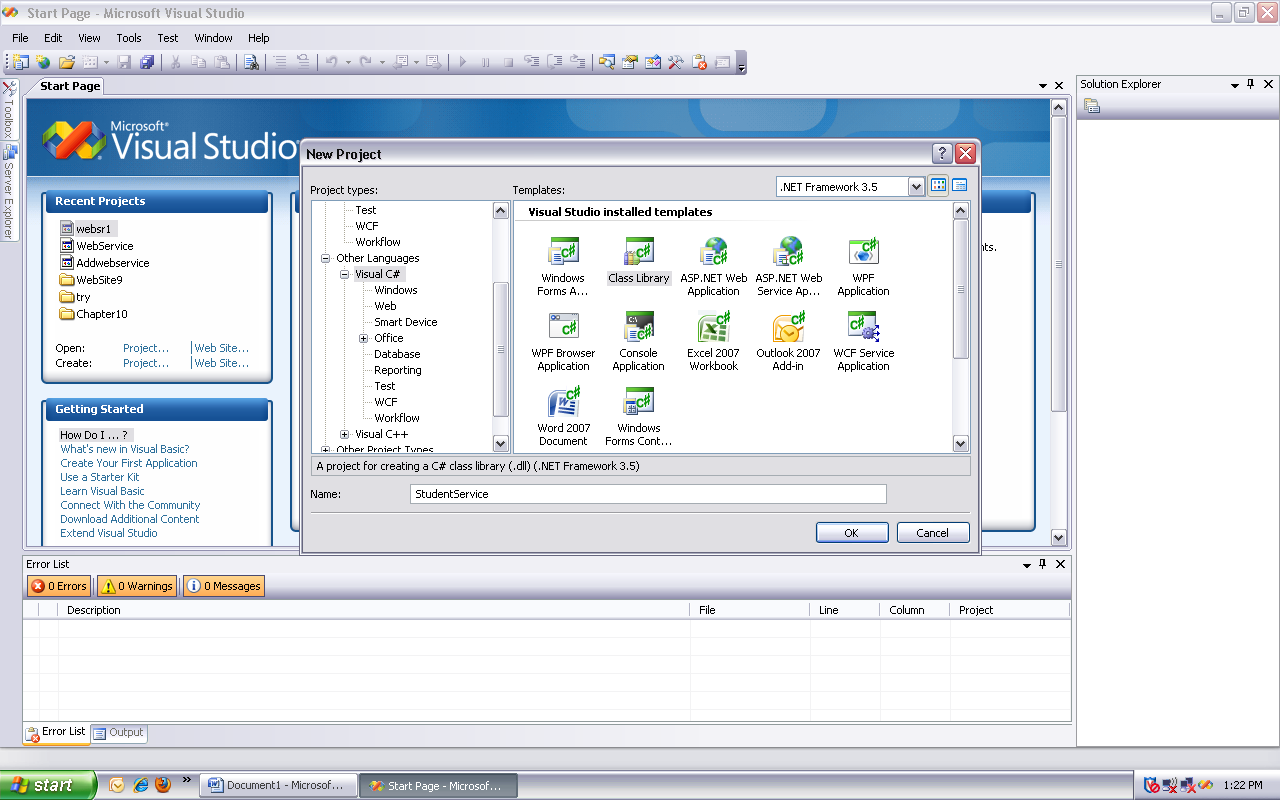
**Aim:** Create a WCF Service using ASP .NET (Visual Studio 2008) to calculate the percentage marks of three student subjects through a client service (ASP Page).

**Scenario:**

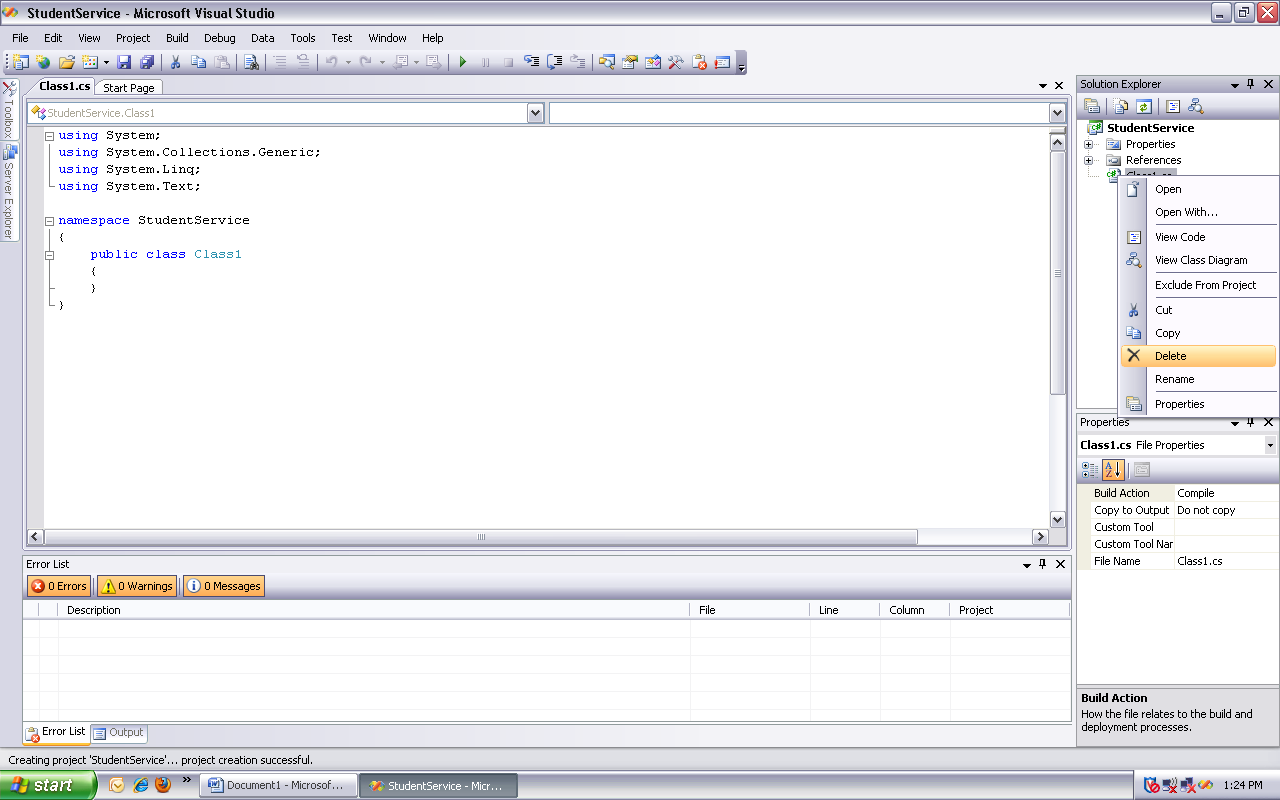
Input student’s three subject marks and display the percentage of three subject marks as the web service.

**Detailed Steps:**

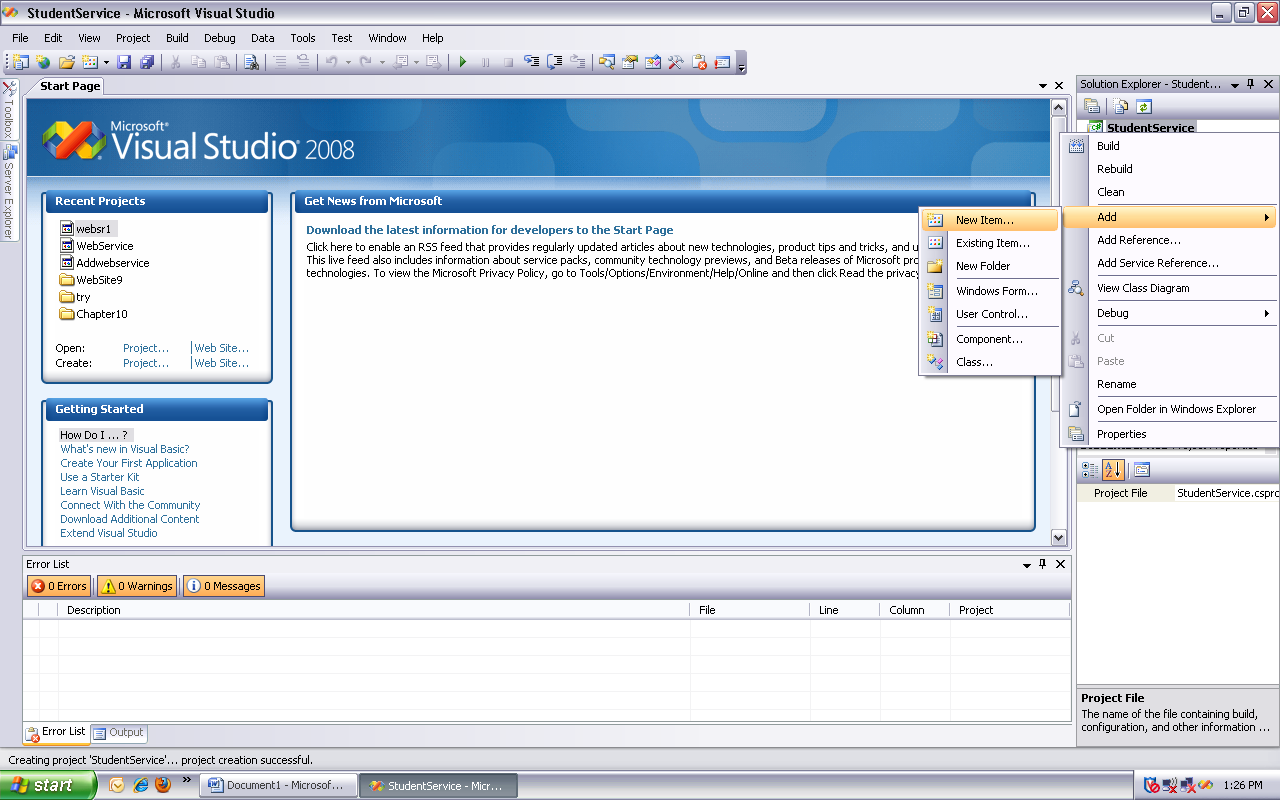
**Creating the WCF Service:**  
Create a new Class Library Project and name it **StudentService.**

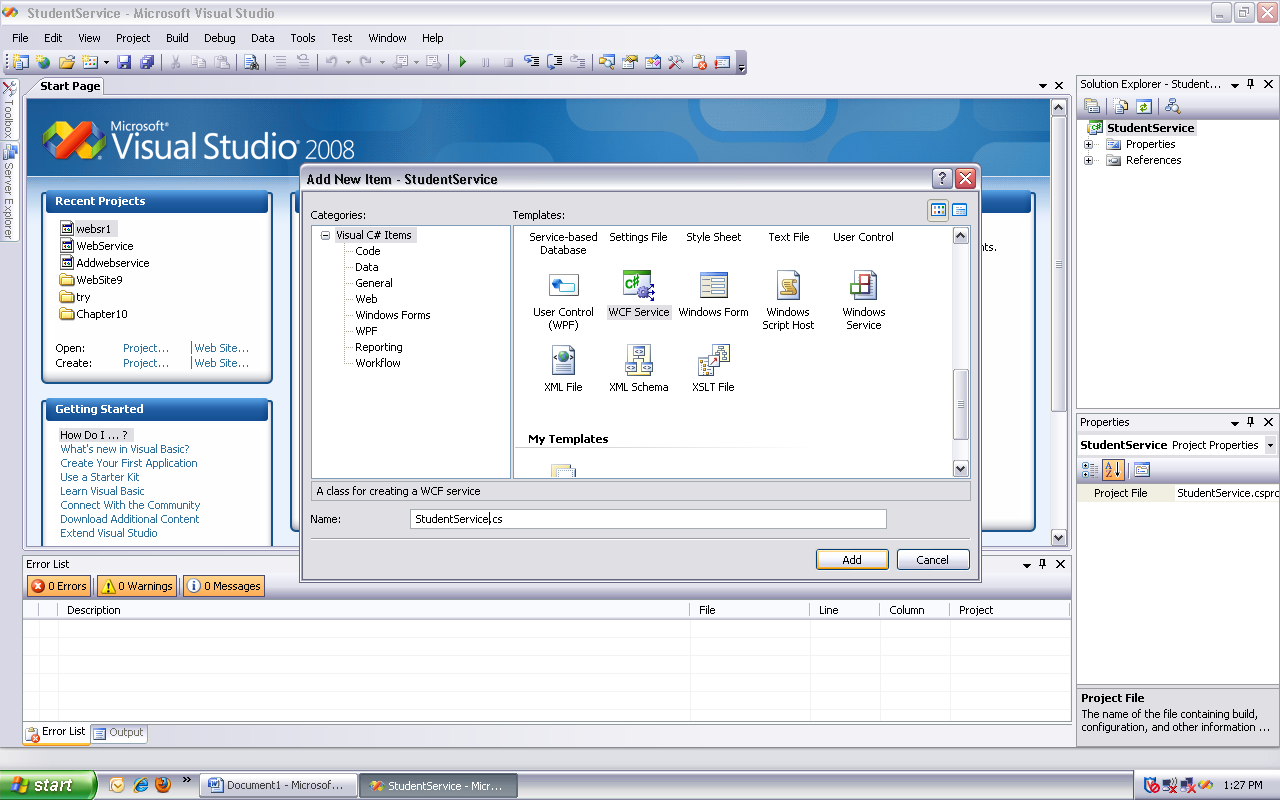


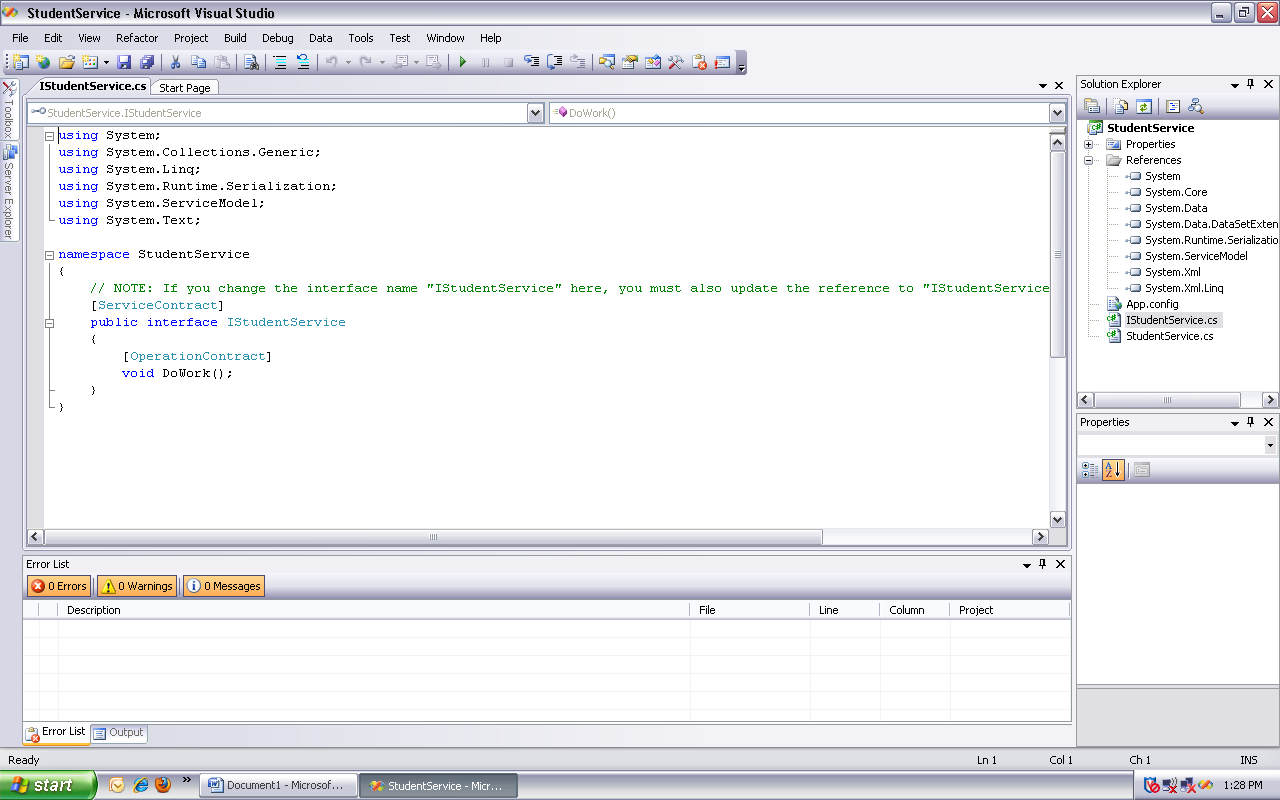
Delete **Class1.cs** file that is auto-generated.

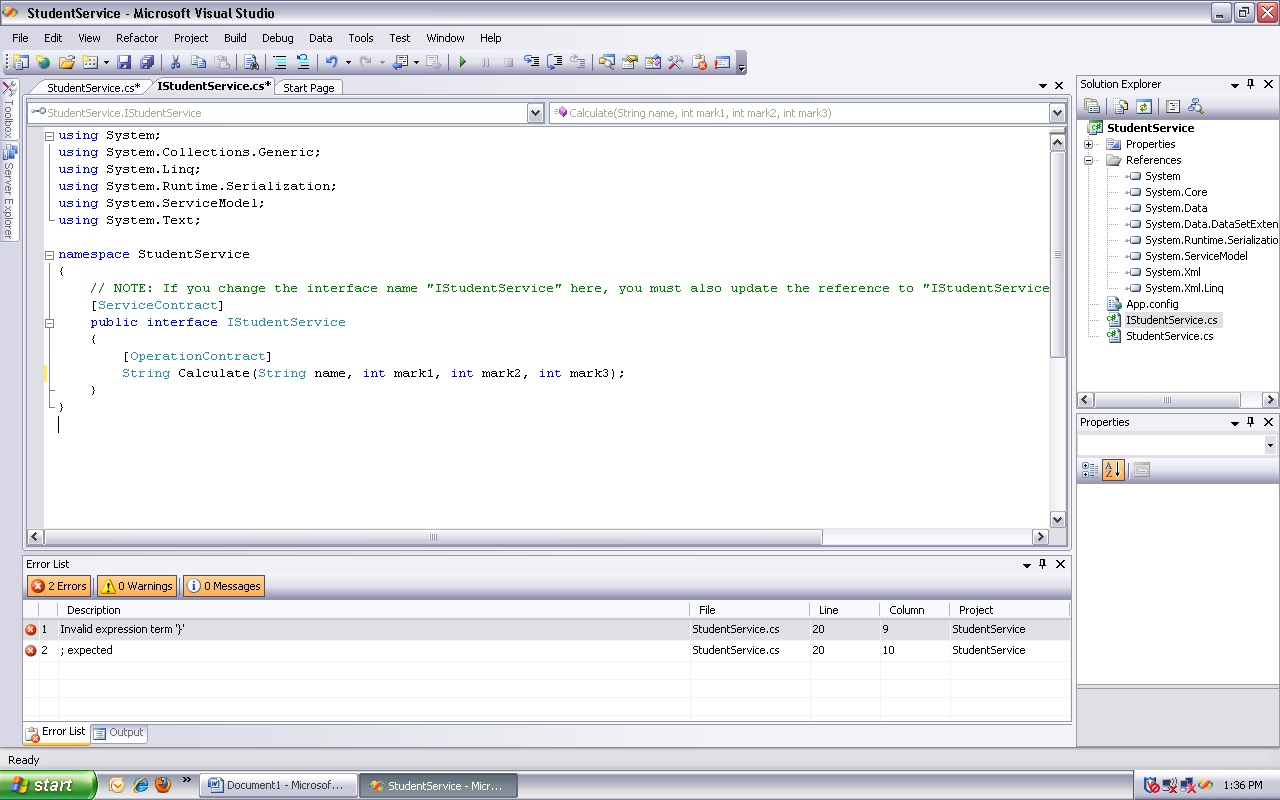


Add a new **WCF Service** with name = **StudentService.** This should automatically generate 2 files (**StudentService.cs** & **IStudentService.cs**). Also a reference to **System.ServiceModel** assembly is added.









Copy and paste the following code in **IStudentService.cs** file

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

namespace StudentService

{

// NOTE: If you change the interface name "IStudentService" here, you must also update the reference to "IStudentService" in App.config.

[ServiceContract]

public interface IStudentService

{

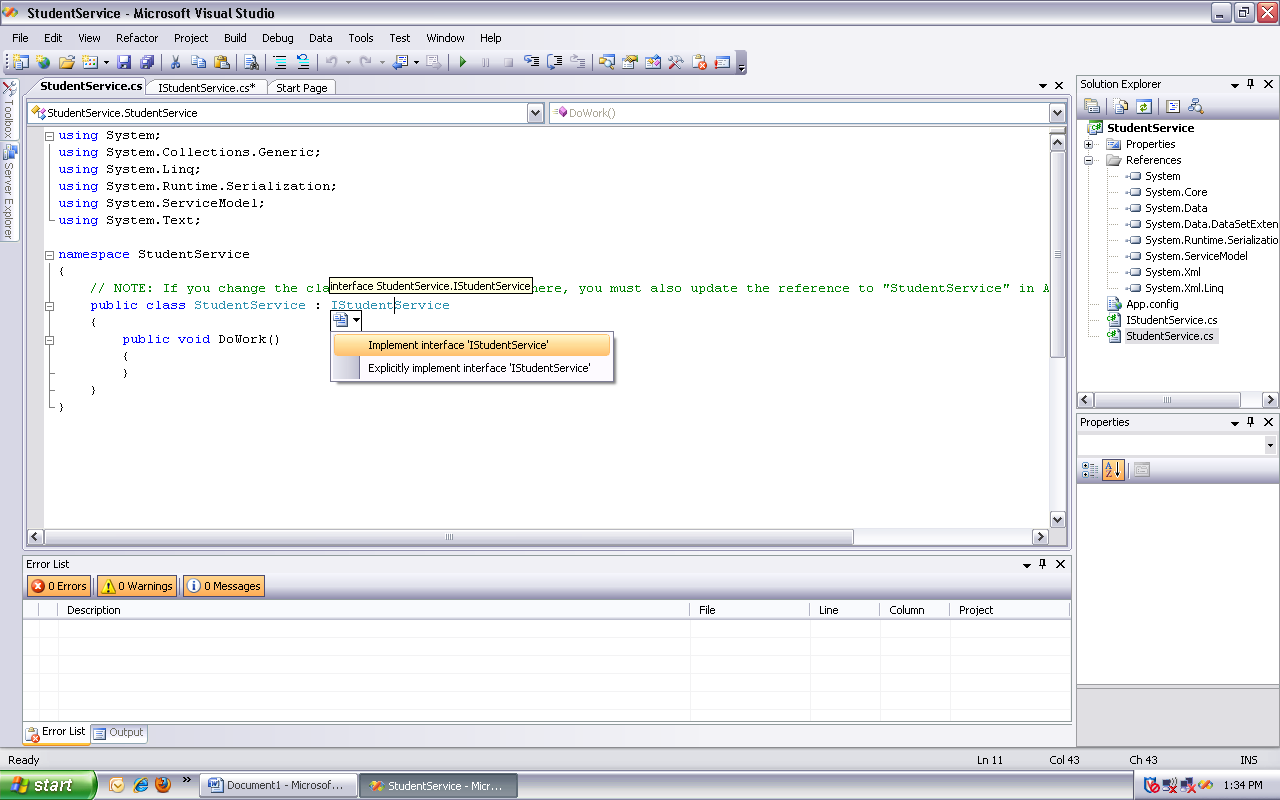
[OperationContract]

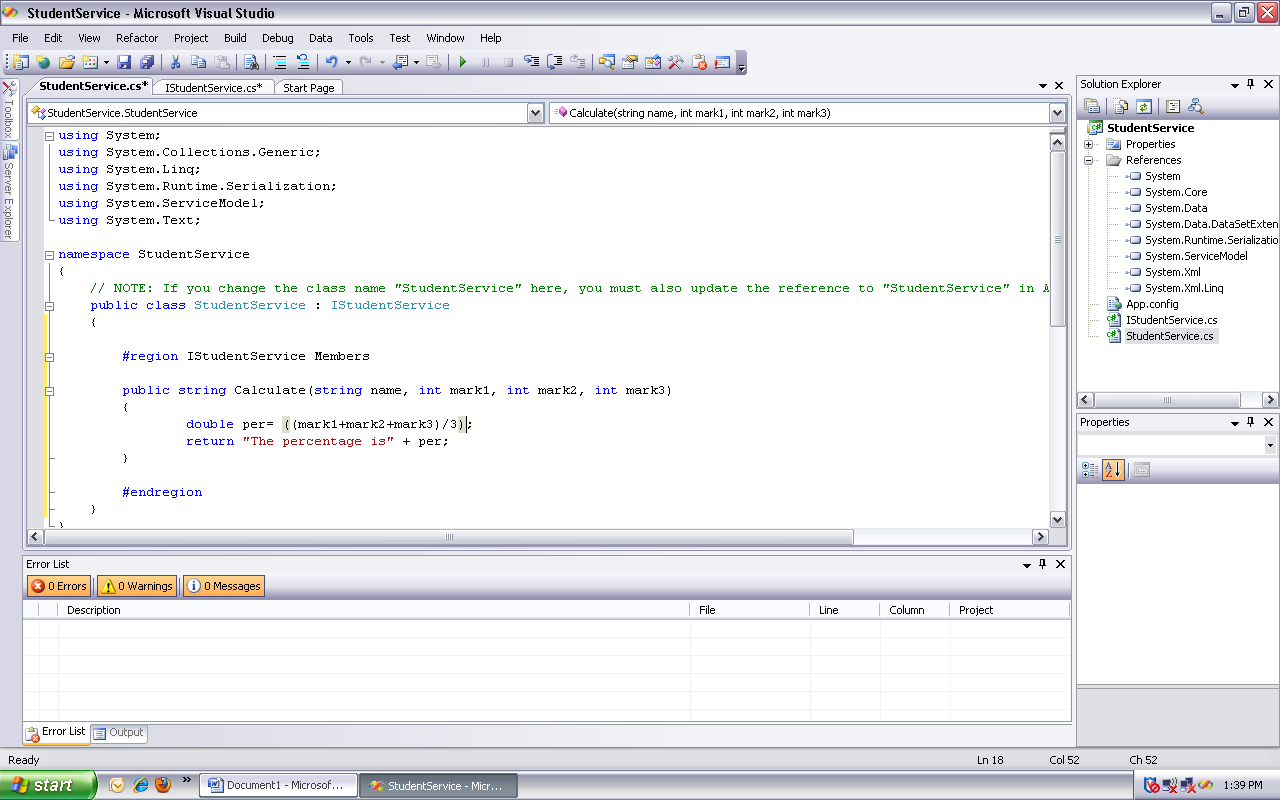
String Calculate(String name, int mark1, int mark2, int mark3);

}

}

Click on StudentService .cs file and implement the method of IStudentService.cs





Give functionality to the calculate method

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

namespace StudentService

{

// NOTE: If you change the class name "StudentService" here, you must also update the reference to "StudentService" in App.config.

public class StudentService : IStudentService

{

#region IStudentService Members

public string Calculate(string name, int mark1, int mark2, int mark3)

{

double per= ((mark1+mark2+mark3)/3);

return "The percentage is: " + per;

}

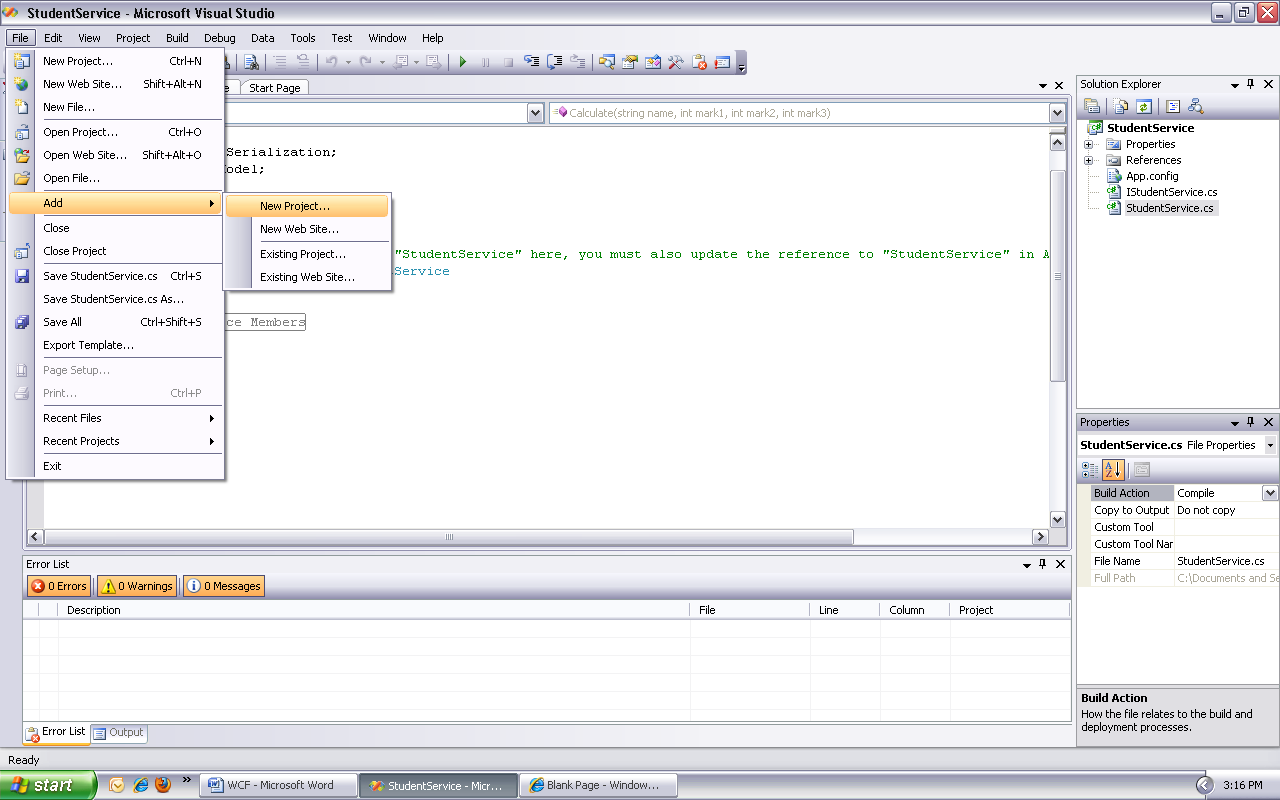
#endregion

}

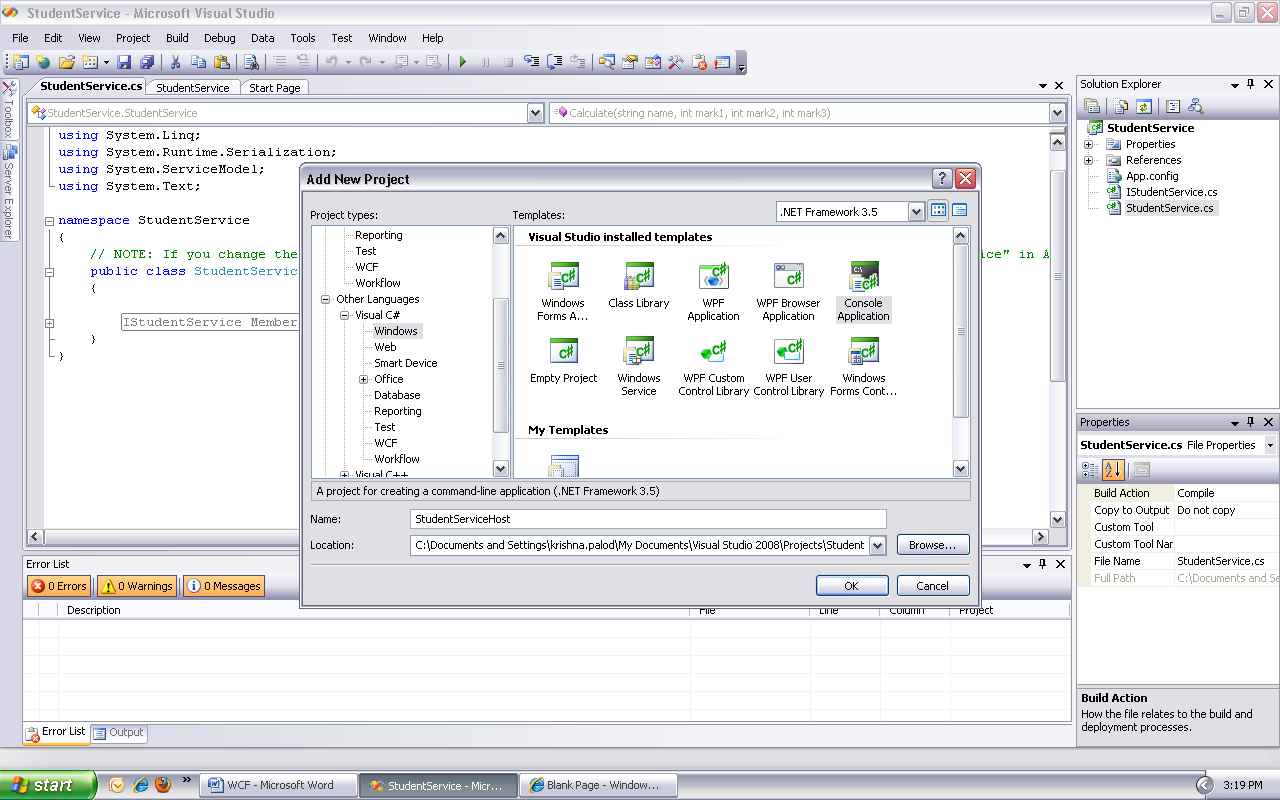
}

**Hosting the WCF service using console application**

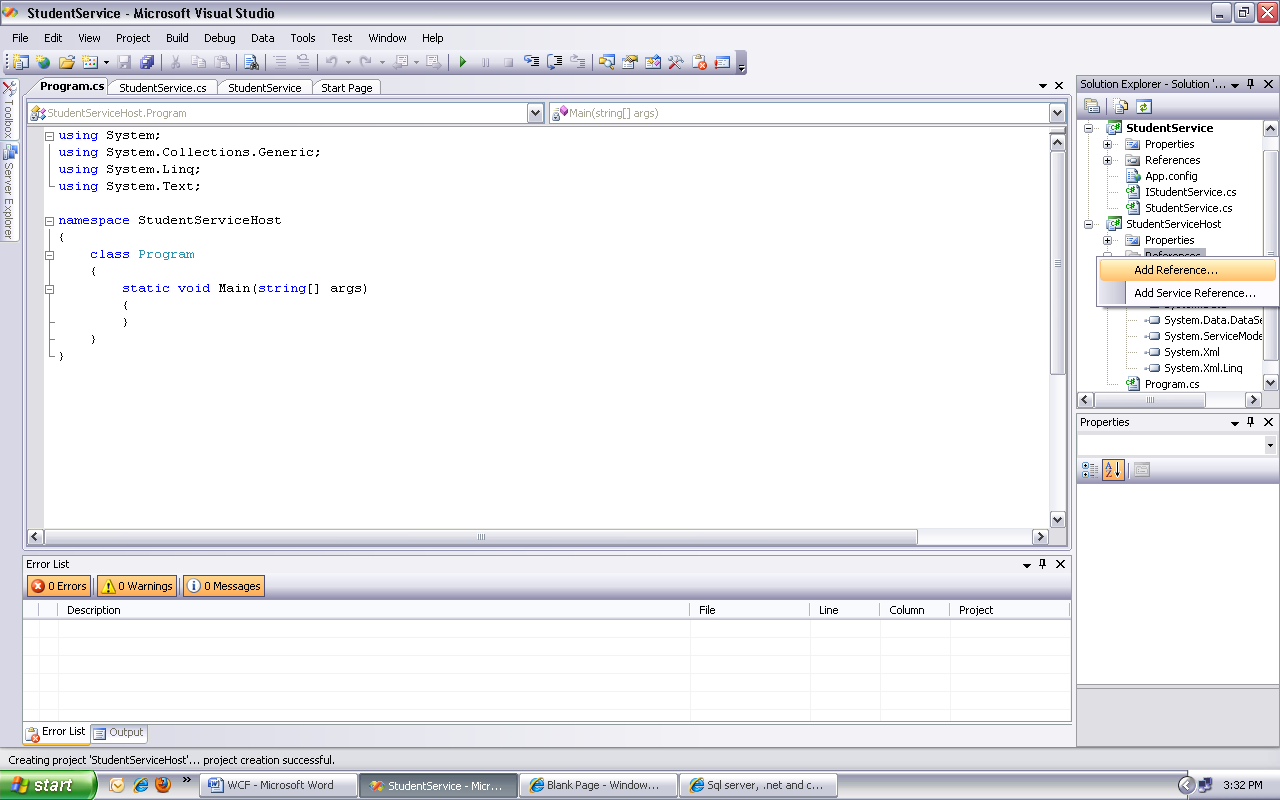
Click on file- add-new project. Select new console Applicaotn and name it StudentServiceHost.

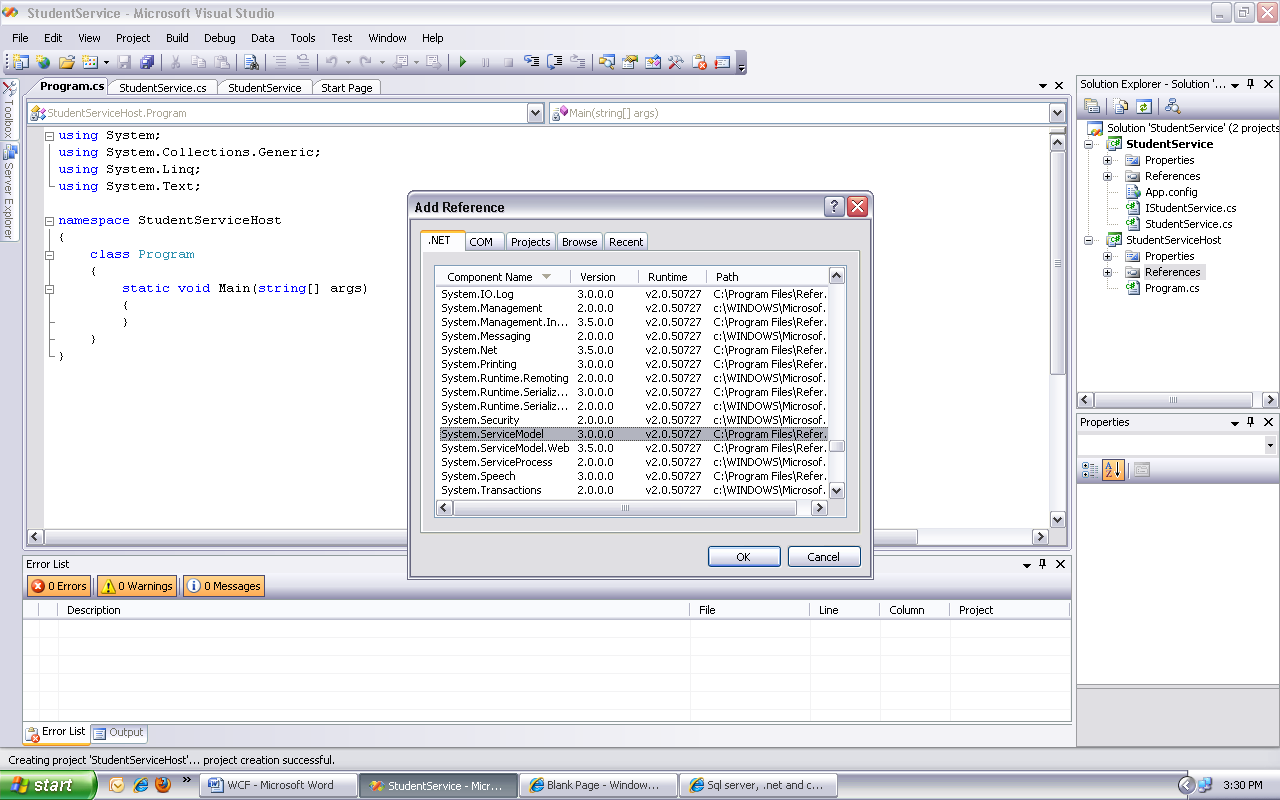


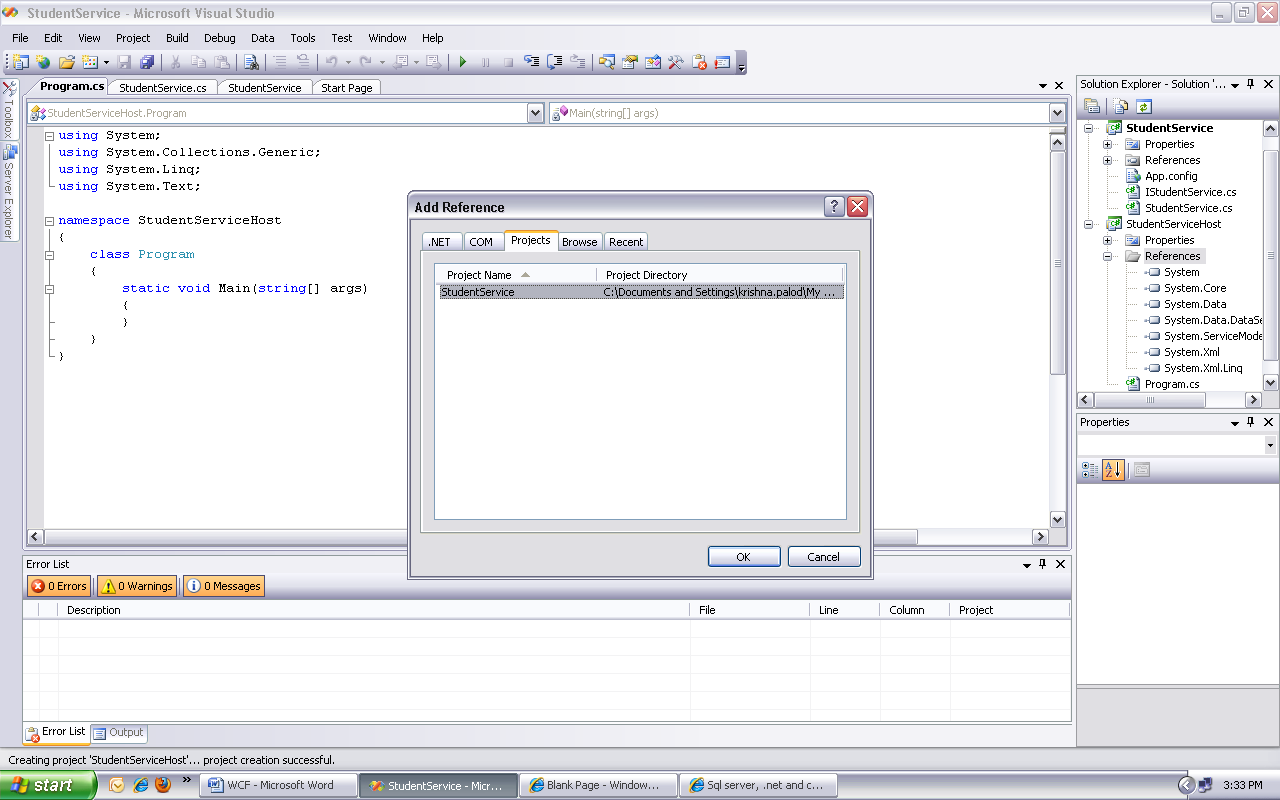
Select new console Application and name it StudentServiceHost.



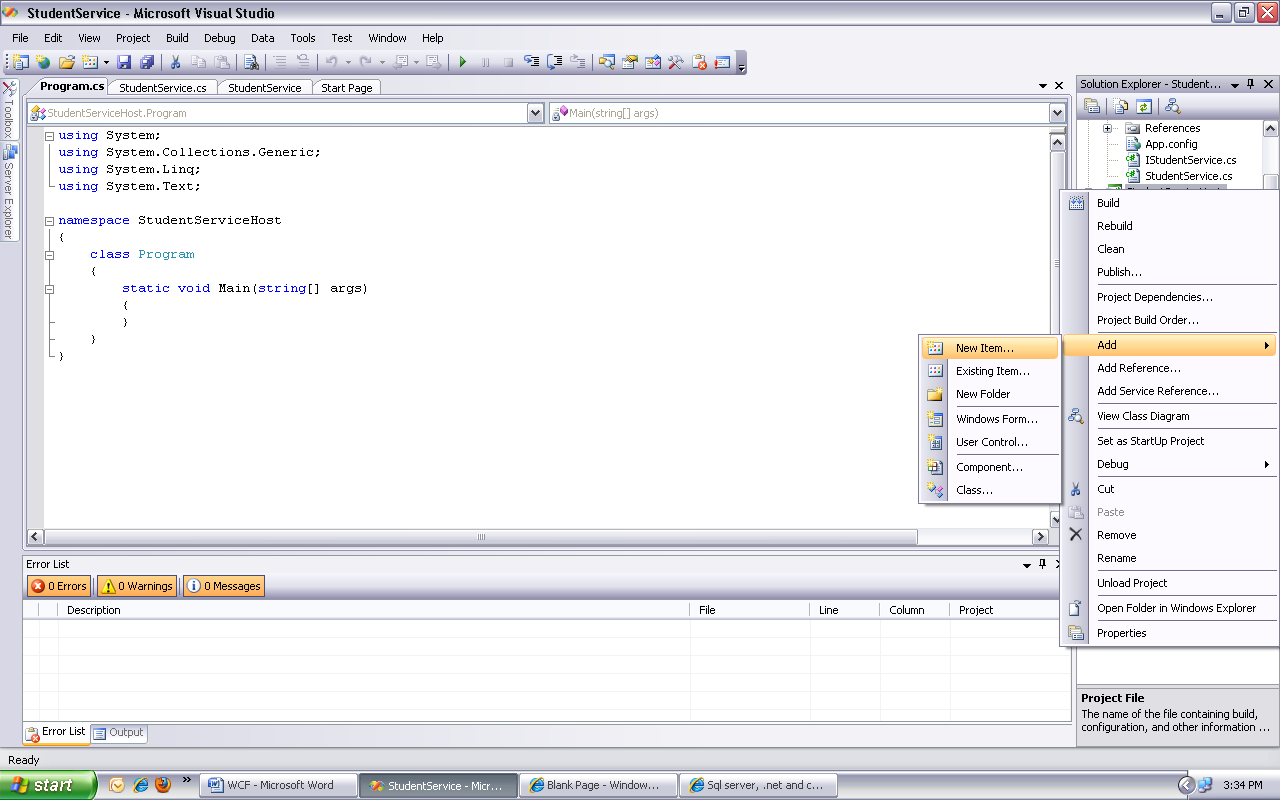
Add a reference to **System.ServiceModel** assembly and **StudentService** project



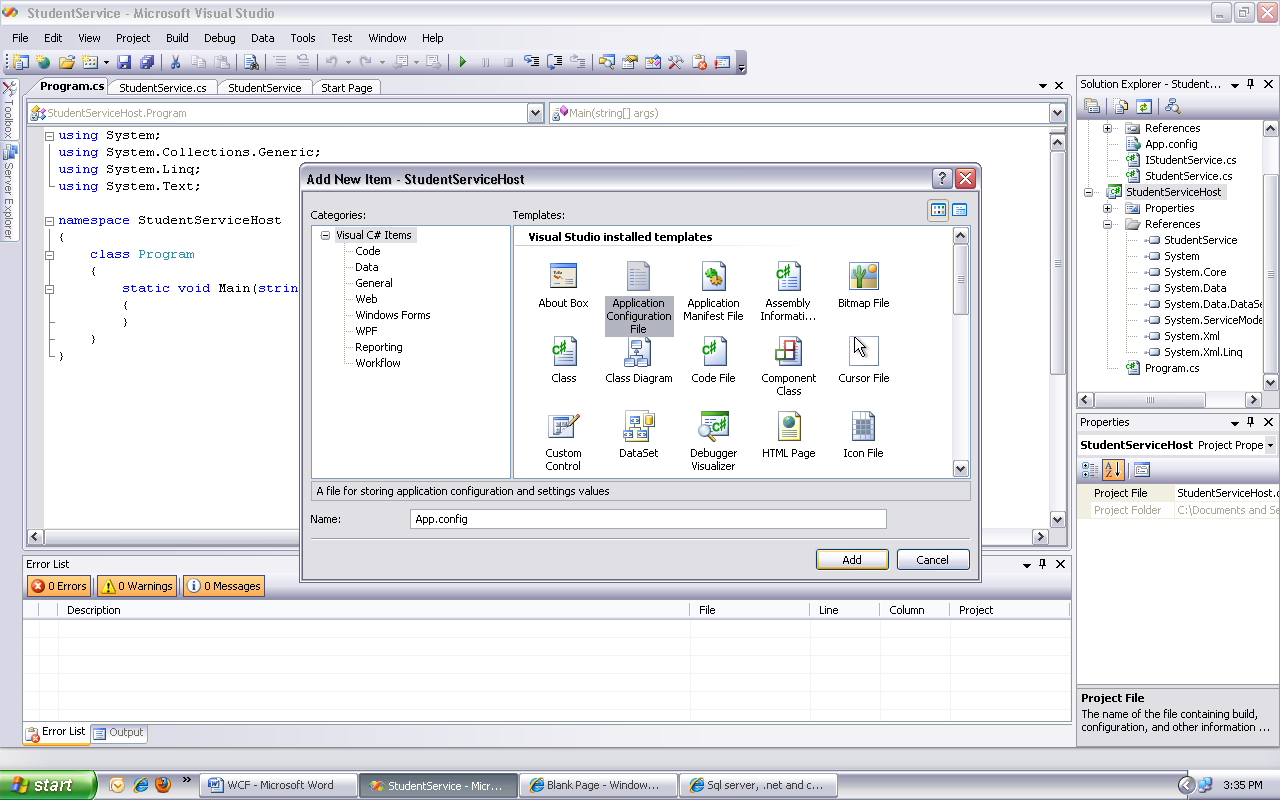




Right click on **HelloServiceHost** project and add **Application Configuration File.**



This should add **App.config** file to the project.



Copy and paste the following XML. Notice that we have specified **2 endpoints** in the configuration. One endpoint uses **basicHttpBinding,** which communicates over **HTTP** protocol using **XML** messages. This endpoint will satisfy the requirement of the first client. The other endpoint uses **netTcpBinding,** which communicates over **TCP** protocol using **binary messages.** This endpoint will satisfy the requirement of the second client.

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

<configuration>

<system.serviceModel>

<behaviors>

<serviceBehaviors>

<behavior name="mexBehaviour">

<serviceMetadata httpGetEnabled="true" />

</behavior>

</serviceBehaviors>

</behaviors>

<services>

<service name="StudentService.StudentService" behaviorConfiguration="mexBehaviour">

<endpoint address="StudentService" binding="basicHttpBinding" contract="StudentService.IStudentService">

</endpoint>

<endpoint address="StudentService" binding="netTcpBinding" contract="StudentService.IStudentService">

</endpoint>

<endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />

<host>

<baseAddresses>

<add baseAddress="http://localhost:8080/" />

<add baseAddress="net.tcp://localhost:8090"/>

</baseAddresses>

</host>

</service>

</services>

</system.serviceModel>

</configuration>

Copy and paste the following code in **Program.cs** file

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace StudentServiceHost

{

class Program

{

static void Main(string[] args)

{

using (System.ServiceModel.ServiceHost host = new

System.ServiceModel.ServiceHost(typeof(StudentService.StudentService)))

{

host.Open();

Console.WriteLine("Host started @ " + DateTime.Now.ToString());

Console.ReadLine();

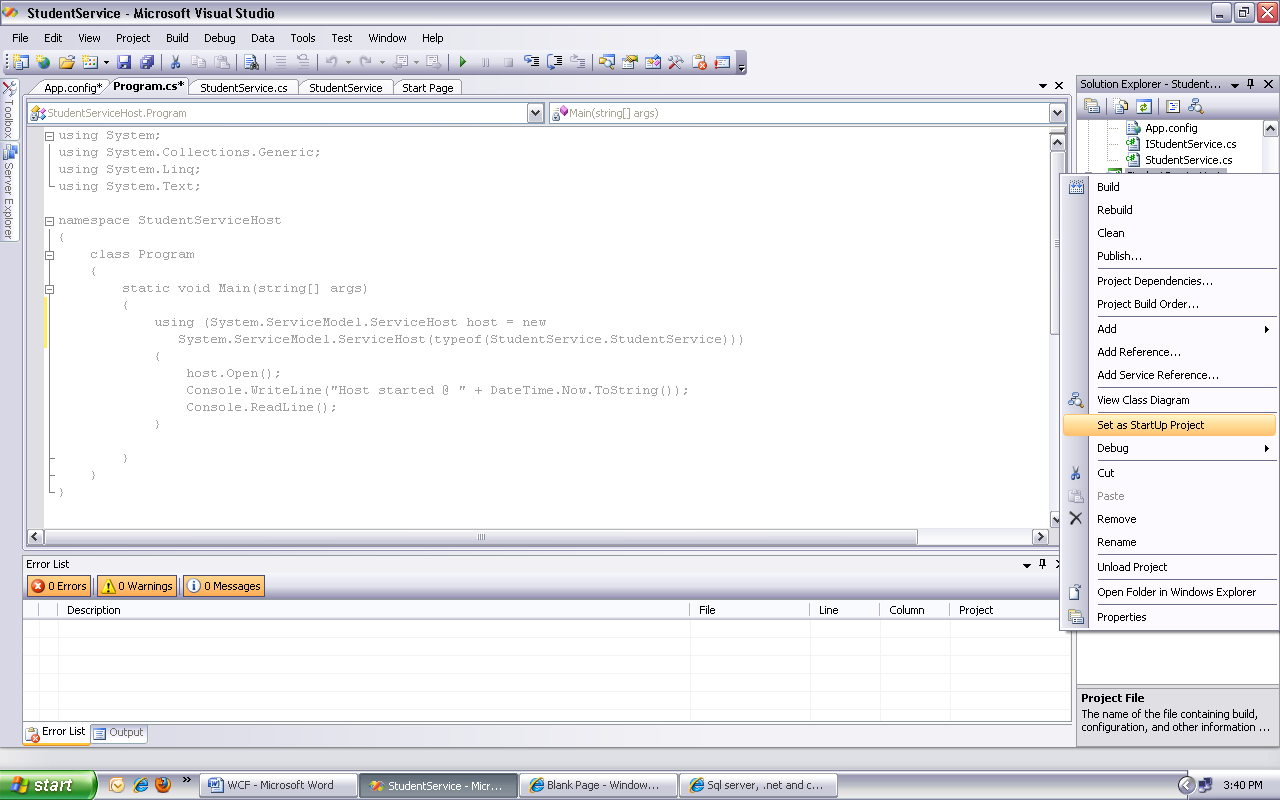
}

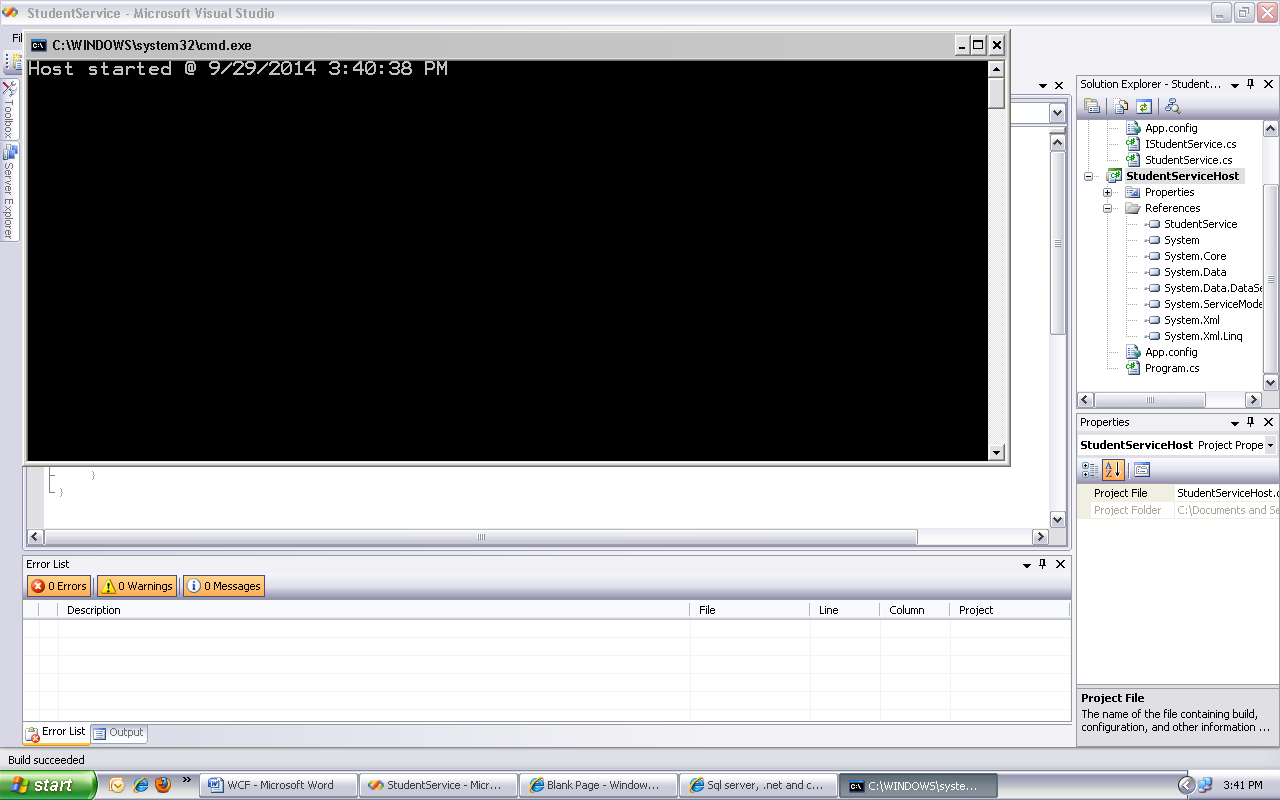
}

}

}

Build the solution. Set **StudentServiceHost** as startup project and run it by pressing **CTRL + F5** keys.



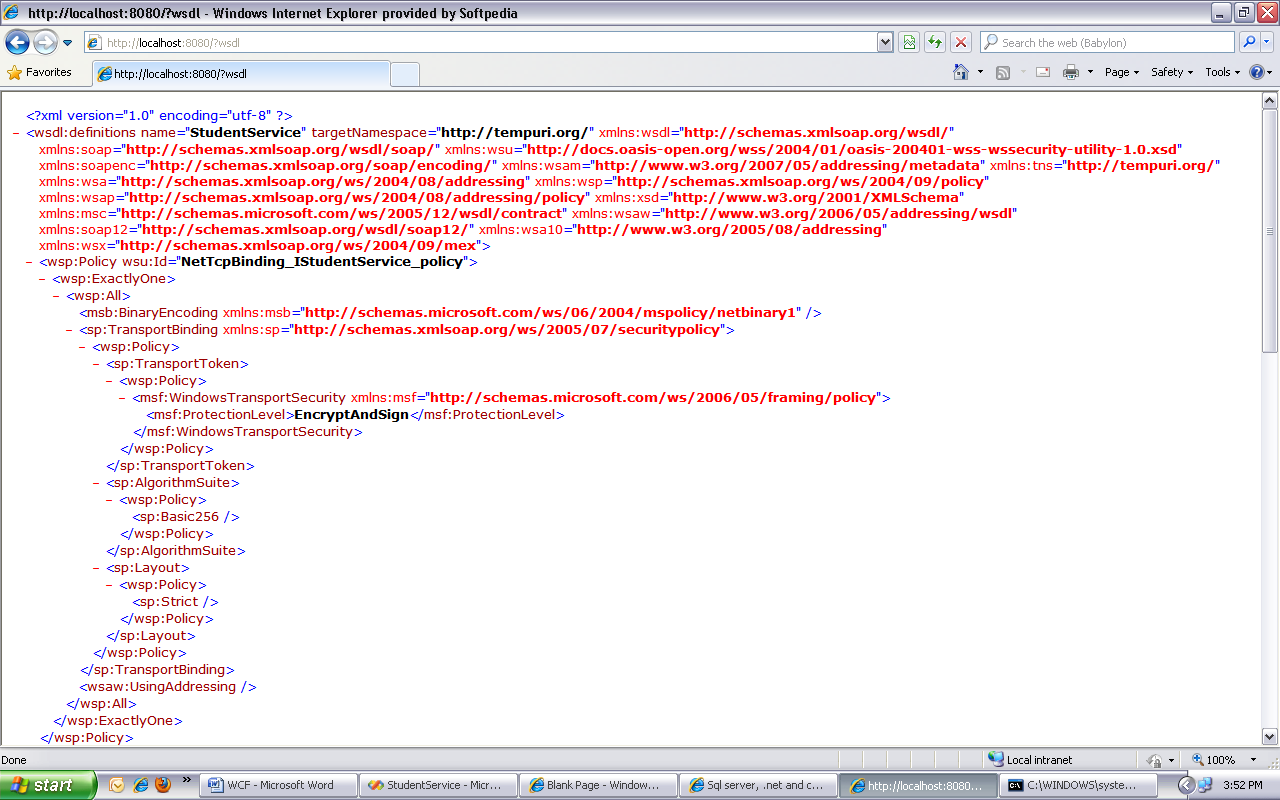


To get the WSDL document for the service copy base address from App.config file i.e. <http://localhost:8080/>

Open web browser and paste it to



Click on WSDL link to get wsdl document

Copy the wsdl url to add the reference for client

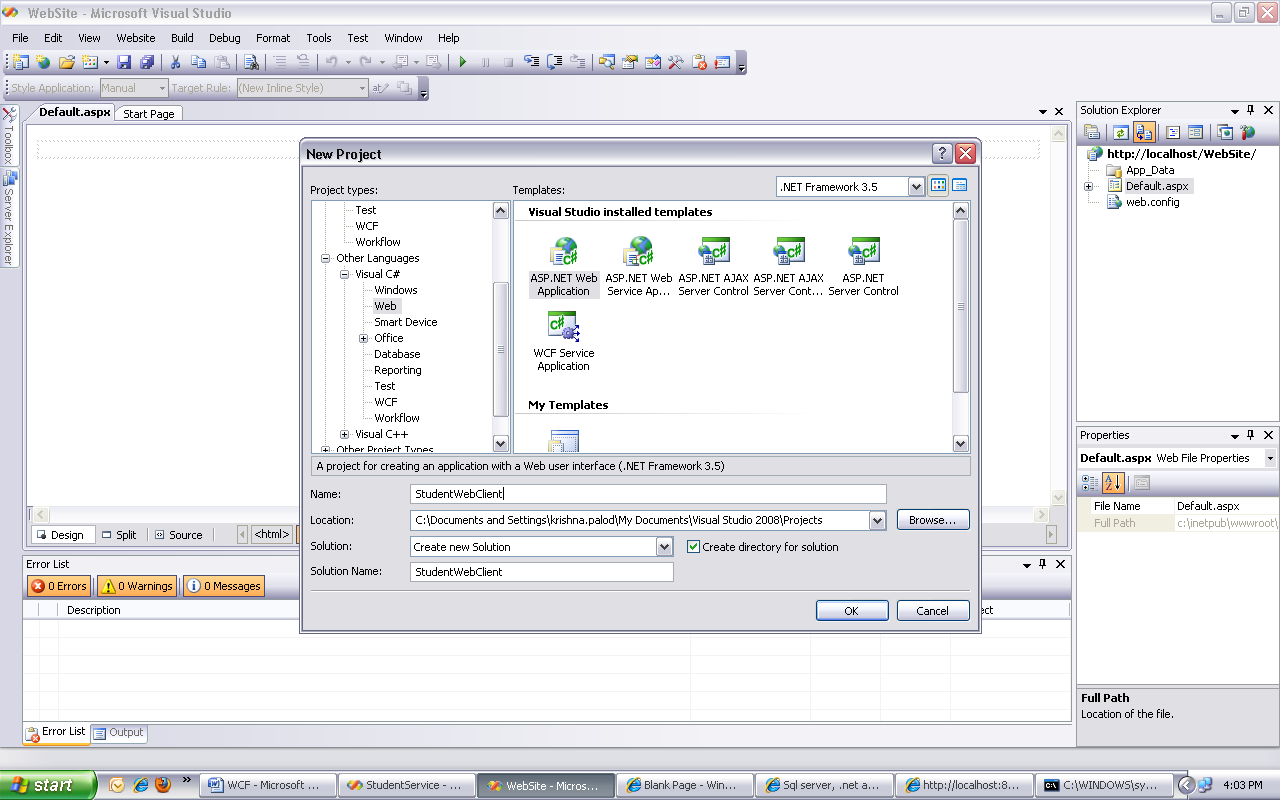
**WCF Web Client**

Now let's build a web application that is going to **consume the WCF service** using the endpoint with **basicHttpBinding**. basicHttpBinding communicates over HTTP protocol using XML messages.

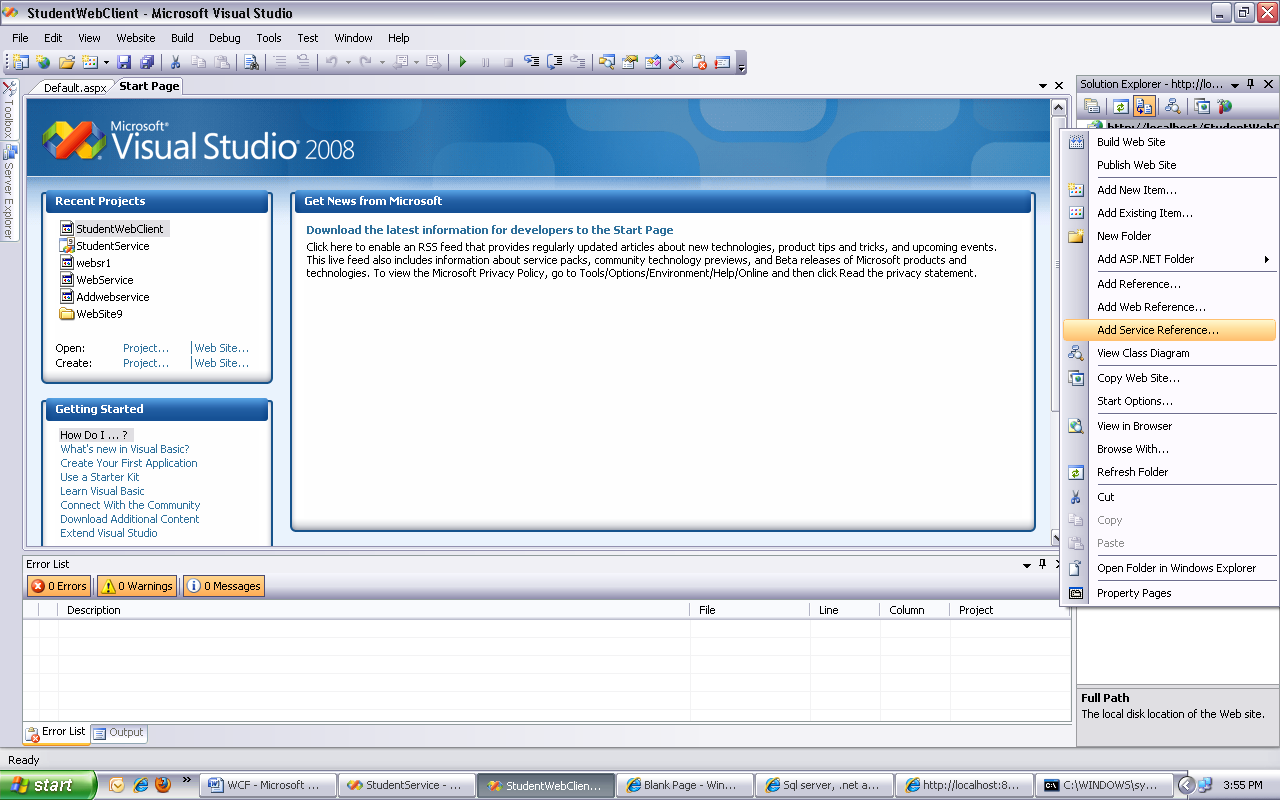
Create a new instance of Visual Studio.

Create a **new asp.net empty web application** and name it **StudentWebClient**

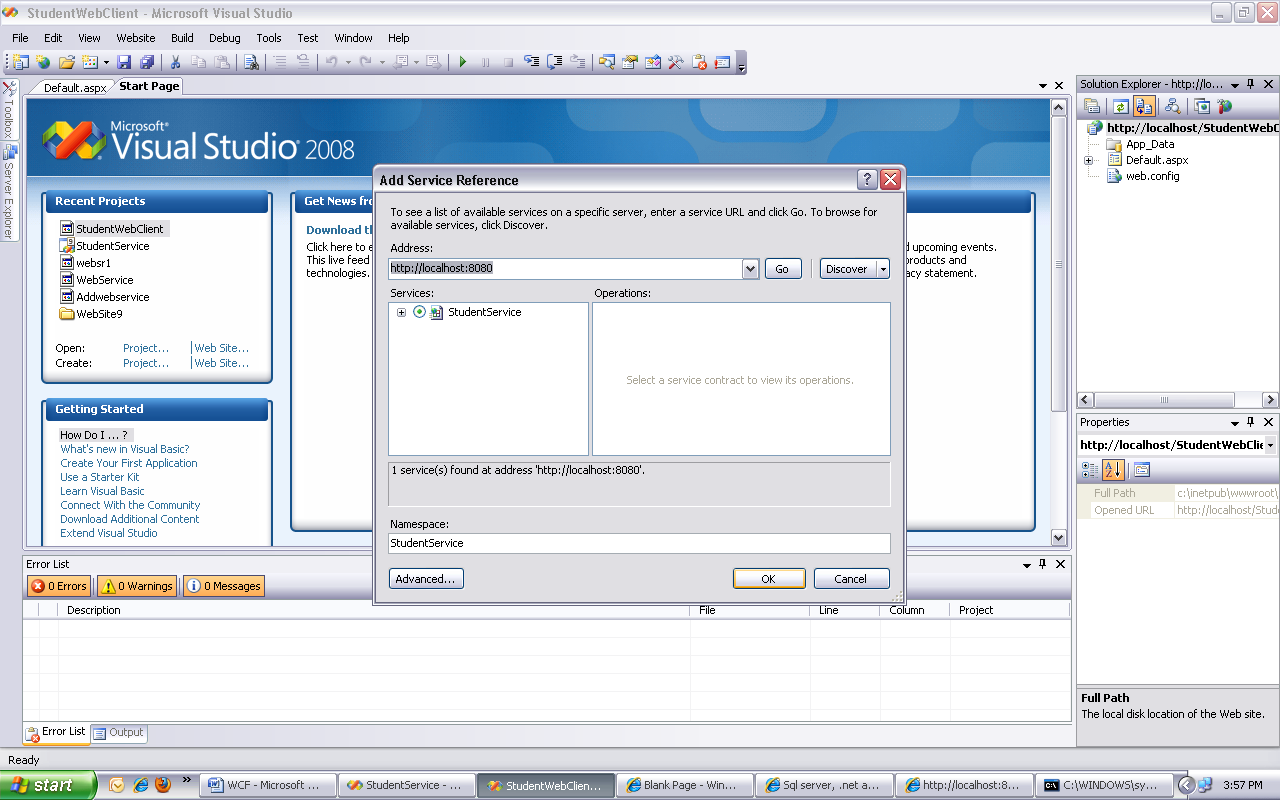
**Click on file new project ASP.net web application**



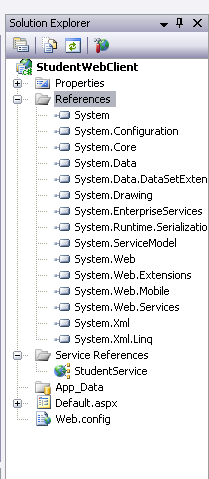
Right click on **References** folder and select **Add Service Reference** option.



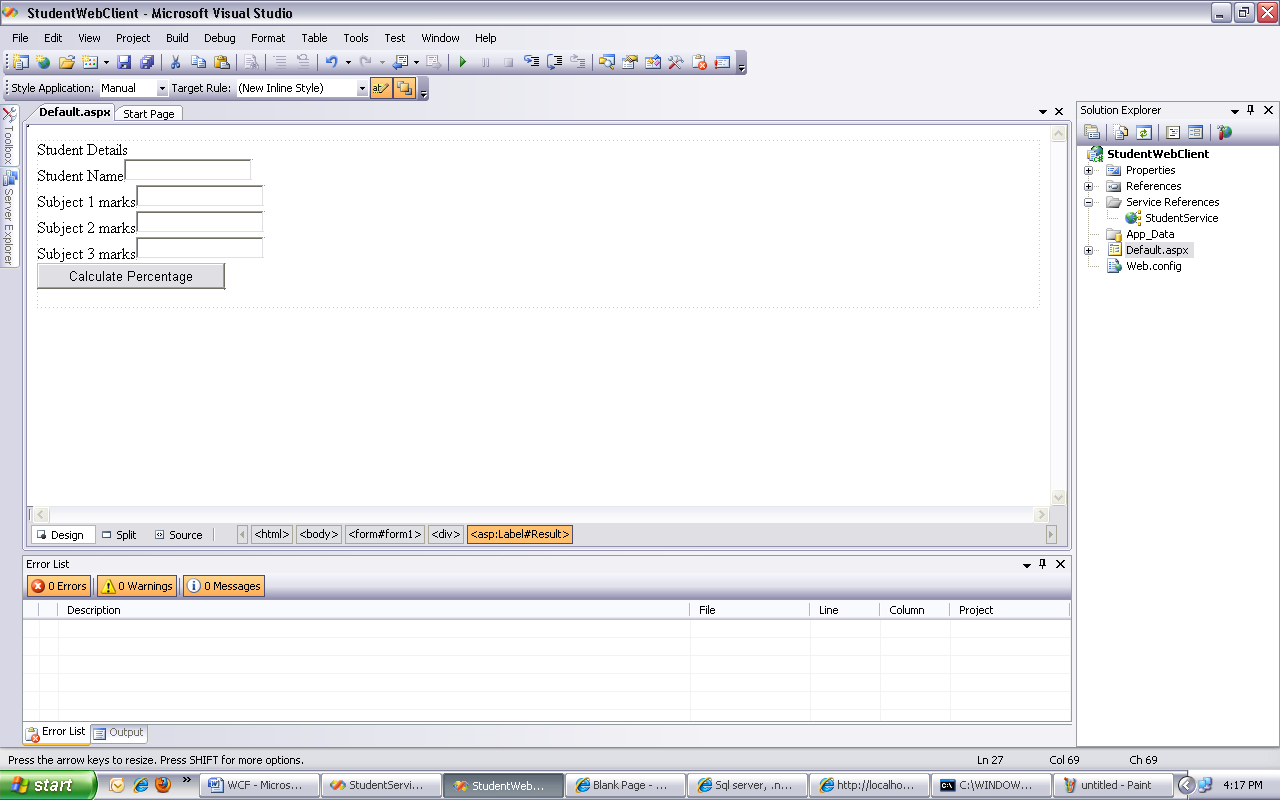
In the address textbox type **http://localhost:8080/** and click on **GO** button. In the namespace textbox type **StudentService** and click **OK.**



This should generate a proxy class to communicate with the service.



Create a GUI which has label, textbox to read the data from the client in default .aspx using controls in toolbox



The code for web form is

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="StudentWebClient.\_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="Student\_Details" runat="server" Text="Student Details"></asp:Label><br />

<asp:Label ID="name" runat="server" Text="Student Name"></asp:Label>

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

<asp:Label ID="M1" runat="server" Text="Subject 1 marks"></asp:Label>

<asp:TextBox ID="txtM1" runat="server"></asp:TextBox><br />

<asp:Label ID="M2" runat="server" Text="Subject 2 marks"></asp:Label>

<asp:TextBox ID="txtM2" runat="server"></asp:TextBox><br />

<asp:Label ID="M3" runat="server" Text="Subject 3 marks"></asp:Label>

<asp:TextBox ID="txtM3" runat="server"></asp:TextBox>

<br />

<asp:Button ID="Button1" runat="server" Text="Calculate Percentage"

onclick="Button1\_Click" />

<br />

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

</div>

</form>

</body>

</html>

Copy and paste the following code in default.aspx.cs file

protected void Button1\_Click(object sender, EventArgs e)

{

StudentService.StudentServiceClient client = new

StudentService.StudentServiceClient("BasicHttpBinding\_IStudentService");

int m1 = int.Parse(txtM1.Text);

int m2 = int.Parse(txtM2.Text);

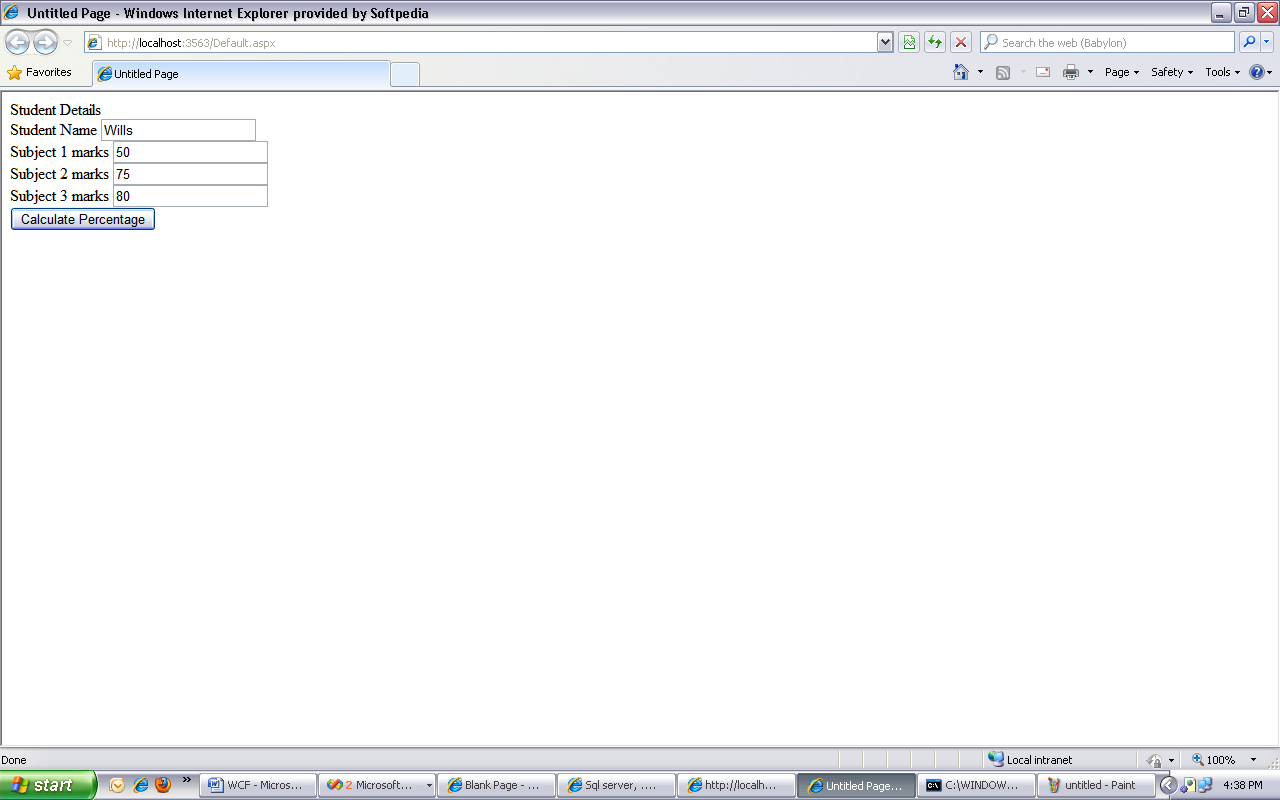
int m3 = int.Parse(txtM3.Text);

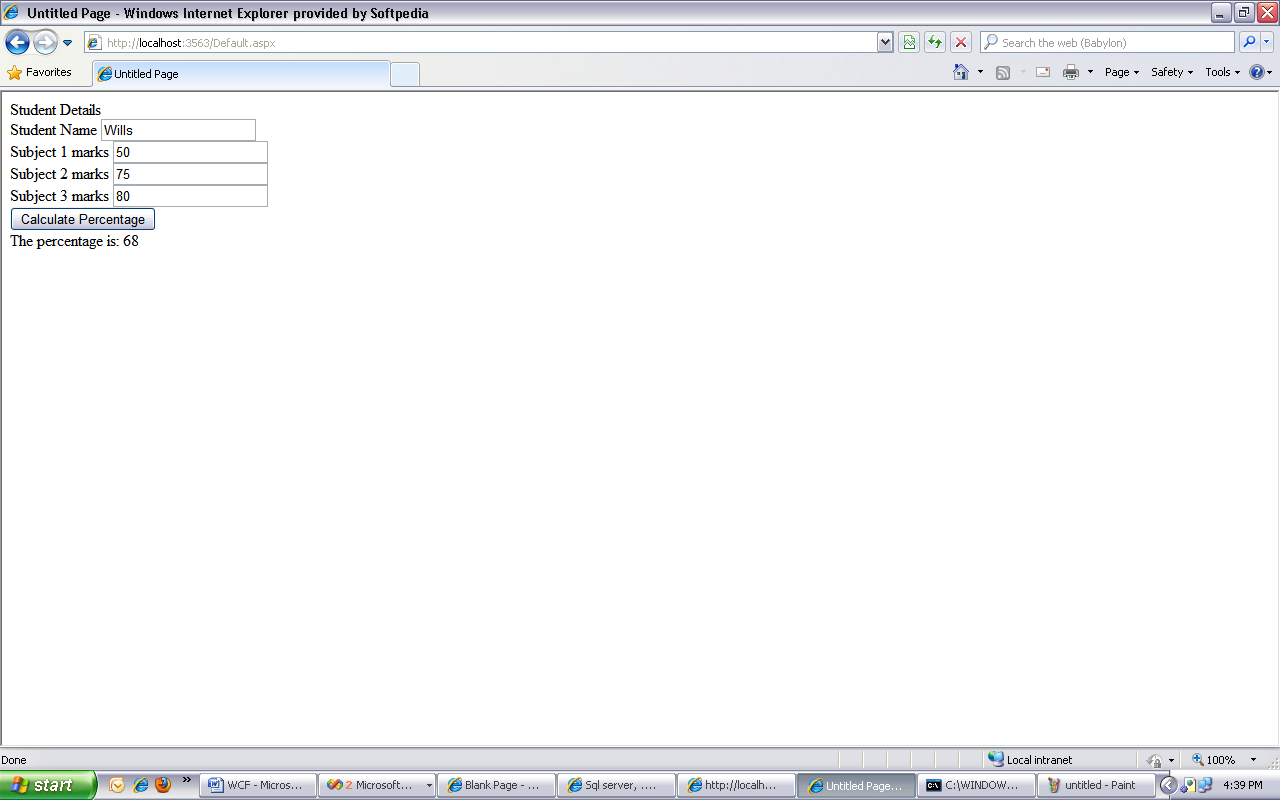
String name = txtname.Text;

Result.Text= client.Calculate(name, m1, m2, m3);

}

Run the client



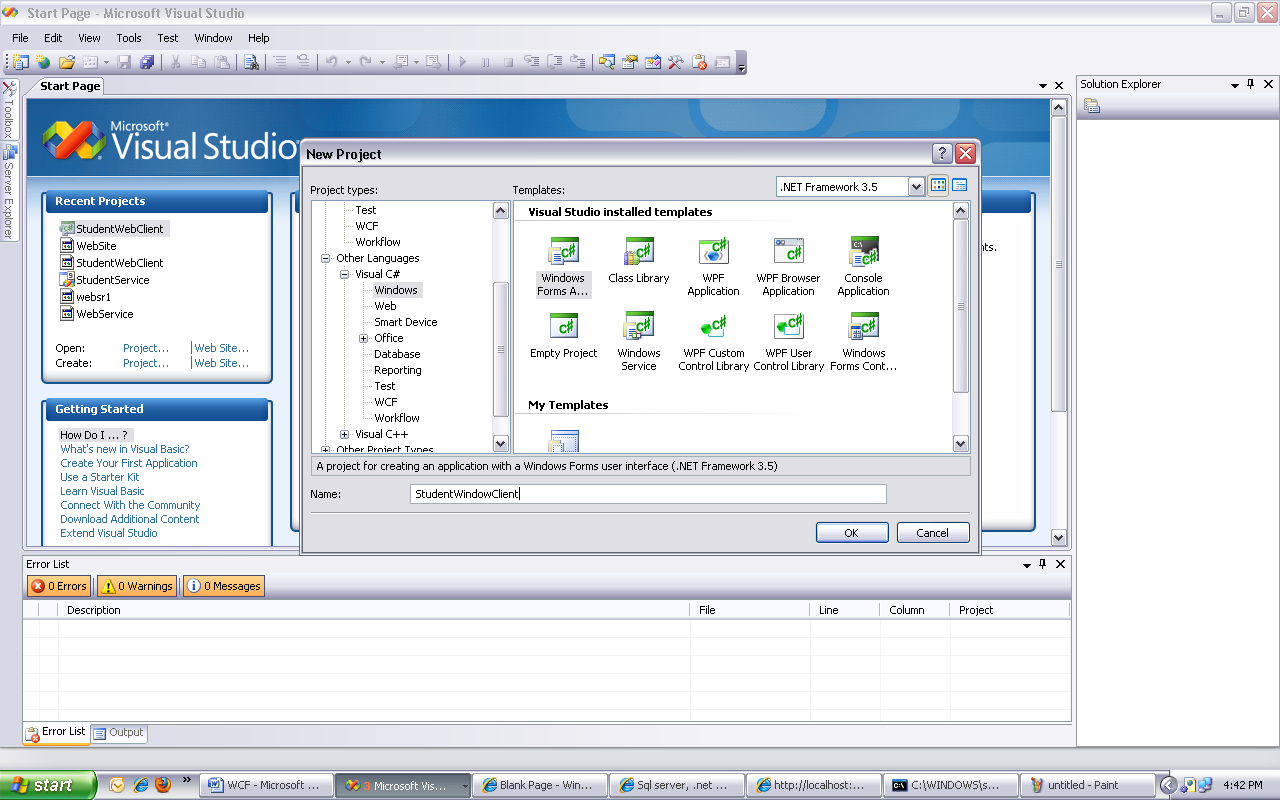


**WCF Window Client**

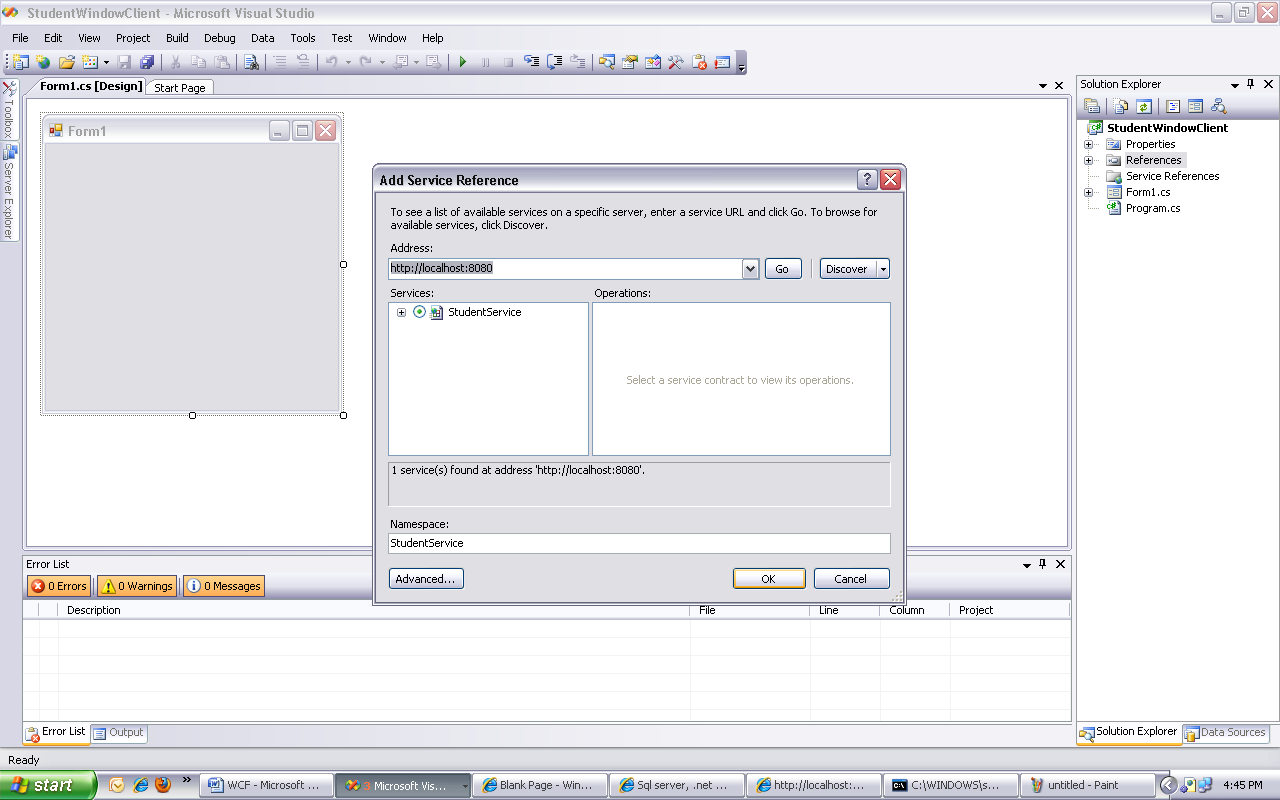
Now let's build a **windows application** that is going to **consume the WCF service** using the endpoint with **netTcpBinding.** netTcpBinding communicated over TCP protocol using binary messages.

Create a new instance of Visual Studio.

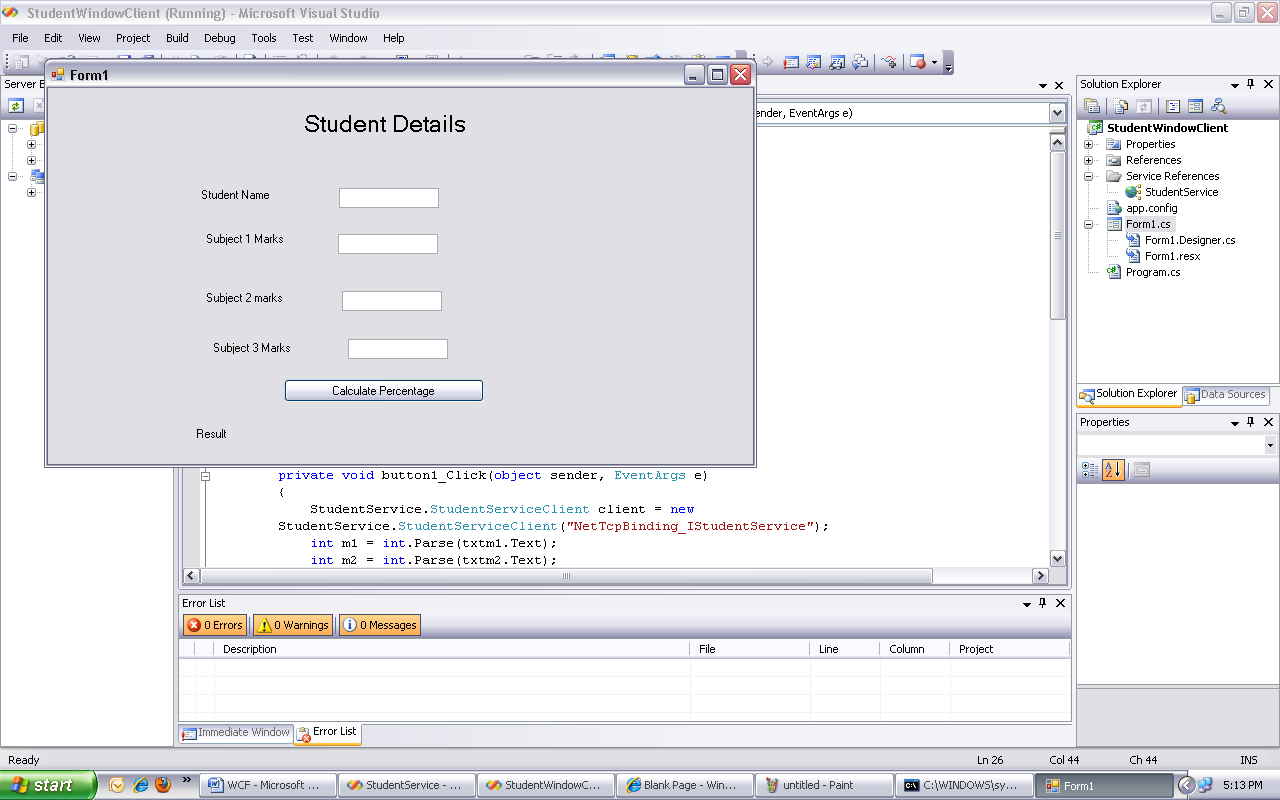
Create a new Windows Forms application by clicking on new project- window form application and named it StudentWindowClient.



Right click on **References** folder and select **Add Service Reference** option. In the address textbox type **http://localhost:8080/** and click on **GO** button. In the namespace textbox type **StudentService** and click **OK.** This should generate a proxy class to communicate with the service



On **Form1**, drag and drop a textboxs, a button and a labels control to design GUI.



Double click the button to generate the click event handler.

**4.** Copy and paste the following code in Form1.cs file

private void button1\_Click(object sender, EventArgs e)

{

StudentService.StudentServiceClient client = new

StudentService.StudentServiceClient("NetTcpBinding\_IStudentService");

int m1 = int.Parse(txtm1.Text);

int m2 = int.Parse(txtm2.Text);

int m3 = int.Parse(txtm3.Text);

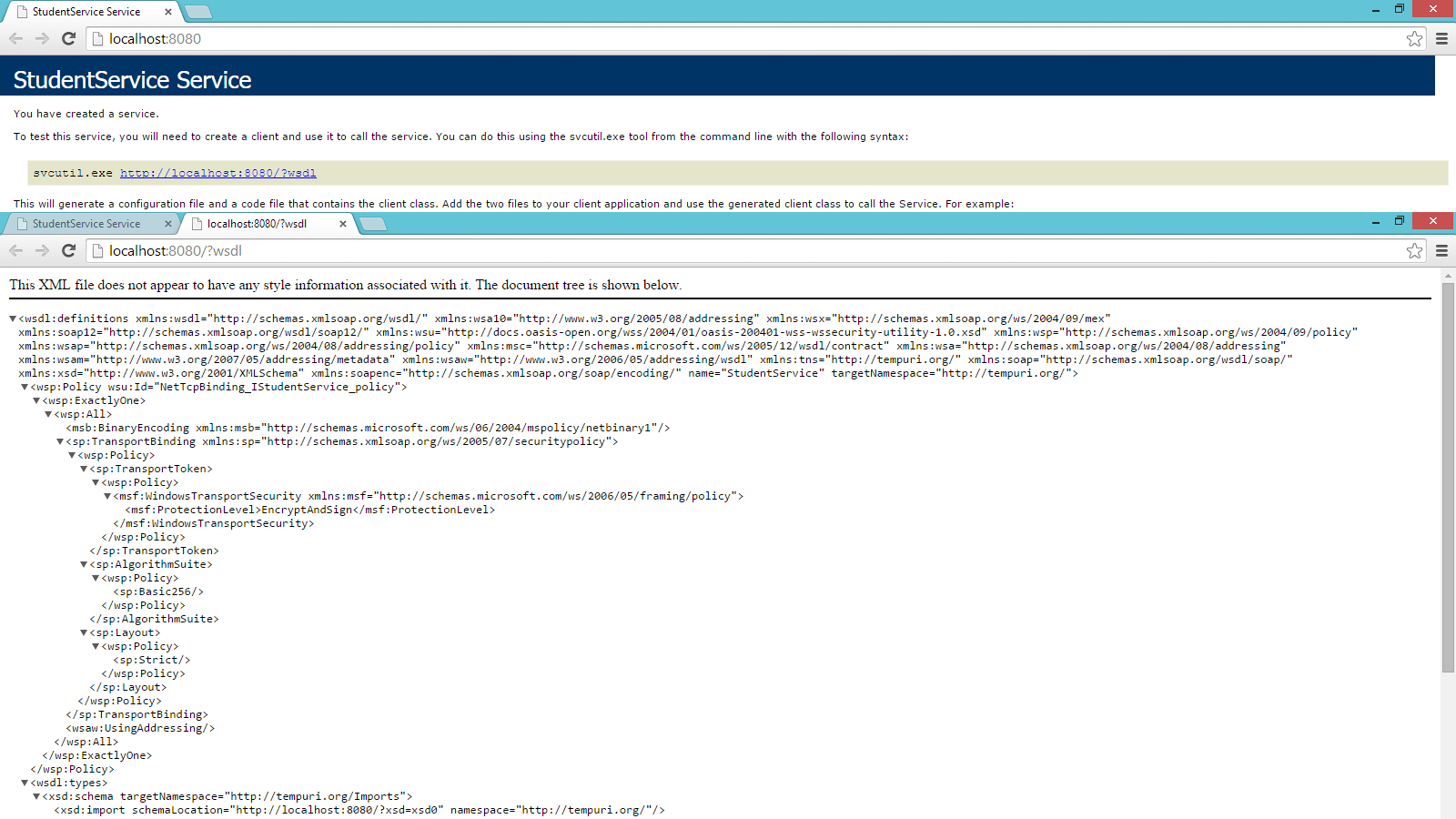
String name = txtname.Text;

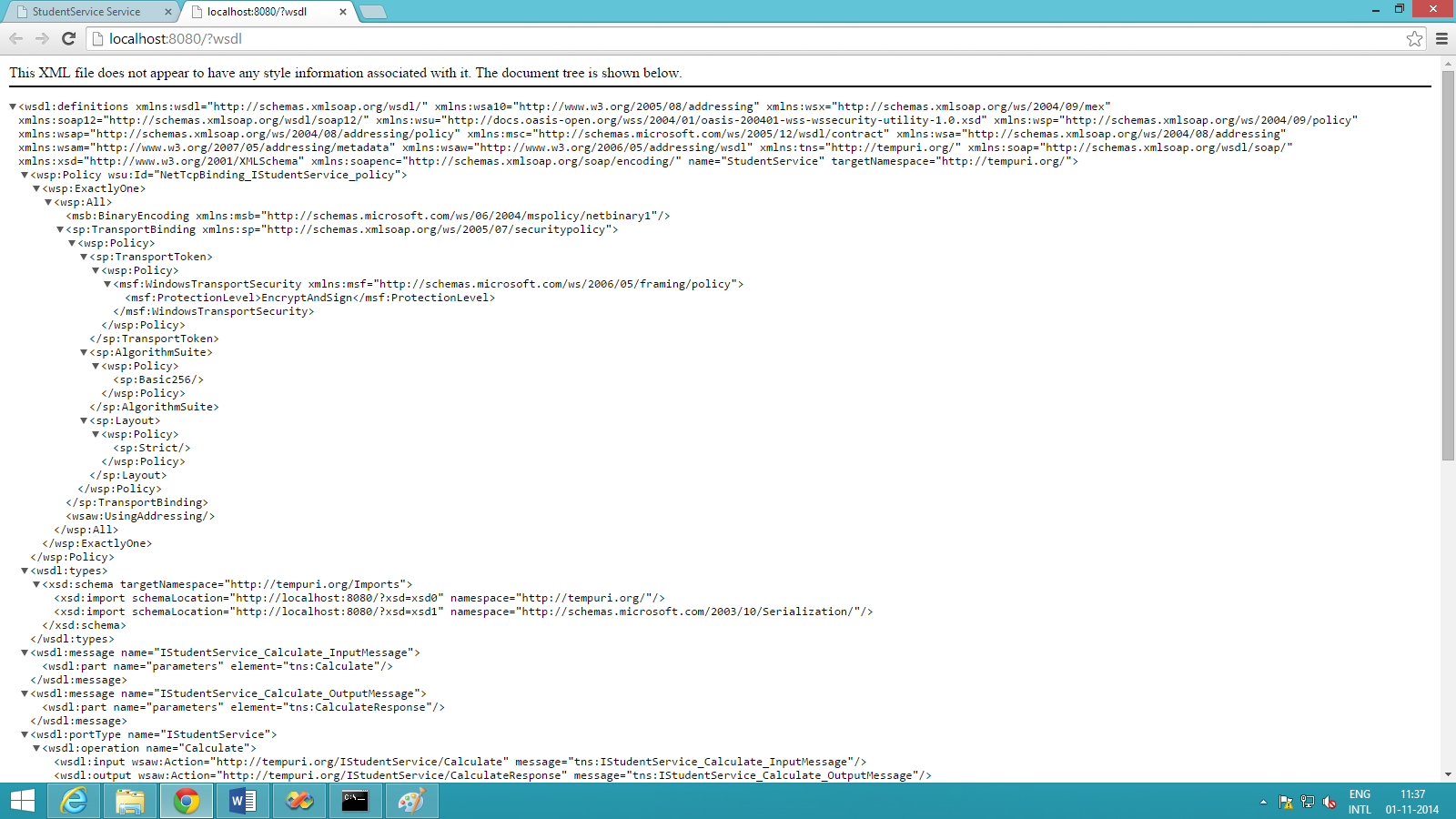
Result.Text = client.Calculate(name,m1, m2, m3);

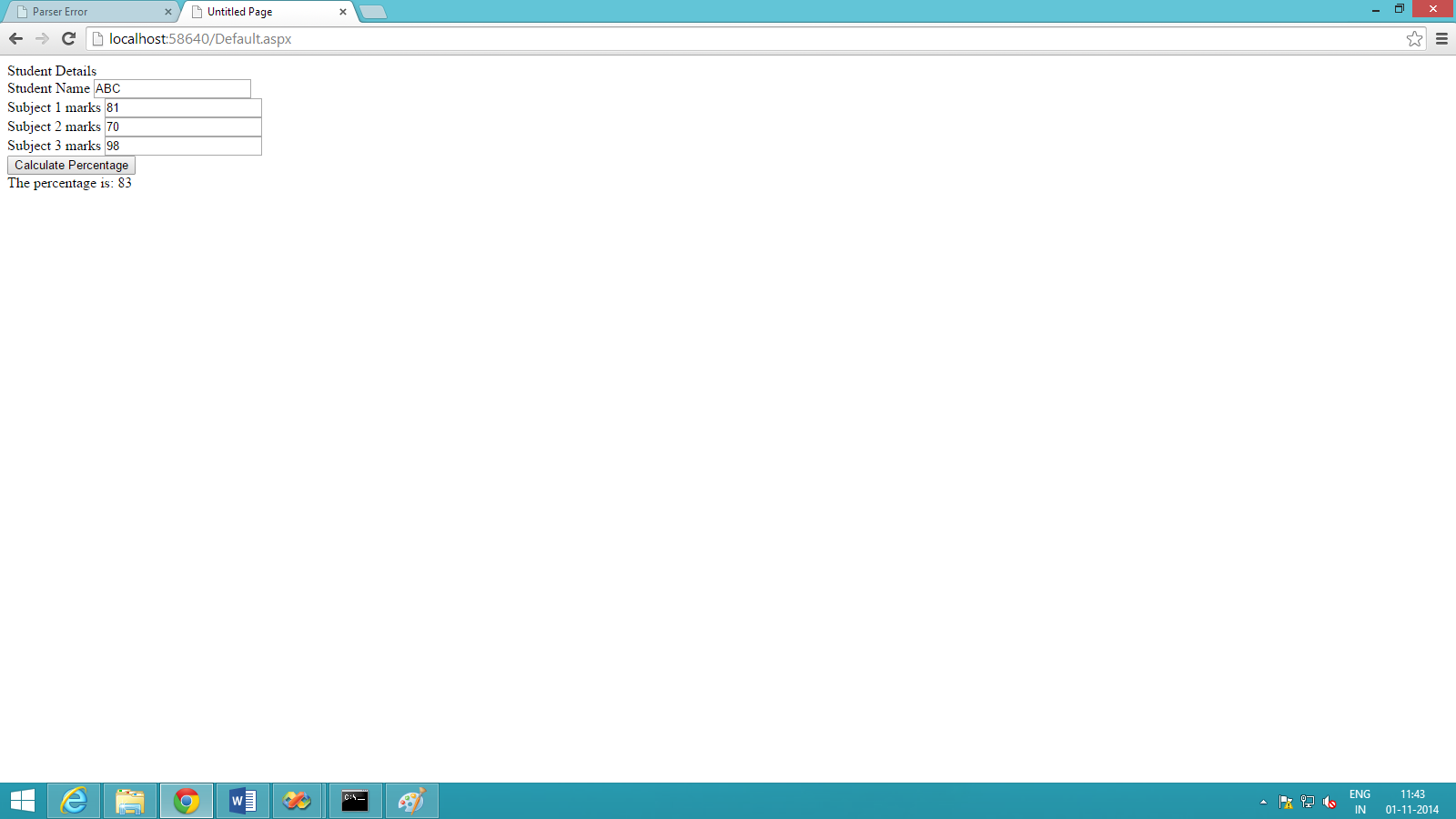
}

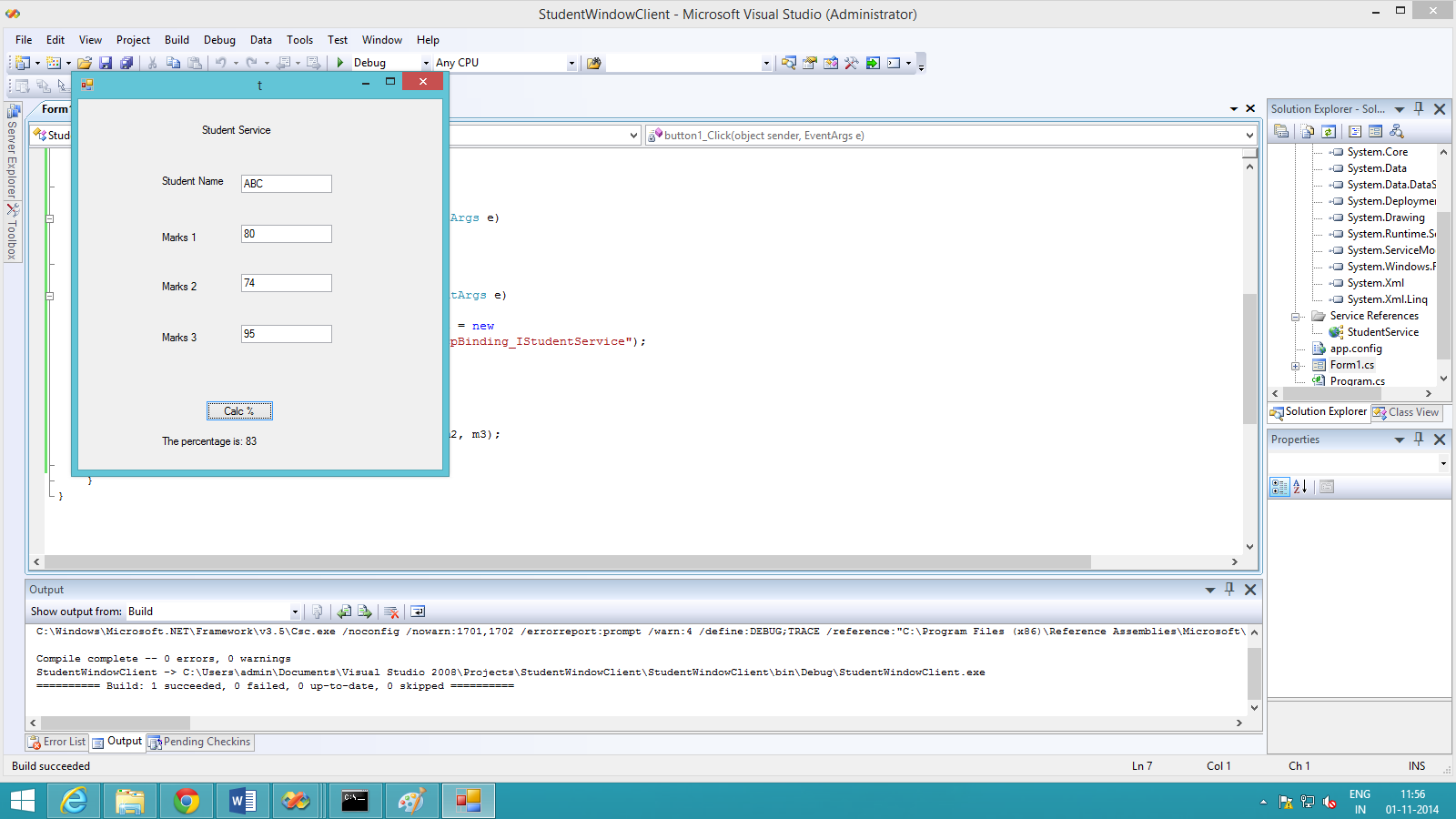
**Run project to see the output.**

**Output Screen Shots**









**Conclusion:**

Thus we have studied and implemented a client web service and are able to identify within the application the need for accessing a web service and create a client web service to access the online web service.