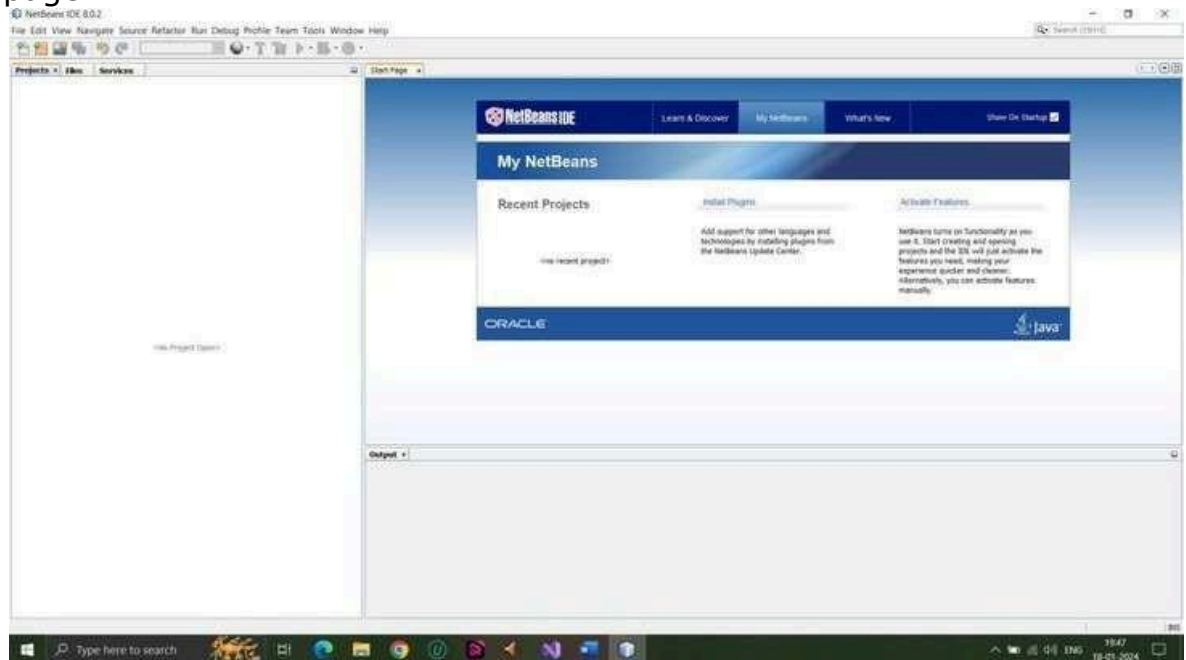


## Practical 3

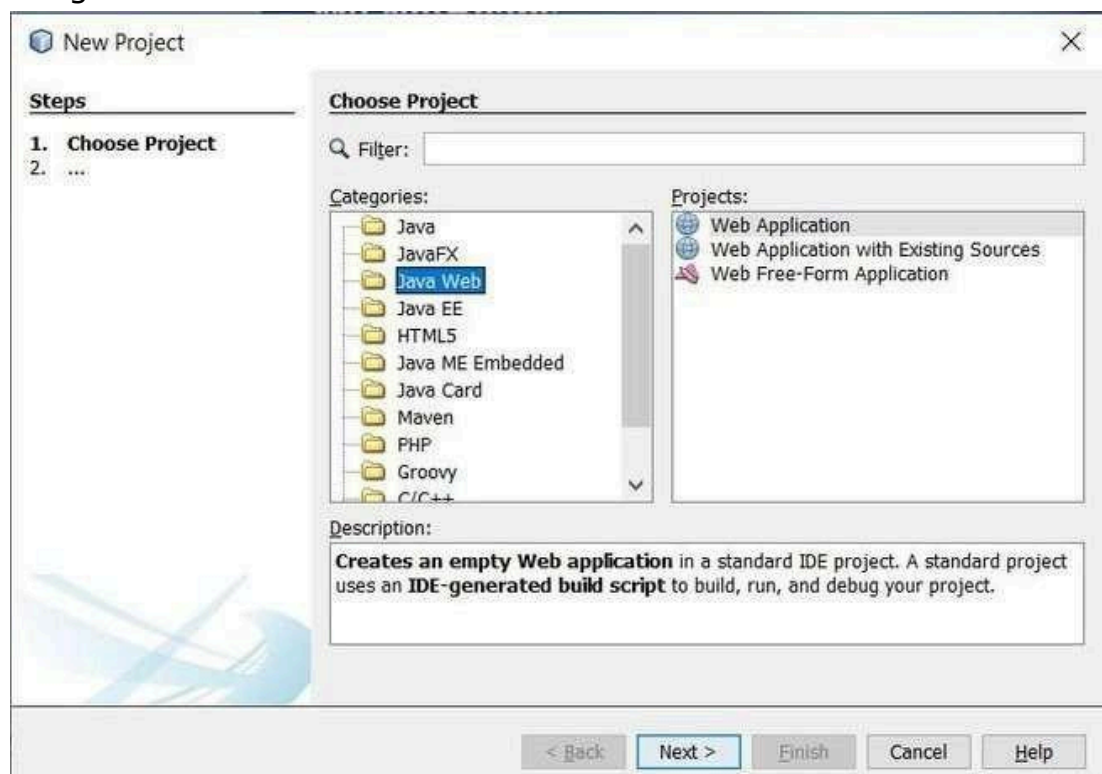
**Aim:** Create a Simple SOAP service for consuming java web services in Python.

### Steps:

1. Open the NetBeans , and you will get the following screen. Close the start page.



2. Now click on the file tab and click on new project you will get the following screen :



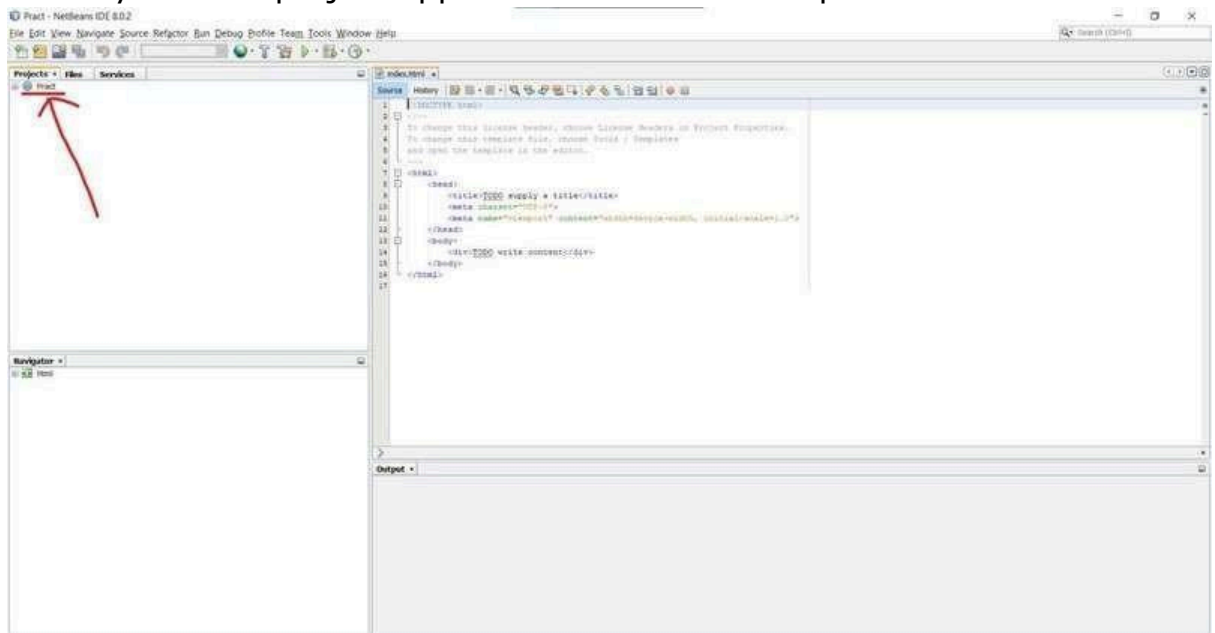
3. In Categories select Java Web and in Projects , Select Web Application .After selecting click on next You will get the following window:

The screenshot shows the 'New Web Application' dialog box with the 'Name and Location' tab selected. The 'Steps' list on the left includes: 1. Choose Project, 2. **Name and Location**, 3. Server and Settings, and 4. Frameworks. The main area contains fields for 'Project Name' (Pract1), 'Project Location' (C:\Users\LENOVO\Documents\NetBeansProjects), and 'Project Folder' (C:\Users\LENOVO\Documents\NetBeansProjects\Pract1). There are 'Browse...' buttons for both the location and folder fields. A checkbox for 'Use Dedicated Folder for Storing Libraries' is unchecked. Below it is a 'Libraries Folder' field with a 'Browse...' button. A note states: 'Different users and projects can share the same compilation libraries (see Help for details)'. A red error message at the bottom reads: 'Project Folder already exists and is not empty.' The bottom of the dialog has buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

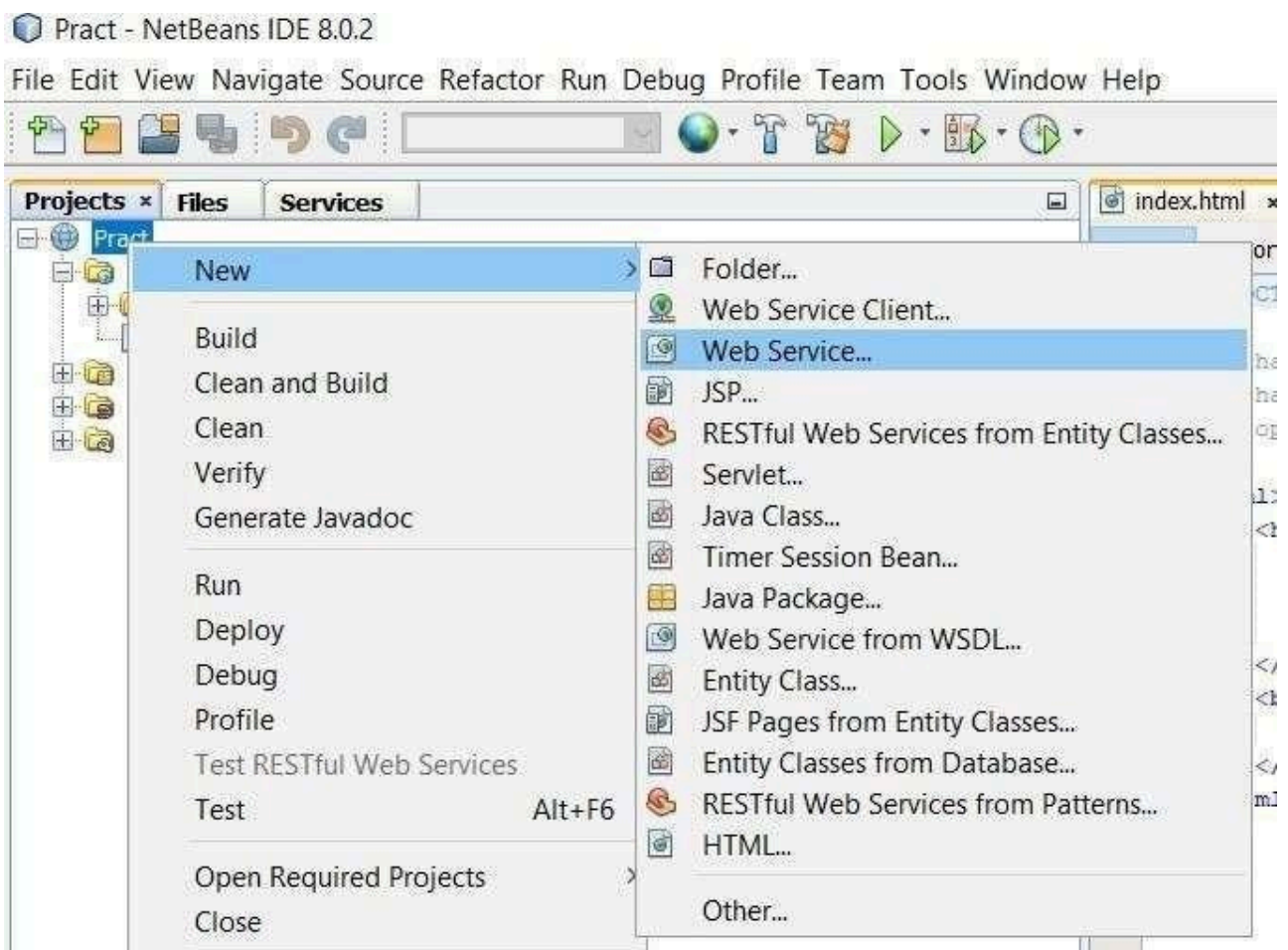
4. Now Give name to the Project Name: , and click on next. You will get the following window then again click on finish

The screenshot shows the 'New Web Application' dialog box with the 'Server and Settings' tab selected. The 'Steps' list on the left includes: 1. Choose Project, 2. Name and Location, 3. **Server and Settings**, and 4. Frameworks. The main area contains a dropdown for 'Add to Enterprise Application' set to '<None>'. The 'Server' dropdown is set to 'GlassFish Server 4.1' with an 'Add...' button. The 'Java EE Version' dropdown is set to 'Java EE 7 Web'. The 'Context Path' field contains '/Pract'. The bottom of the dialog has buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

5. You will get the following screen now carefully see in projects section your recently created project appears double click to expand it:



6. Now Right click on the project and select new and then select Web Service ,As shown Below



7. After clicking Web Service the following window should appear , now give name to web service and give package as "server". As shown in the image

**New Web Service**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Web Service Name: Currency

Project: Pract

Location: Source Packages

Package: server

☒ Create Web Service from Scratch

☐ Create Web Service from Existing Session Bean

Enterprise Bean:

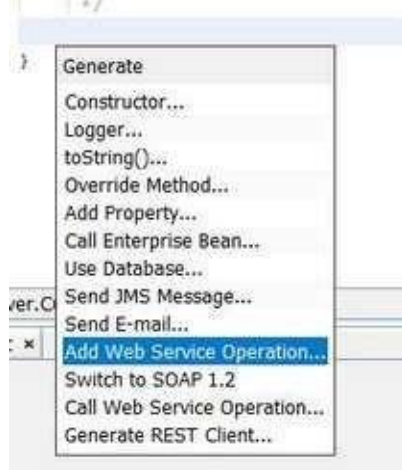
☐ Implement Web Service as Stateless Session Bean

< Back Next > **Finish** Cancel Help

After clicking finish you should get the following window erase the mentioned code

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package server;
7
8  import javax.jws.WebService;
9  import javax.jws.WebMethod;
10 import javax.jws.WebParam;
11
12 /**
13  *
14  * @author LENOVO
15  */
16 @WebService(serviceName = "Currency")
17 public class Currency {
18
19     /**
20      * This is a sample web service operation
21      */
22     @WebMethod(operationName = "hello")
23     public String hello(@WebParam(name = "name") String txt) {
24         return "Hello " + txt + " !";
25     }
26 }
27
```

8. Now right click anywhere and click on insert code and select Add Web Service Operation



9. The following window would appear :  
Just give name to the method or operation and click on add button to add parameters to the method as here we are converting dollar to rupees we should need only one parameter .

A screenshot of the 'Add Operation' dialog box. The dialog has a title bar with a close button. It contains a 'Name' field with the text 'operation' and a 'Return Type' field with the text 'java.lang.String'. There is a 'Browse...' button next to the 'Return Type' field. Below these fields are two tabs: 'Parameters' and 'Exceptions'. The 'Parameters' tab is selected. It contains a table with three columns: 'Name', 'Type', and 'Final'. To the right of the table are four buttons: 'Add', 'Remove', 'Up', and 'Down'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Name	Type	Final
------	------	-------

10. Give the name to the variable select data type as double and click on ok as shown below:

Dialog: Add Operation

Name: InrtoDollr

Return Type: java.lang.String

Parameters

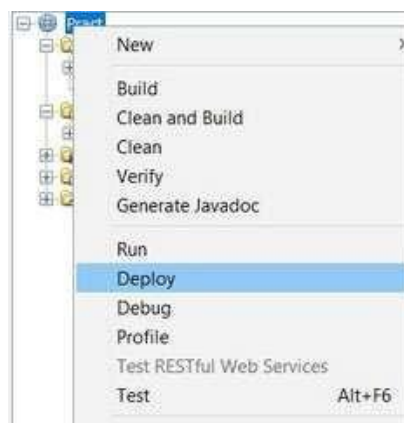
Name	Type	Final
a	double	<input type="checkbox"/>

Buttons: Add, Remove, Up, Down, OK, Cancel

11. After clicking ok code will auto generated , make changes In that code as mentioned below:

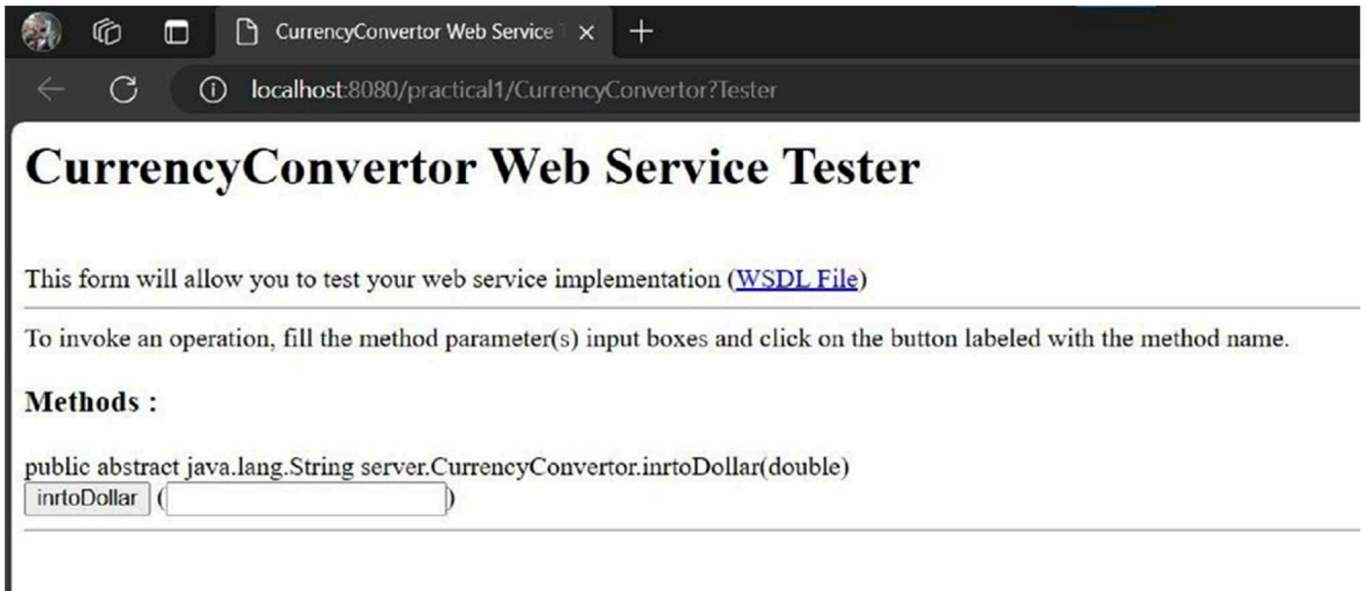
```
@WebMethod(operationName = "InrtoDollar")
public String InrtoDollar(@WebParam(name = "a") double a) {
    //TODO write your implementation code here:
    return "The Indian rupees "+a+" in Dollars is "+(a/83.17);
}
```

12. Now our web service is ready now right click on project and click on deploy

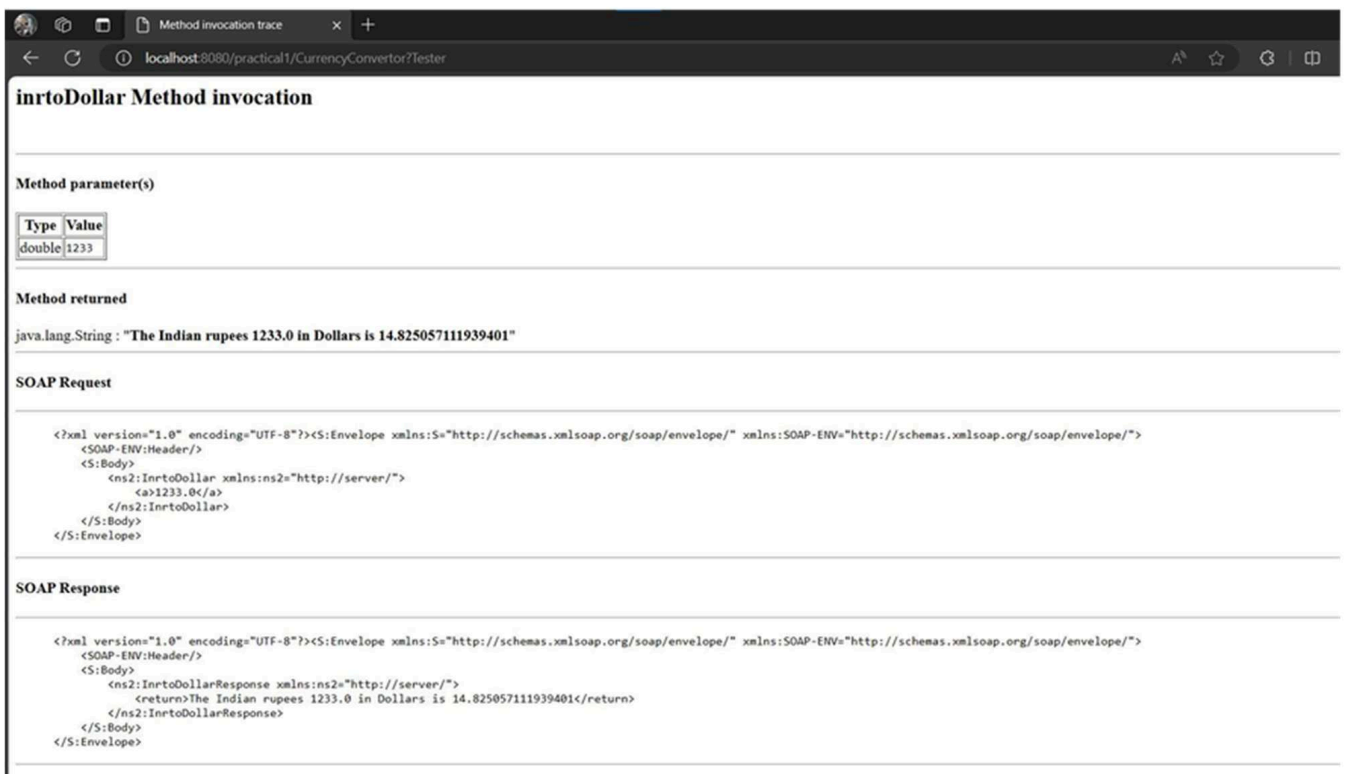




13. Now right click on web service and click on "test web service" you will get the following output:



The screenshot shows a web browser window with the title "CurrencyConverter Web Service". The address bar shows "localhost:8080/practical1/CurrencyConverter?Tester". The main heading is "CurrencyConverter Web Service Tester". Below the heading, there is a paragraph: "This form will allow you to test your web service implementation ([WSDL File](#))". Another paragraph follows: "To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name." Below this, the section "Methods :" is displayed. Under "Methods :", the text "public abstract java.lang.String server.CurrencyConverter.inrtoDollar(double)" is shown. Below this text, there is a button labeled "inrtoDollar" and an input field containing the value "1233".



The screenshot shows a web browser window with the title "Method invocation trace". The address bar shows "localhost:8080/practical1/CurrencyConverter?Tester". The main heading is "inrtoDollar Method invocation". Below the heading, there is a section "Method parameter(s)". Under "Method parameter(s)", there is a table with two columns: "Type" and "Value". The table contains one row with "double" in the "Type" column and "1233" in the "Value" column. Below the table, there is a section "Method returned". Under "Method returned", the text "java.lang.String : "The Indian rupees 1233.0 in Dollars is 14.825057111939401"" is displayed. Below this, there is a section "SOAP Request". Under "SOAP Request", the XML code is shown: 

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:InrtoDollar xmlns:ns2="http://server/">
      <a>1233.0</a>
    </ns2:InrtoDollar>
  </S:Body>
</S:Envelope>
```

 Below the SOAP Request, there is a section "SOAP Response". Under "SOAP Response", the XML code is shown: 

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:InrtoDollarResponse xmlns:ns2="http://server/">
      <return>The Indian rupees 1233.0 in Dollars is 14.825057111939401</return>
    </ns2:InrtoDollarResponse>
  </S:Body>
</S:Envelope>
```

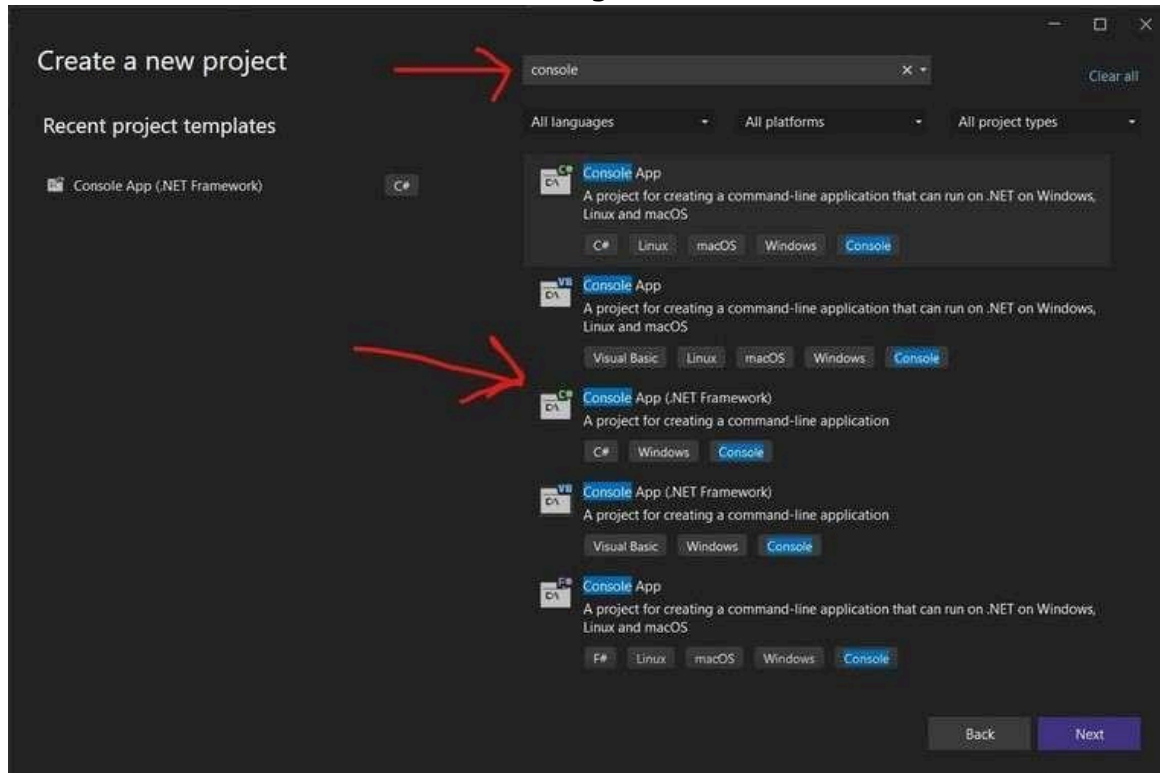
So this is how we created our web service and deployed it.

---

## Creating a .Net Client

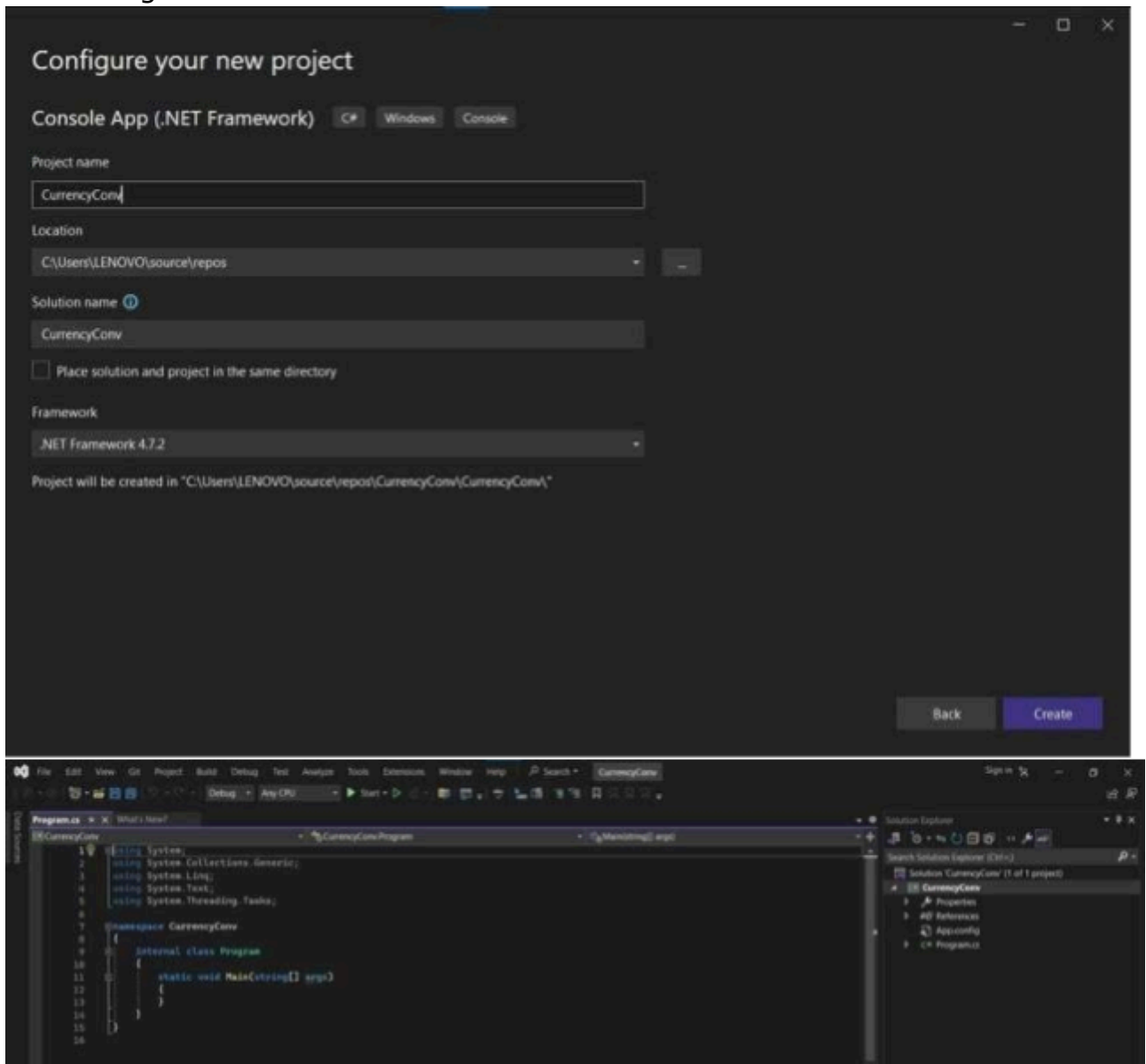
14. Open visual studio and select create a new project the following window will appear:

Here in search bar type "console c#" and select the console app for the .Net framework as shown in the image.





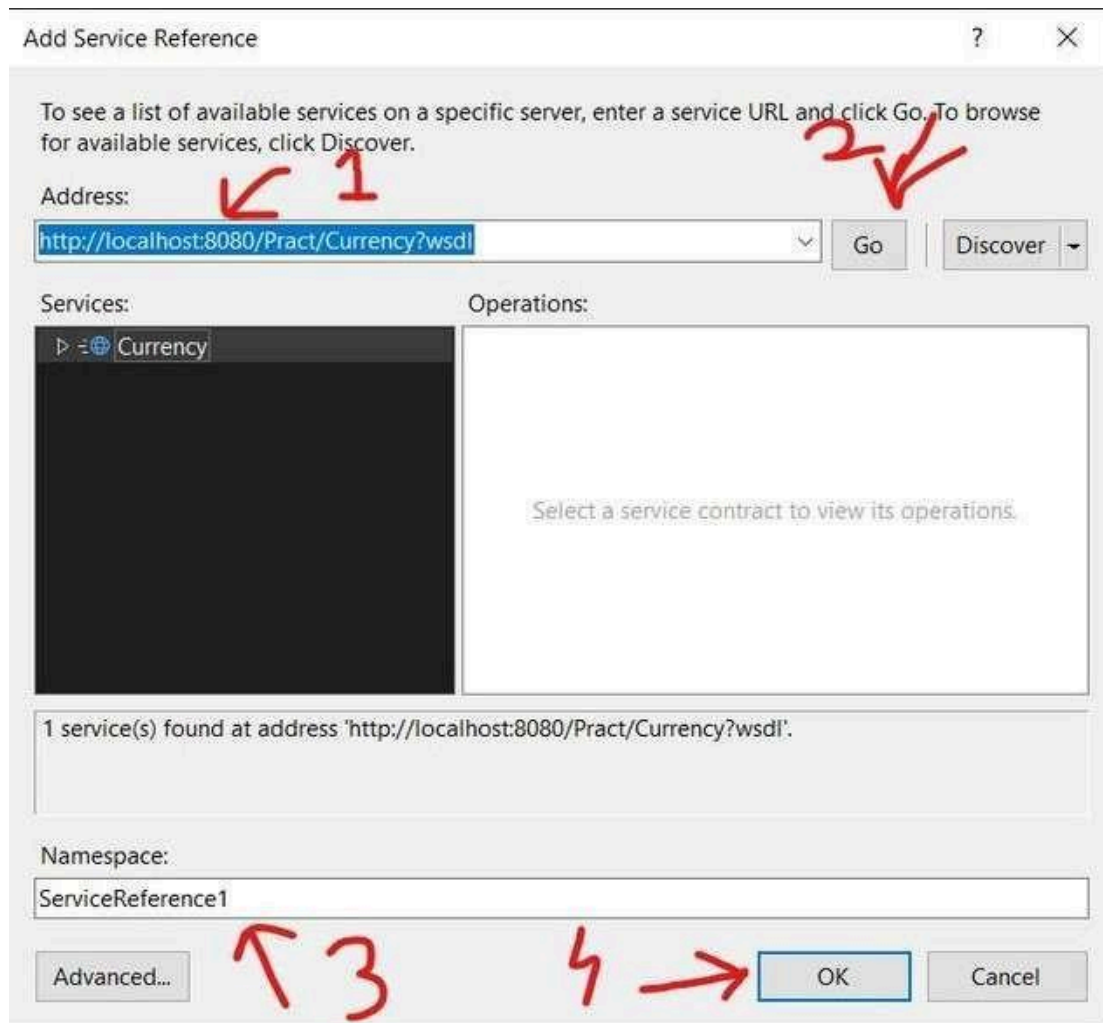
15. Give name to the project and press create button as shown in the image:



16. This window should appear after clicking the create button now. On the right side there is solution Explorer.

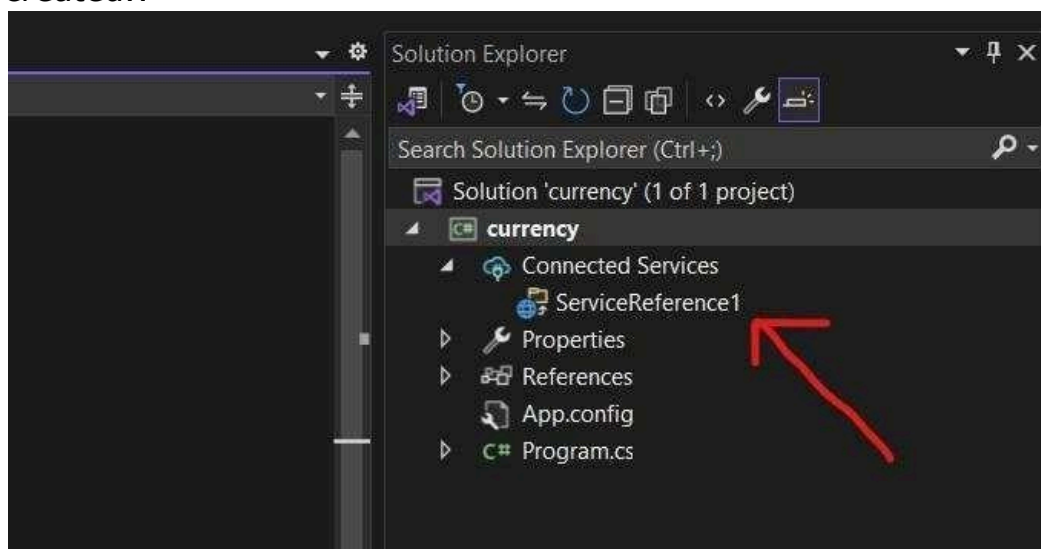
In that right click on your project name and click on add > Service Reference . You will get the following window :

**Note: before executing the code make sure that the web service is deployed.**



17. Here in the address bar paste the wsdl url. And press the go button . After set the namespace In my case I am letting is as it is then press ok button.

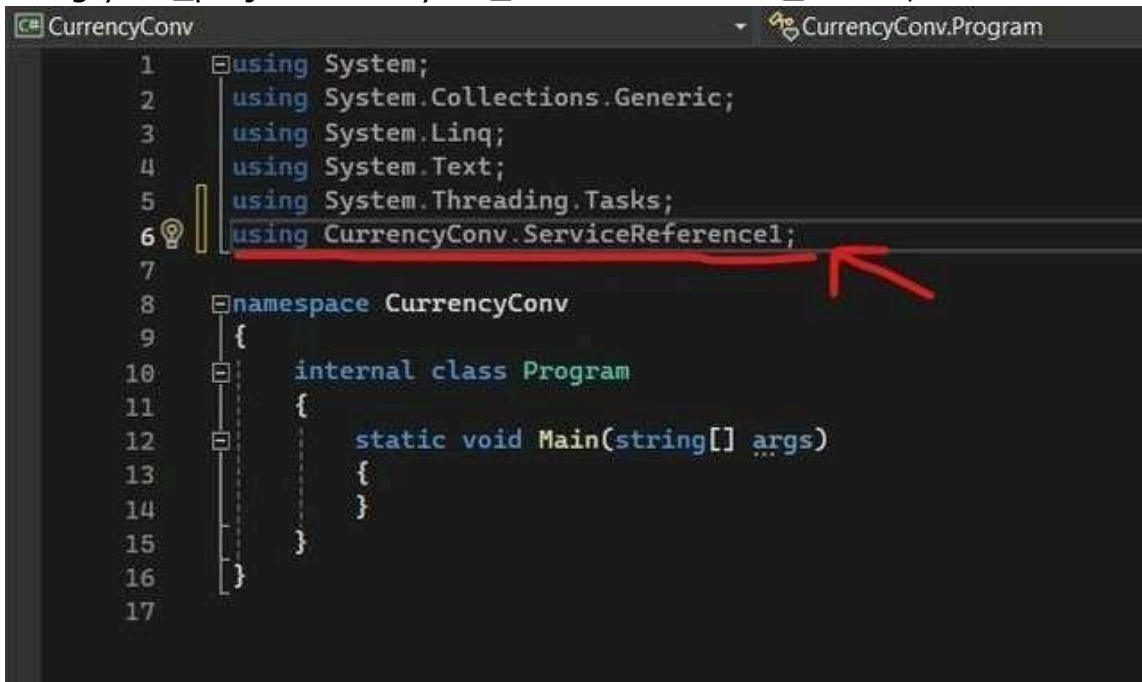
In following image u can check in solution Explorer service reference was created.:



18. Now in code write the following line as shown in image:

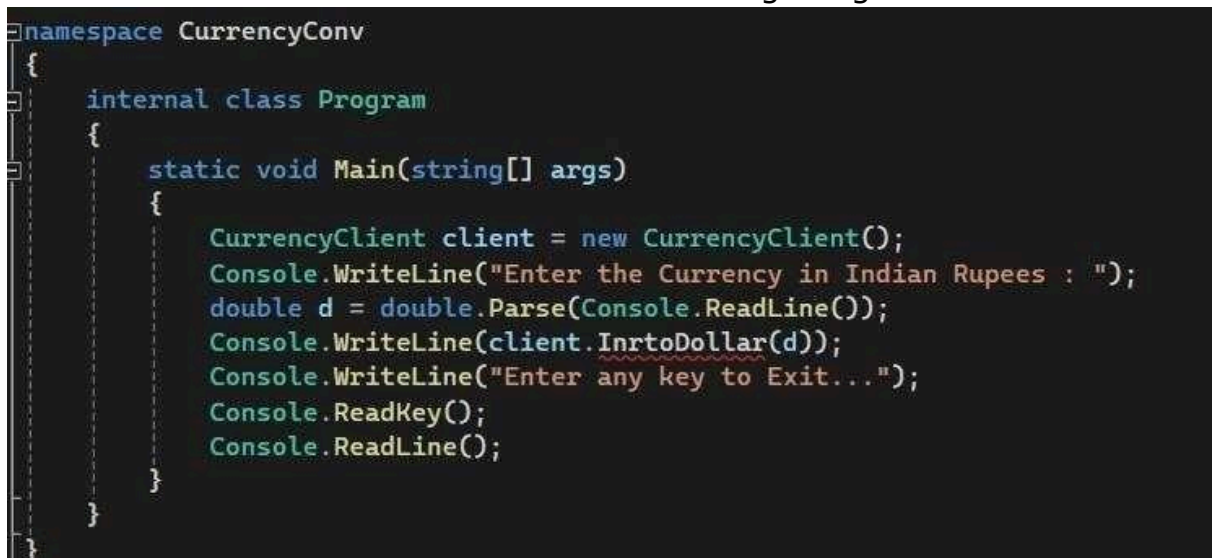
Here type:

Using your\_projectname .your\_serviceReference\_name ;



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using CurrencyConv.ServiceReference1;
7
8 namespace CurrencyConv
9 {
10     internal class Program
11     {
12         static void Main(string[] args)
13         {
14         }
15     }
16 }
17
```

19. Now write the code as shown in the following image:



```
namespace CurrencyConv
{
    internal class Program
    {
        static void Main(string[] args)
        {
            CurrencyClient client = new CurrencyClient();
            Console.WriteLine("Enter the Currency in Indian Rupees : ");
            double d = double.Parse(Console.ReadLine());
            Console.WriteLine(client.InrtoDollar(d));
            Console.WriteLine("Enter any key to Exit...");
            Console.ReadKey();
            Console.ReadLine();
        }
    }
}
```

**Note:**

Here **CurrencyClient** is my webservice class name and **operation()** is my method name as show you can also check your own



20. When you run the above code you get the following output:



```
C:\Windows\system32\cmd.exe
Enter the Currency in indian rupees:
1000
The Indian rupees 100.0 in Dollars is 1.2023566189731874
Enter any key to Exit...
```

The screenshot shows a Windows command prompt window with the title bar 'C:\Windows\system32\cmd.exe'. The program prompts the user to 'Enter the Currency in indian rupees:'. The user enters '1000'. The program then outputs 'The Indian rupees 100.0 in Dollars is 1.2023566189731874'. Finally, it prompts the user to 'Enter any key to Exit...'. The background of the command prompt shows some faint, semi-transparent text from the code being executed, including 'using System;', 'using System.Collections.Generic;', 'using System.Linq;', 'using System.Text;', and 'using System.Threading.Tasks;'.