Lab Manual for Cloud Computing

Lab No. 7

ASP.NET Web Service Basics

LAB 07: ASP.NET WEB SERVICE BASICS

1. INTRODUCTION:

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 customer-relationship 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.*

Key Web Service Technologies

- **XML**- Describes only data. So, any application that understands XML-regardless of the application's programming language or platform has the ability to format XML in a variety of ways.
- **SOAP**- Provides a communication mechanism between services and applications.
- **WSDL**- Offers a uniform method of describing web services to other programs.
- **UDDI**- Enables the creation of searchable Web services registries.

When these technologies are deployed together, they allow developers to package applications as services and publish those services on a network.

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.
- You can make your own web service and let others use it. For example you can make Free SMS Sending Service with footer with your companies' advertisement, so whosoever uses this service indirectly advertises your company. You can apply your ideas in N no. of ways to take advantage of it.

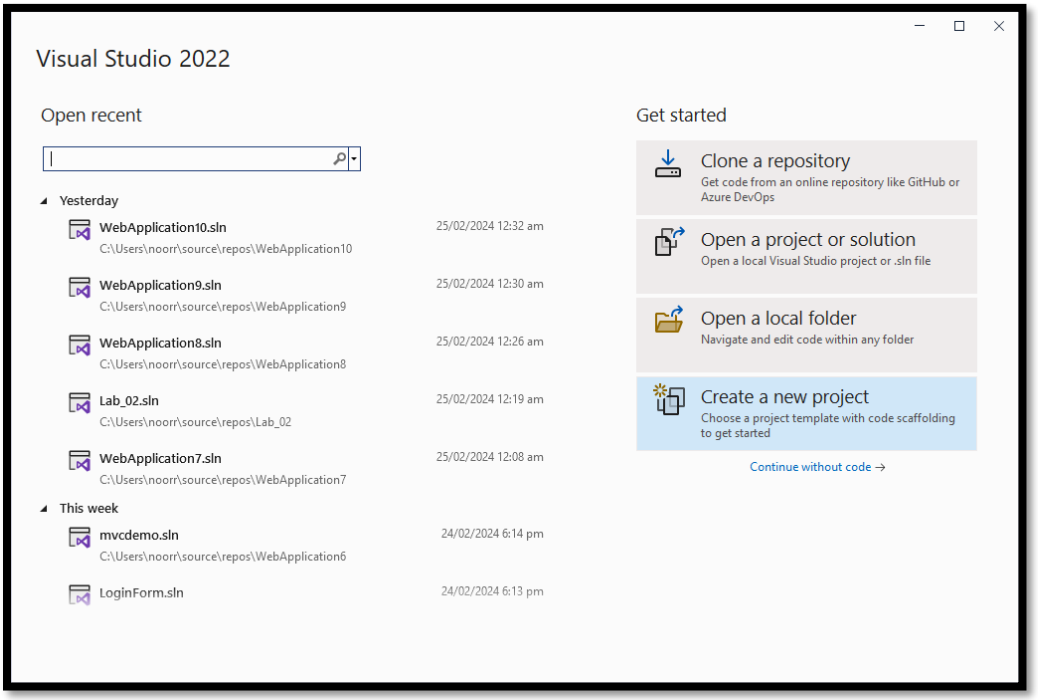
Example of Creating Web Service in .Net

Following is an example of creating simple web service of calculator.

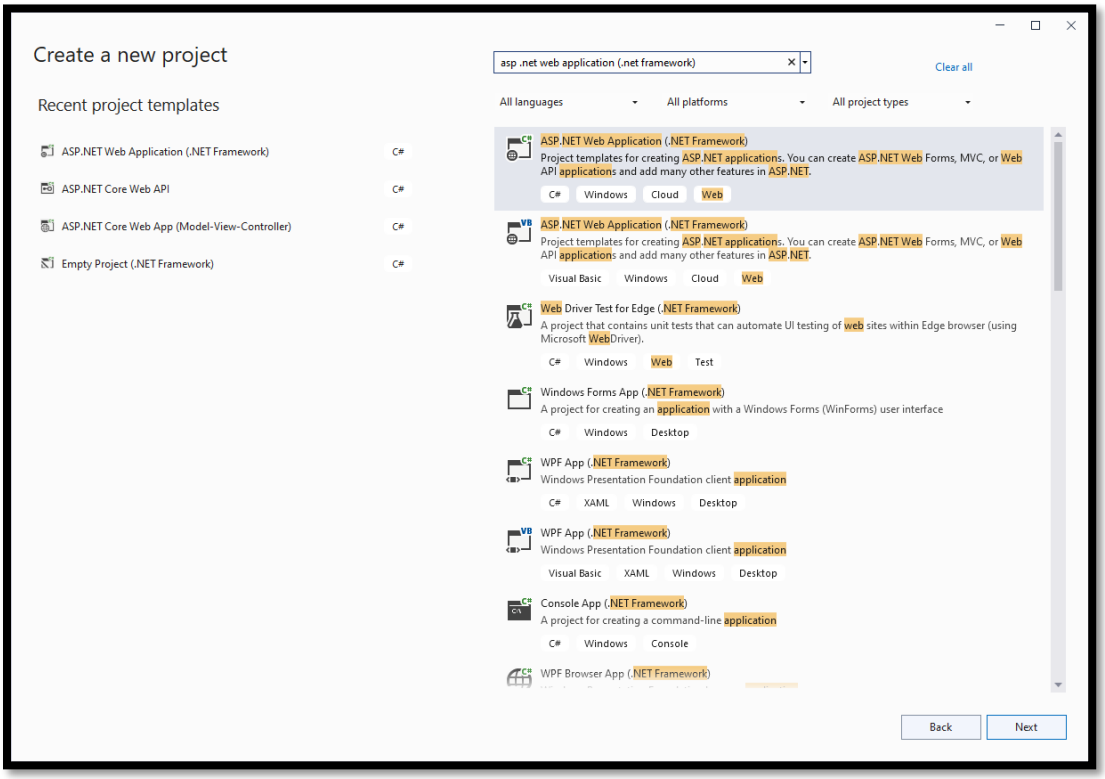
Step 1: Create a new project in Visual Studio 2022

- Open Visual Studio 2022.

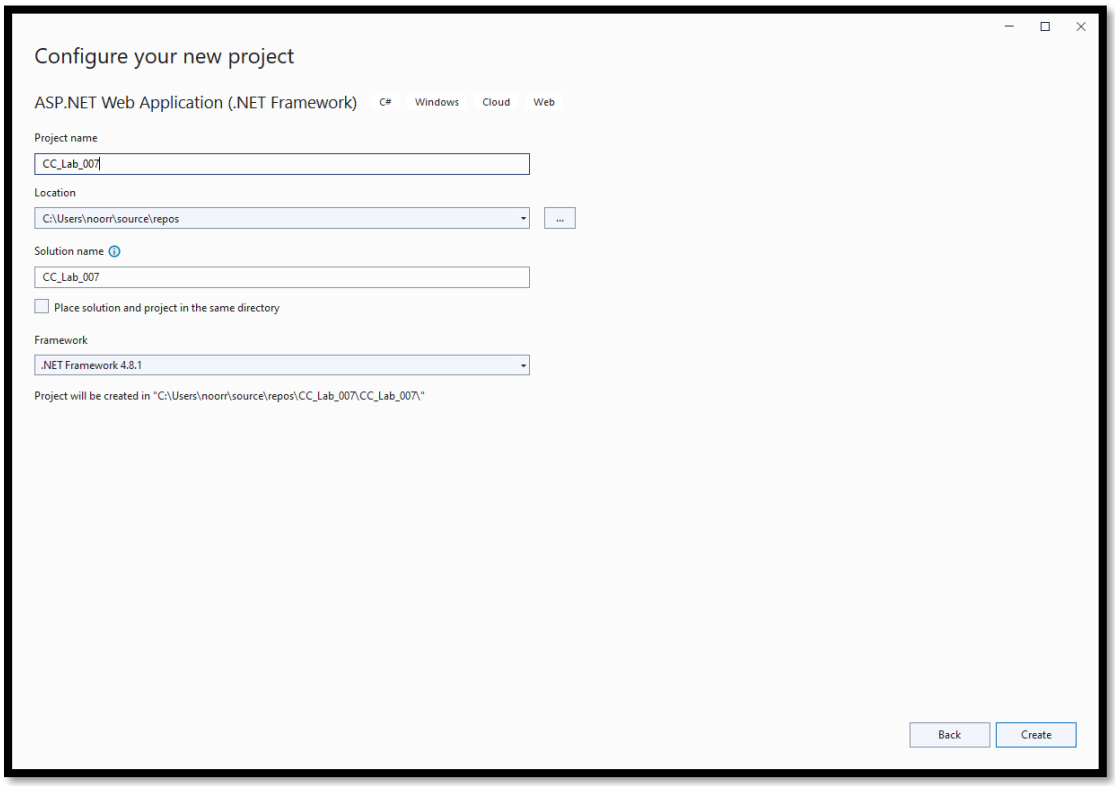
- Click on "Create a new project" in the start window.



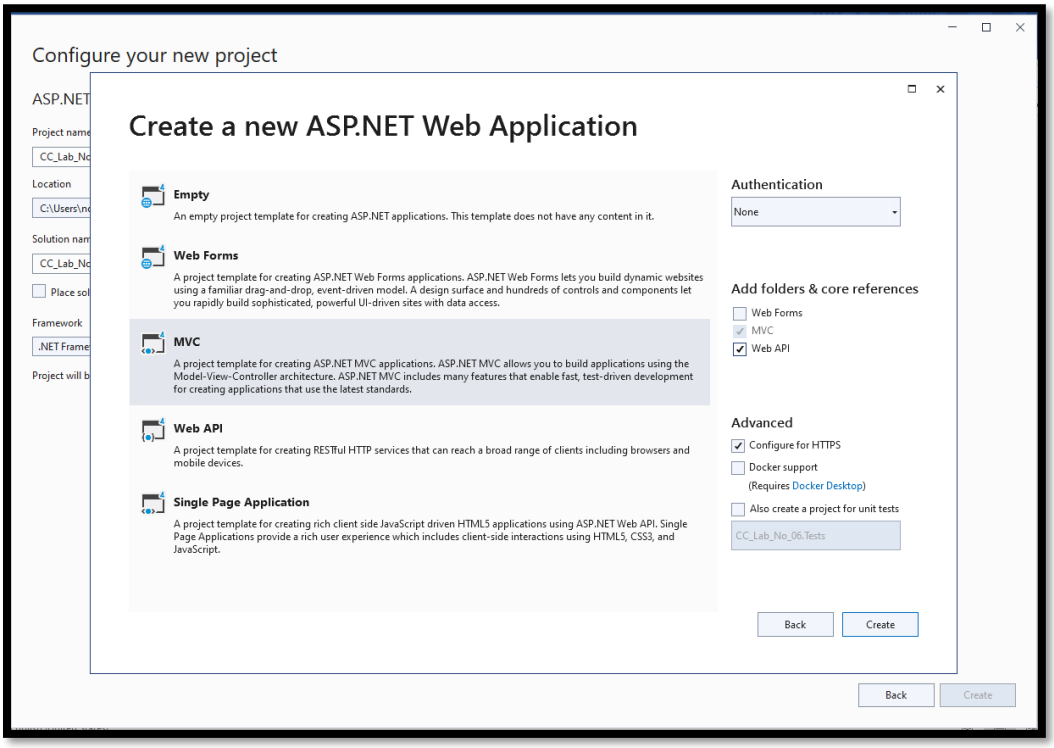
- In the "Create a new project" window, search for **ASP.NET Web Application (.net framework)** in the search bar, and then click NEXT.

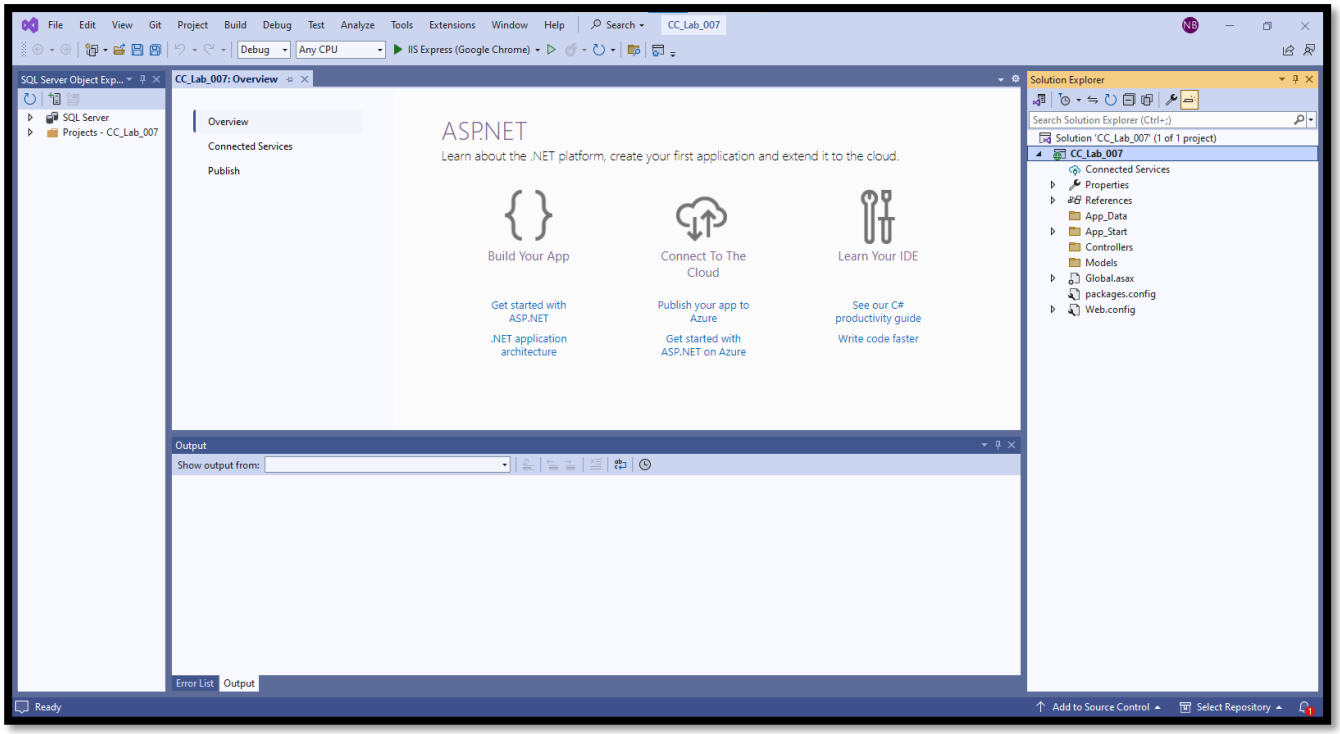


- Provide a name and location for your project and then Click NEXT.



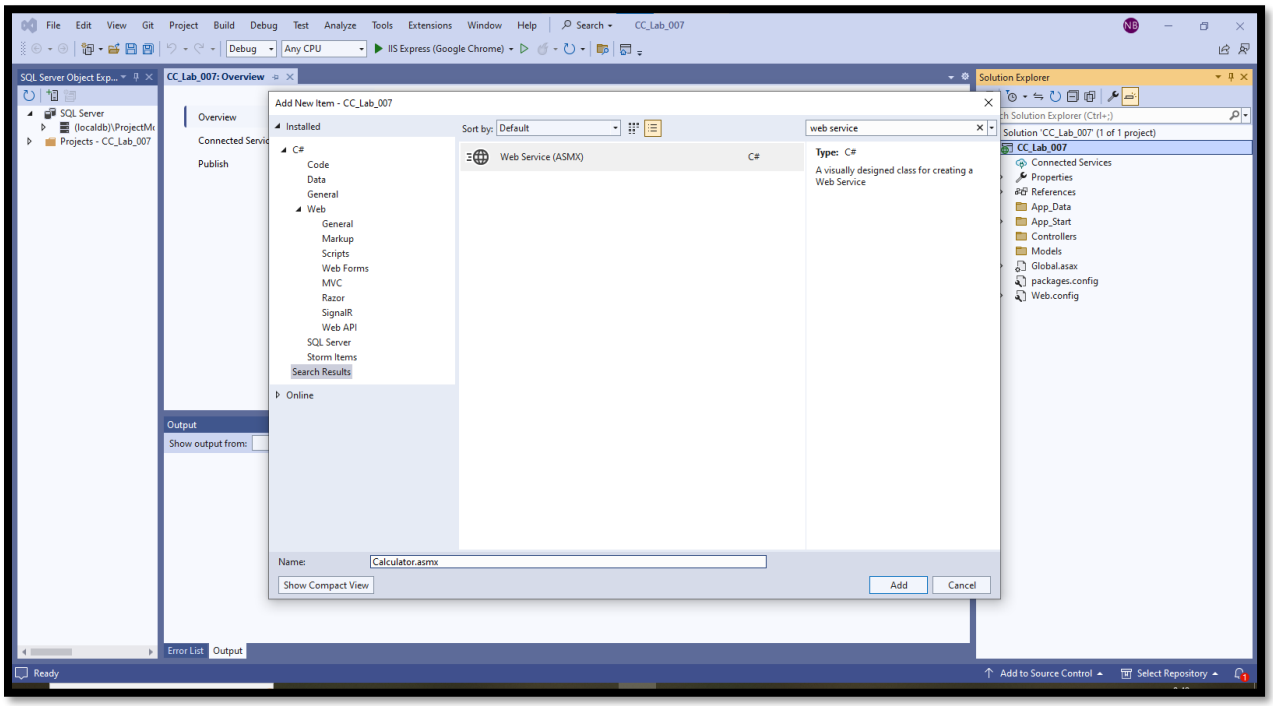
- Select the **Empty template** and check **web API** option from the list of "add folders and core references", and Click CREATE.

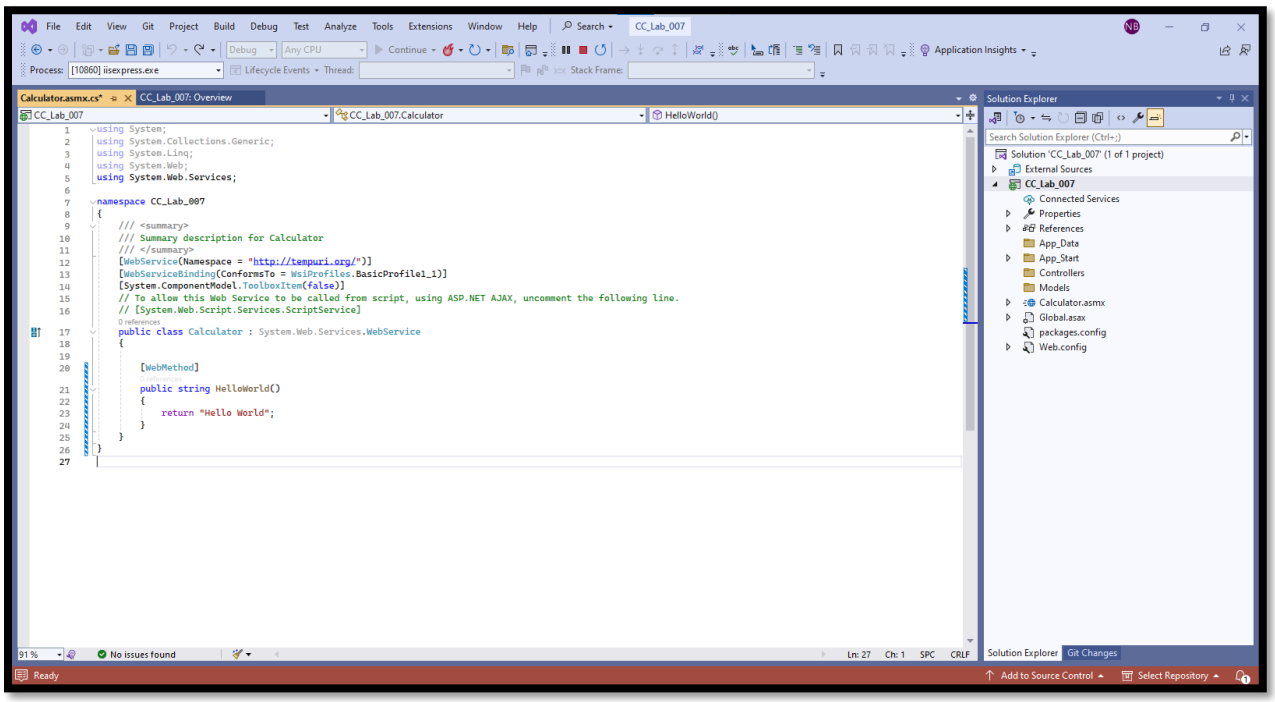




Step 2: Adding a Web Service

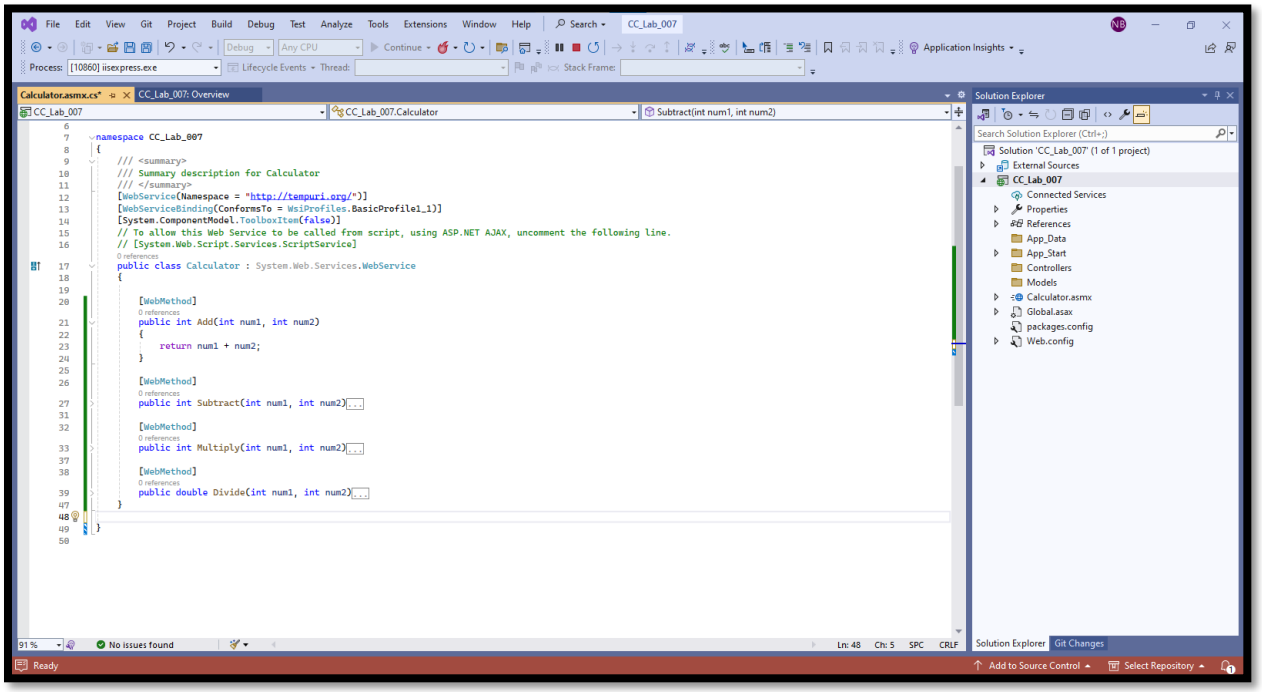
- Right-click on your project in Solution Explorer, click on add > new item. In the new opened window, search “web service”, name it calculator and then add it.





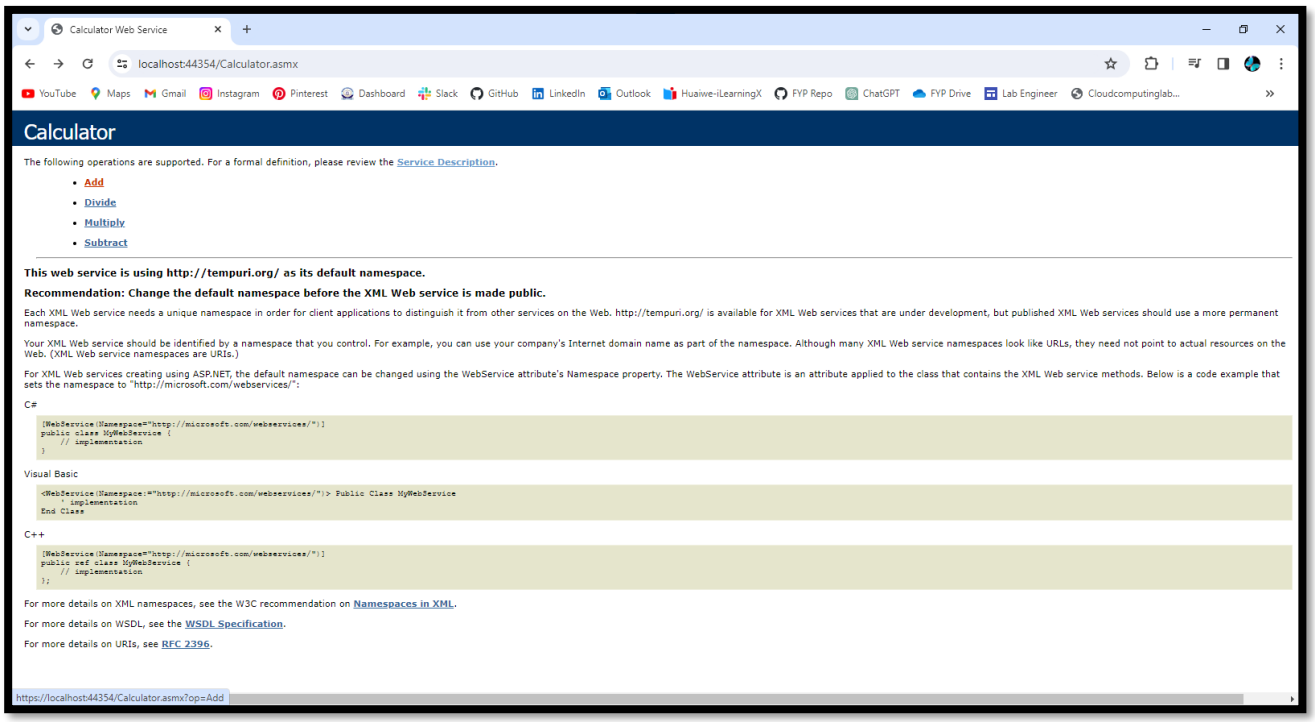
Here (in the above figure), you will note that there is predefined method "**HelloWorld**" which returns the string "**Hello World**". You can use your own method and can perform various operations

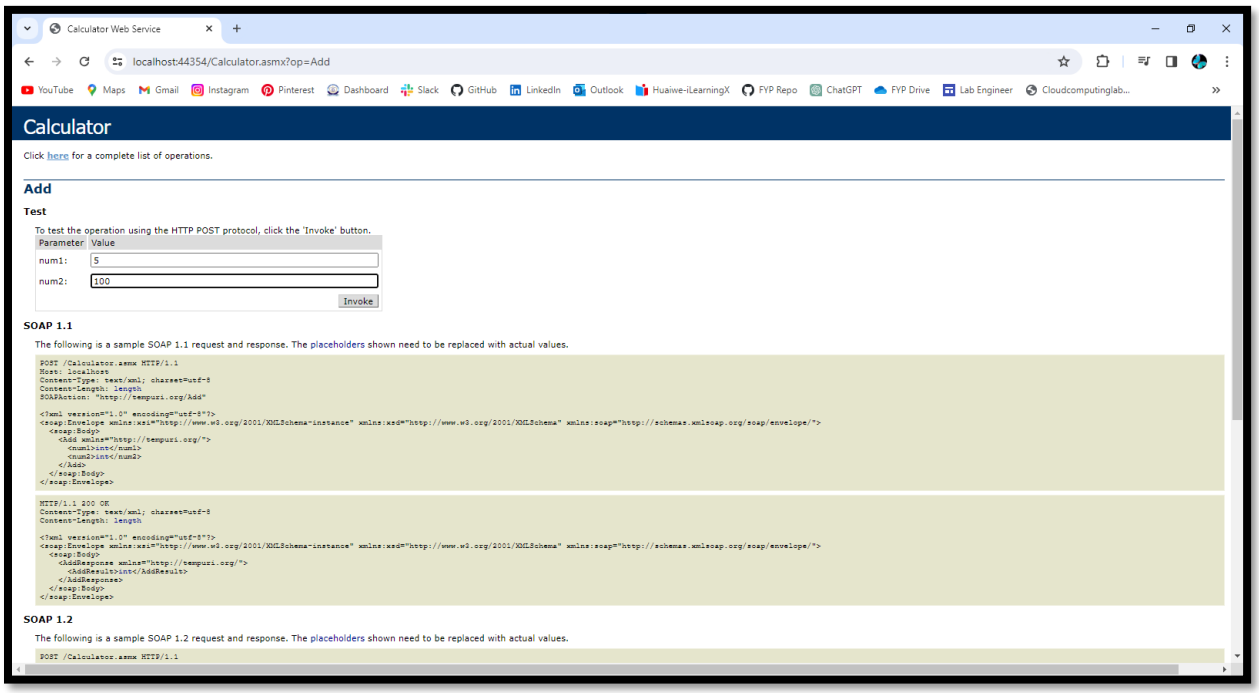
- As we've to make the simple calculator service, so remove the method present in it, and add the methods or essential for simple calculator.

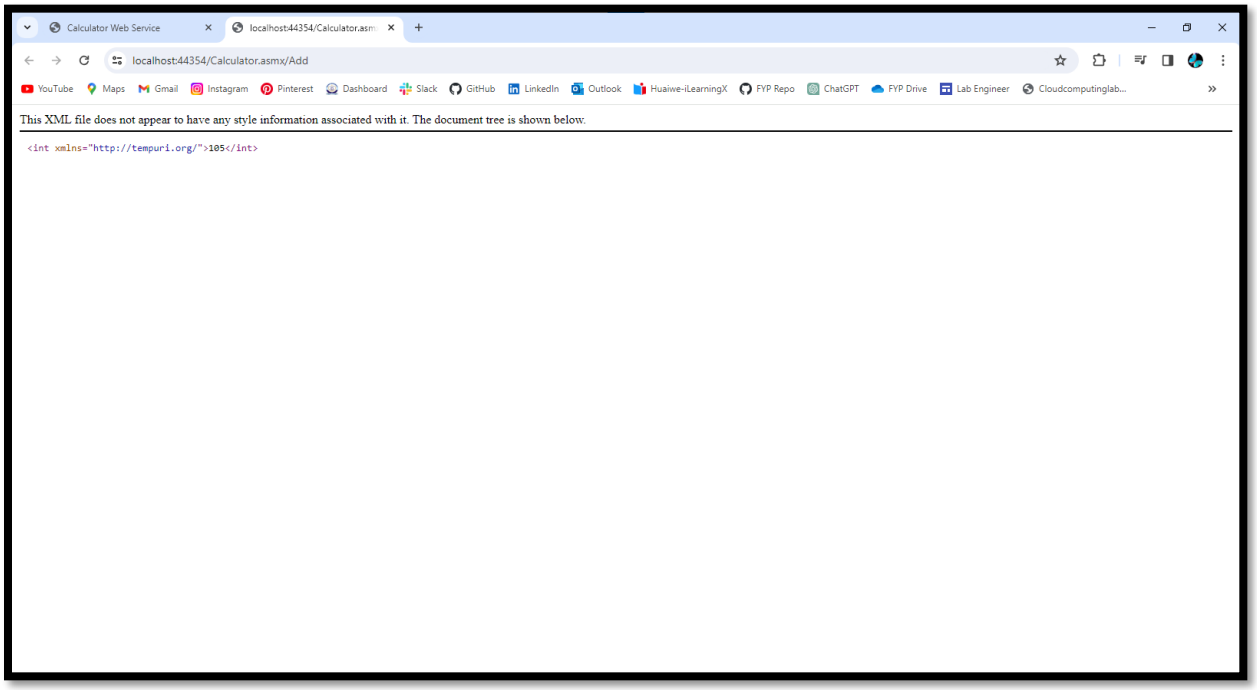


Step 3: Build and run the application

- Build Web Service and Run the Web Service for testing by pressing F5 function key.




Calculator Web Service
localhost:44354/Calculator.asmx?op=Add
Calculator
Click here for a complete list of operations.
Add
Test
To test the operation using the HTTP POST protocol, click the 'Invoke' button.
Parameter
Value
num1:
5
num2:
100
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.
SOAP 1.2
The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.


Calculator Web Service
localhost:44354/Calculator.asmx/Add
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<int xmlns="http://tempuri.org/">105</int>

2. Time Boxing

Activity Name	Activity Time	Total Time
Login Systems + Setting up Visual studio Environment	3 mints + 5 mints	8 mints
Walk through Theory & Tasks	60 mints	60 mints
Implement Tasks	80 mints	80 mints
Evaluation Time	30 mints	30 mints
	Total Duration	178 mints

3. Objectives

After completing this lab the student should be able to:

a. *Understand the concept of web services in ASP.NET.*

b. *Understand its significant role in facilitating communication between various software applications.*

c. *Know how to create web services.*

4. Lab Tasks/Practical Work

1. Design and implement a web service that provides currency conversion functionality. The web service should accept requests to convert an amount from one currency to another and return the converted amount.
2. Design and implement a web service that provides scientific calculator functionality over the internet. The web service should allow users to perform various mathematical operations, including basic arithmetic, trigonometric functions, logarithms, and more.