| **Name of Student:** Ajay Karthikesan | | | |
| --- | --- | --- | --- |
| **Roll Number:** 57 | | **Assignment Number:** 9 | |
| **Aim of Assignment:**   1. Design Web Application to produce and consume a web Service. Write an Application that uses these services. 2. Design Web Application to produce and consume a web Service. | | | |
| **DOP:** | | **DOS:** 3.623 | |
| **CO Mapped:**  CO4 | **PO Mapped:**  PO3, PO5, PSO1, PSO2 | **Faculty Signature:** | **Marks:** |

## 

## Practical No. 9

**Aim:**

1. Design Web Application to produce and consume a web Service. Write an Application that uses these services.
2. Design Web Application to produce and consume a web Service.

**Theory:**

Web Services:

A Web Service is a software program that uses XML to

exchange information with other software via common internet

protocols. In a simple sense, Web Services are a way of

interacting with objects over the Internet.

A web service is

• Language Independent.

• Protocol Independent.

• Platform Independent.

• It assumes a stateless service architecture.

• Scalable (e.g. multiplying two numbers together to an entire

customerrelationship management system).

• Programmable (encapsulates a task).

• Based on XML (open, text-based standard).

• Self-describing (metadata for access and use).

• Discoverable (search and locate in registries)- ability of

applications and developers to search for and locate desired Web

services through registries. This is based on UDDI.

Web services advantages:-

• Use open, text-based standards, which enable components

written in various languages and for different platforms to

communicate.

• Promote a modular approach to programming, so multiple

organizations can communicate with the same Web service.

• Comparatively easy and inexpensive to implement, because

they employ an existing infrastructure and because most

applications can be repackaged as Web services.

• Significantly reduce the costs of enterprise application (EAI)

integration and B2B communications.

• Implemented incrementally, rather than all at once which

lessens the cost and reduces the organizational disruption from

an abrupt switch in technologies.

• The Web Services Interoperability Organization (WS-I)

consisting of over 100 vendors promotes interoperability.

Web Services Limitations :-

• SOAP, WSDL, UDDI- require further development.

• Interoperability.

• Royalty fees.

• Too slow for use in high-performance situations.

• Increase traffic on networks.

• The lack of security standards for Web services.

• The standard procedure for describing the quality (i.e. levels of

performance, reliability, security etc.) of particular Web services

– management of Web services.

• The standards that drive Web services are still in draft form

(always will be in refinement).

• Some vendors want to retain their intellectual property rights

to certain Web services standards.

Web Service Example:-

A web service can perform almost any kind of task.

• Web Portal- A web portal might obtain top news headlines

from an Associated press web service.

• Weather Reporting- You can use Weather Reporting web

service to display weather information in your personal website.

• Stock Quote- You can display latest update of Share market

with Stock Quote on your web site.

• News Headline-You can display latest news update by using

News Headline Web Service in your website.

Web Services using http:

If the web service uses an HTTPS connection, then this will

communicate using SSL. SSL, or Secure Socket Layer, is a

technology which allows the web service client and remote web

service server to communicate over a secured connection.

The data is encrypted before being transmitted and decrypted

when the data is received before processing. This is a two-way

process, meaning that both the server AND the browser encrypt

all traffic before sending out data. A certificate is required for

SSL communication.

Web Service in ASP.NET:

A Web Service is programmable application logic accessible via

standard Web protocols. One of these Web protocols is the

Simple Object Access Protocol (SOAP). SOAP is a W3C

submitted note (as of May 2000) that uses standards based

technologies (XML for data description and HTTP for transport)

to encode and transmit application data.

Consumers of a Web Service do not need to know anything

about the platform, object model, or programming language

used to implement the service; they only need to understand

how to send and receive SOAP messages (HTTP and XML).

WCF Service:

Windows Communication Foundation (WCF) is a framework

for building serviceoriented applications. Using WCF, you can

send data as asynchronous messages from one service endpoint

to another. A service endpoint can be part of a continuously

available service hosted by IIS, or it can be a service hosted in

an application. An endpoint can be a client of a service that

requests data from a service endpoint. The messages can be as

simple as a single character or word sent as XML, or as complex

as a stream of binary data.

In what scenarios must WCF be used

• A secure service to process business transactions.

• A service that supplies current data to others, such as a traffic

report or other monitoring service.

• A chat service that allows two people to communicate or

exchange data in real time.

• A dashboard application that polls one or more services for

data and presents it in a logical presentation.

• Exposing a workflow implemented using Windows Workflow

Foundation as a WCF service.

• A Silverlight application to poll a service for the latest data

feeds.

Features of WCF:

• Service Orientation

• Interoperability

• Multiple Message Patterns

• Service Metadata

• Data Contracts

• Security

• Multiple Transports and Encodings

• Reliable and Queued Messages

• Durable Messages

• Transactions

• AJAX and REST Support

• Extensibility

**Code:**

File: WebForm1.aspx

﻿<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Practical9.WebForm1" %>

<!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:GridView ID="GridView1" runat="server">

</asp:GridView>

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Fetch Data" />

</div>

</form>

</body>

</html>

File: WebForm1.aspx.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical9

{

public partial class WebForm1 : System.Web.UI.Page

{

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

{

}

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

{

WebService1 db = new WebService1();

GridView1.DataSource = db.Get();

GridView1.DataBind();

}

}

}

File: WebForm2.aspx

﻿<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="Practical9.WebForm2" %>

<!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:GridView ID="GridView1" runat="server">

</asp:GridView>

<br />

<b>Enter Game Information:-</b><br />

ID&nbsp;

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

<br />

Name&nbsp;

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

<br />

<asp:CheckBox ID="outdoorChkBx" runat="server" Text="Is Outdoor?" />

<br />

Player count&nbsp;

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

<br />

Status&nbsp;

<asp:Label ID="statusLbl" runat="server">\_\_\_\_\_\_\_</asp:Label>

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Insert" />

</div>

</form>

</body>

</html>

File: WebForm2.aspx.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical9

{

public partial class WebForm2 : System.Web.UI.Page

{

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

{

WebService1 obj1 = new WebService1();

GridView1.DataSource = obj1.Get();

GridView1.DataBind();

}

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

{

int id = Int32.Parse(idTxt.Text);

String name = nameTxt.Text;

bool is\_outdoor = outdoorChkBx.Checked;

int pcount = Convert.ToInt32(pcountTxt.Text);

WebService1 obj1 = new WebService1();

obj1.AddData(name, id, is\_outdoor, pcount);

statusLbl.Text = "Success";

GridView1.DataSource = obj1.Get();

GridView1.DataBind();

}

}

}

File: WebService1.asmx.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

using System.Data.SqlClient;

using System.Data;

namespace Practical9

{

/// <summary>

/// Summary description for WebService1

/// </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 WebService1 : System.Web.Services.WebService

{

private SqlConnection sqlcon = new SqlConnection("Data Source=SNAKEJAZZ\\SQLEXPRESS;Initial Catalog=coll;Integrated Security=True");

[WebMethod]

public string HelloWorld()

{

return "Hello World";

}

public DataTable Get()

{

using (SqlCommand cmd = new SqlCommand("SELECT \* FROM ajayk57\_games"))

{

using (SqlDataAdapter sda = new SqlDataAdapter())

{

cmd.Connection = sqlcon;

sda.SelectCommand = cmd;

DataTable dt = new DataTable();

dt.TableName = "Games";

sda.Fill(dt);

return dt;

}

}

}

[WebMethod]

public void AddData(String name, int id, bool is\_outdoor, int pcount)

{

SqlCommand cmd = new SqlCommand("INSERT INTO ajayk57\_games(id, name, is\_outdoor, pcount) VALUES( @id,@Name, @is\_outdoor, @pcount)");

cmd.Parameters.AddWithValue("@name", name);

cmd.Parameters.AddWithValue("@id", id);

cmd.Parameters.AddWithValue("@is\_outdoor", is\_outdoor);

cmd.Parameters.AddWithValue("@pcount", pcount);

cmd.Connection = sqlcon;

sqlcon.Open();

cmd.ExecuteNonQuery();

sqlcon.Close();

}

}

}

File: Form1.cs

﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Practical9B

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

Service1 obj = new

Service1();

resultLbl.Text = obj.Addition(Convert.ToInt32(textBox1.Text),

Convert.ToInt32(textBox2.Text)).ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

Service1 obj = new

Service1();

resultLbl.Text = obj.Subtraction(Convert.ToInt32(textBox1.Text),

Convert.ToInt32(textBox2.Text)).ToString();

}

private void button3\_Click(object sender, EventArgs e)

{

Service1 obj = new Service1();

resultLbl.Text =

obj.Multiplication(Convert.ToInt32(textBox1.Text),

Convert.ToInt32(textBox2.Text)).ToString();

}

private void button4\_Click(object sender, EventArgs e)

{

Service1 obj = new

Service1();

resultLbl.Text = obj.Division(Convert.ToInt32(textBox1.Text),

Convert.ToInt32(textBox2.Text)).ToString();

}

}

}

File: Service1.cs

﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Linq;

using System.ServiceProcess;

using System.Text;

namespace Practical9B

{

partial class Service1 : ServiceBase

{

public Service1()

{

InitializeComponent();

}

protected override void OnStart(string[] args)

{

// TODO: Add code here to start your service.

}

protected override void OnStop()

{

// TODO: Add code here to perform any tear-down necessary to stop your service.

}

public int Addition(int num1, int num2)

{

return num1 + num2;

}

public int Subtraction(int num1, int num2)

{

return num1 - num2;

}

public int Multiplication(int num1, int num2)

{

return num1 \* num2;

}

public int Division(int num1, int num2)

{

return num1 / num2;

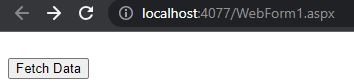
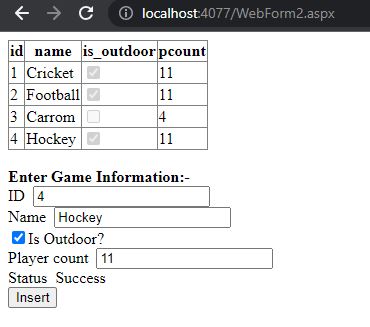
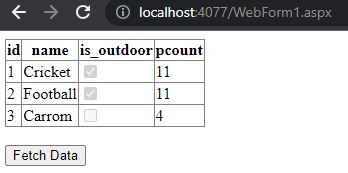
}

}

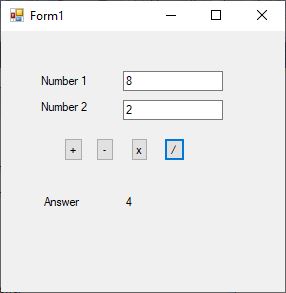
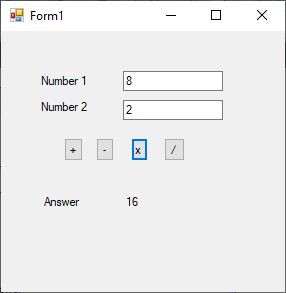
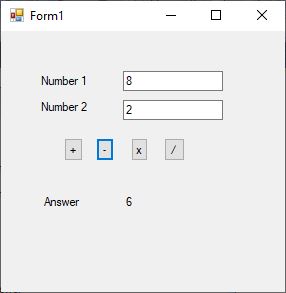
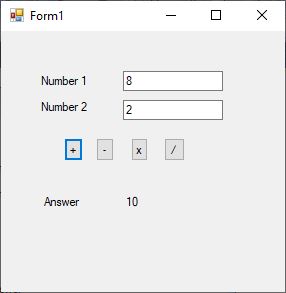
}

**Output:**

A.



B.



**Conclusion:**

I learnt how to create and use web services.