

**Name: Patel Shivam S.**  
**Enroll no: 21162101019**  
**Sem: 5      Batch: 51      Branch: CBA**  
**Sub: Microservices**

## **Practical 1**

**AIM:** Implement an application of web service using JAX-RPC in JAVA.

**Exercise:** A shop owner wants to make an arithmetic calculator using JAX-RPC and wants to acquire that service as a java client. Help them to make this service.

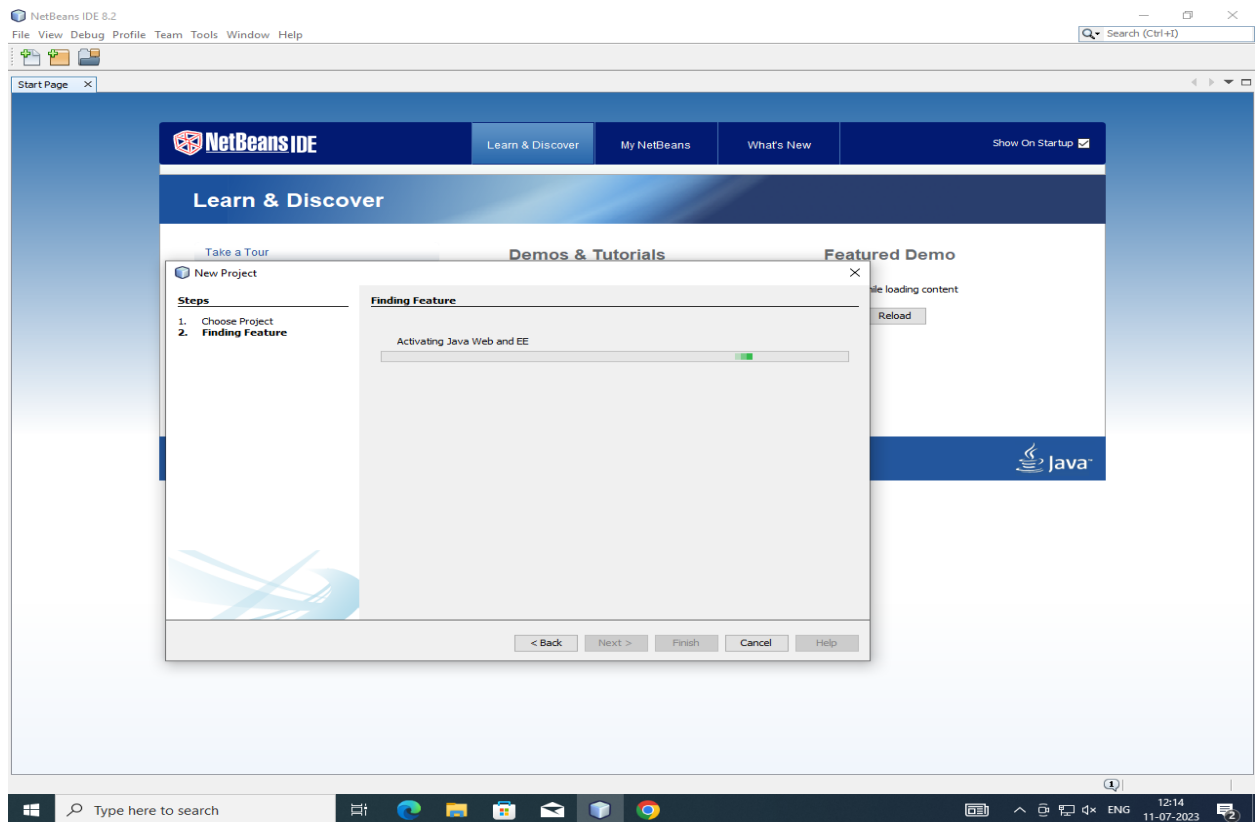
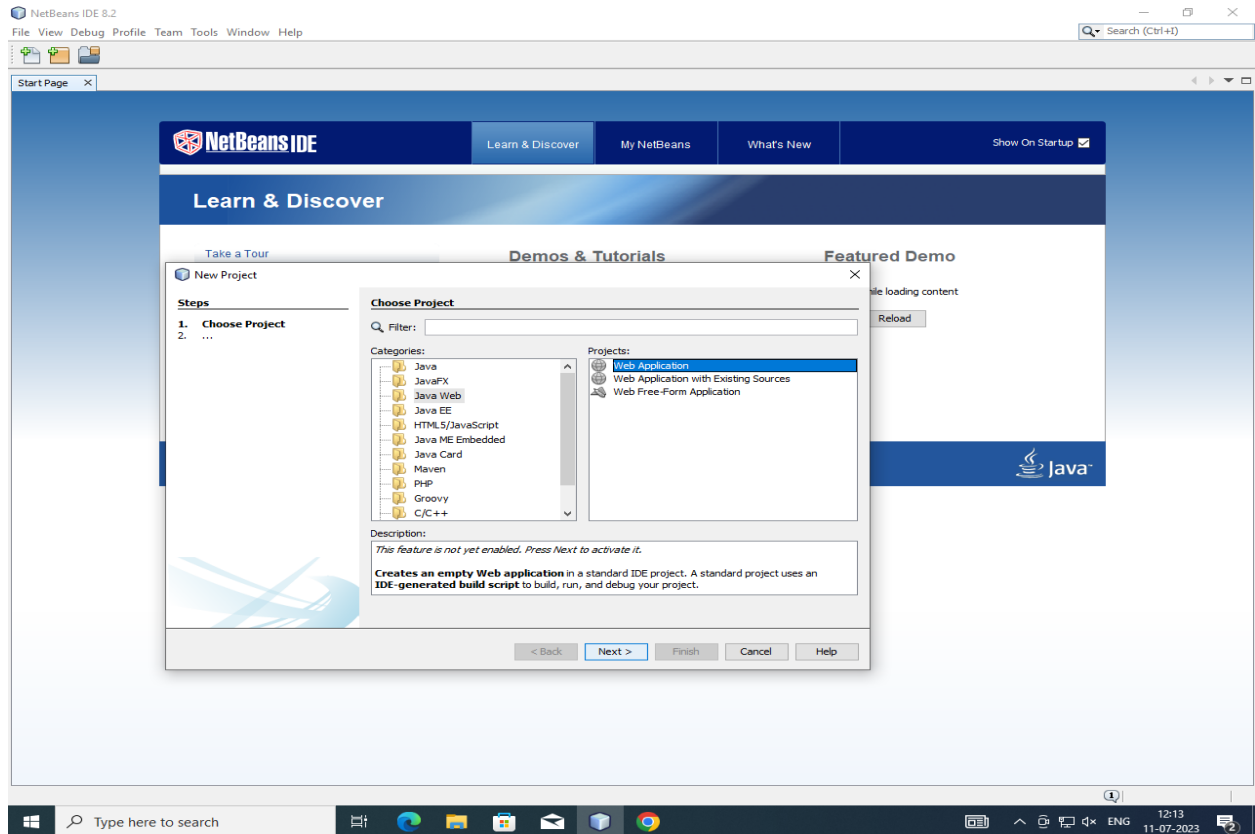
**Solution:**

For creating service, First of all we have to open netbeans in that

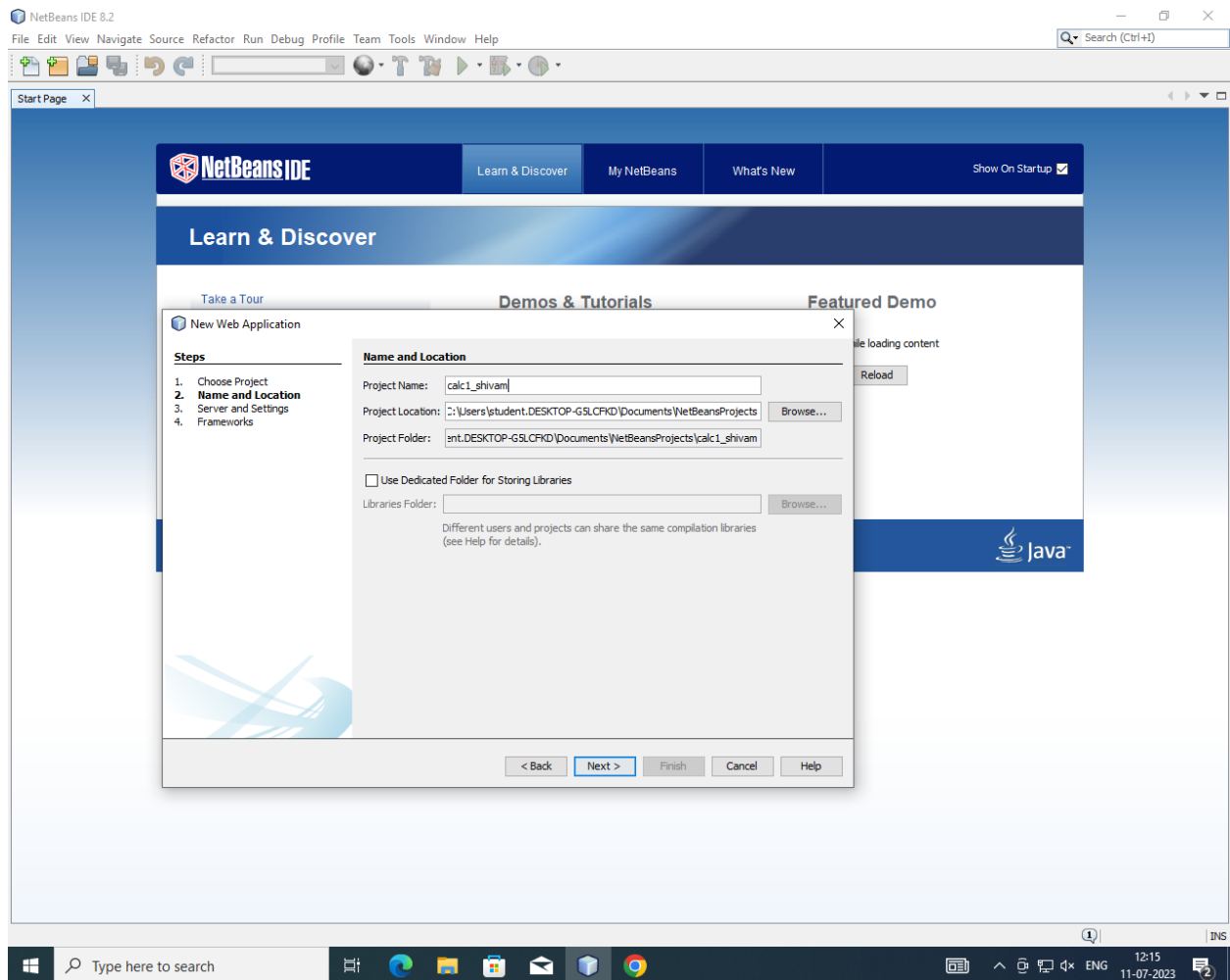
Click on File → New Project. We can see like this.

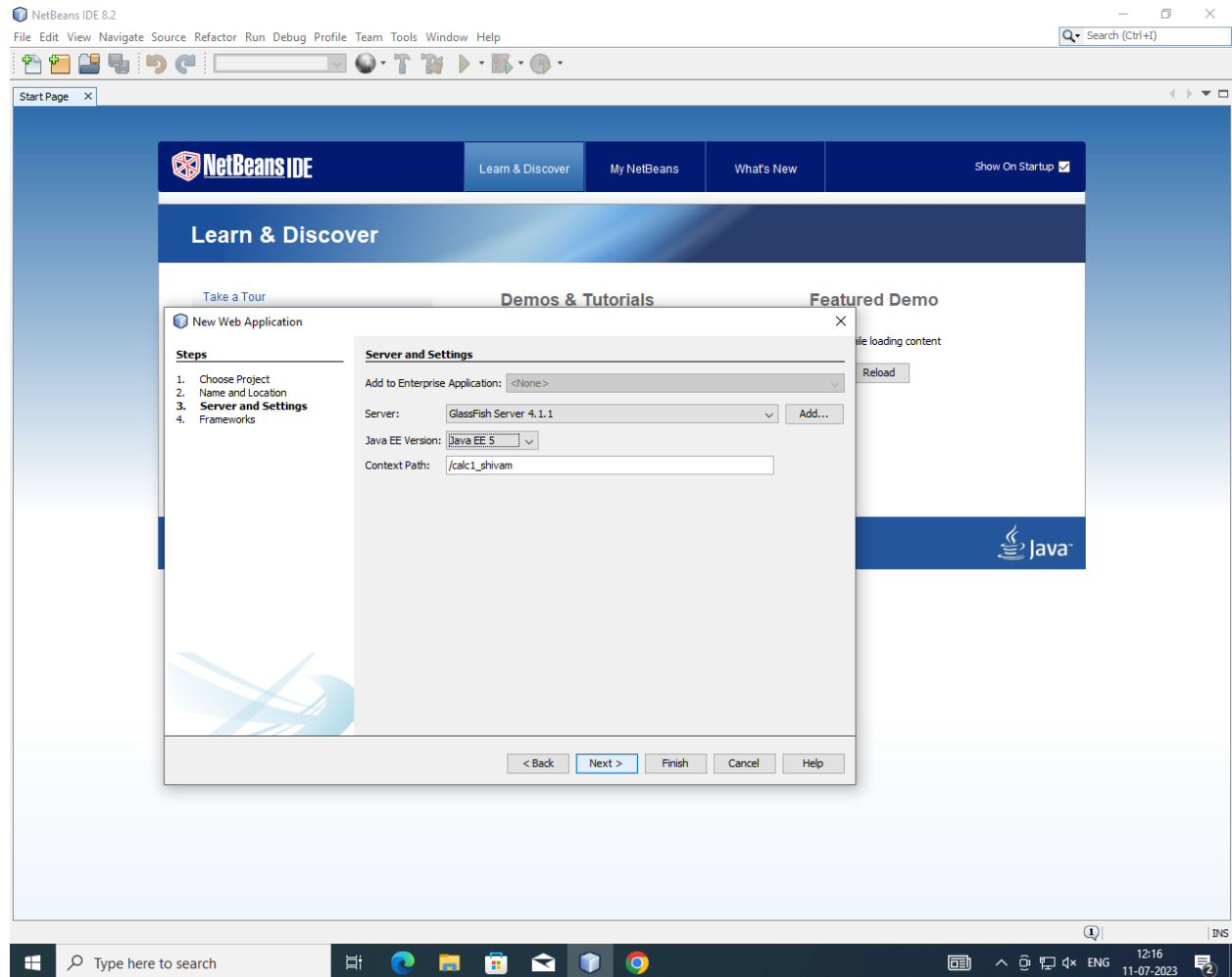
Click on Java Web → Web Application

Click on Next.

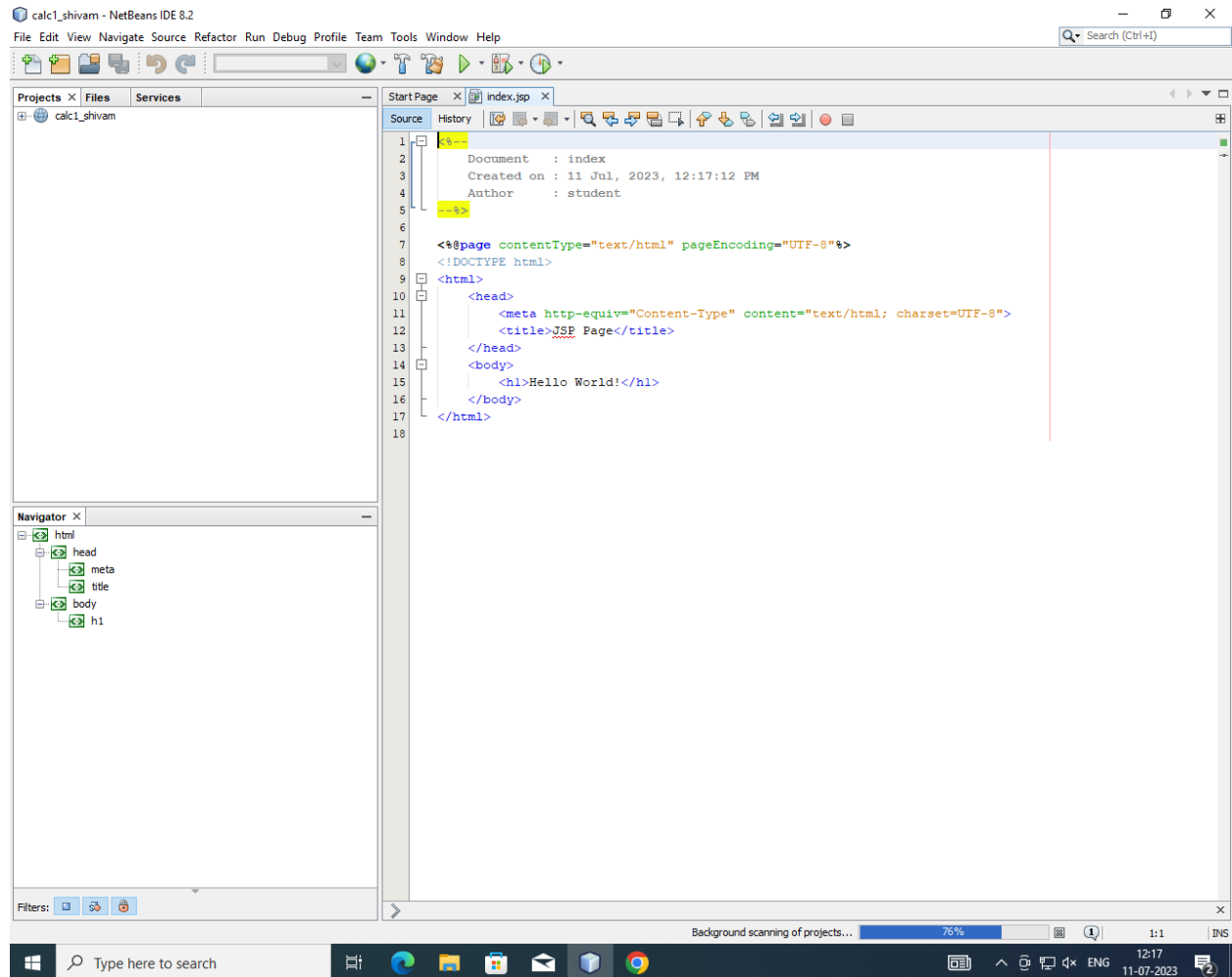


After that type project name and click on next and in server add Glassfish server 4.1.1 also Java EE5.





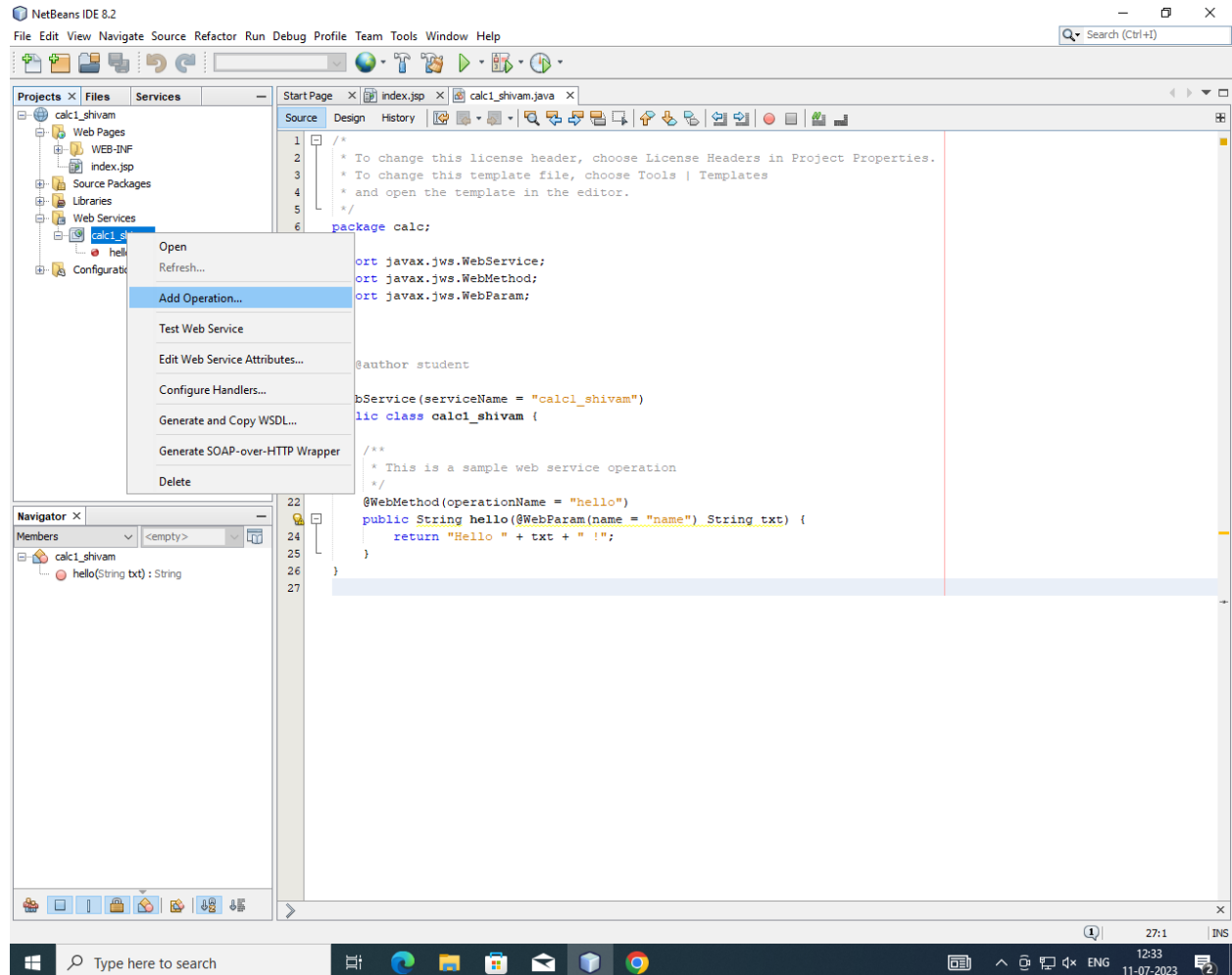
Now Click on Finish. We can see like this..



Now, Right Click on Project name → New → Web Service and enter Web Service name, Package and choose Create a web service from scratch. Then Click on Finish.



Now we can see Web Services folder in that Right click on Project name and click Add Operation. And enter name, return type = Float. And add parameters then click Ok. after that we can see in .java file that operation service entered automatically.



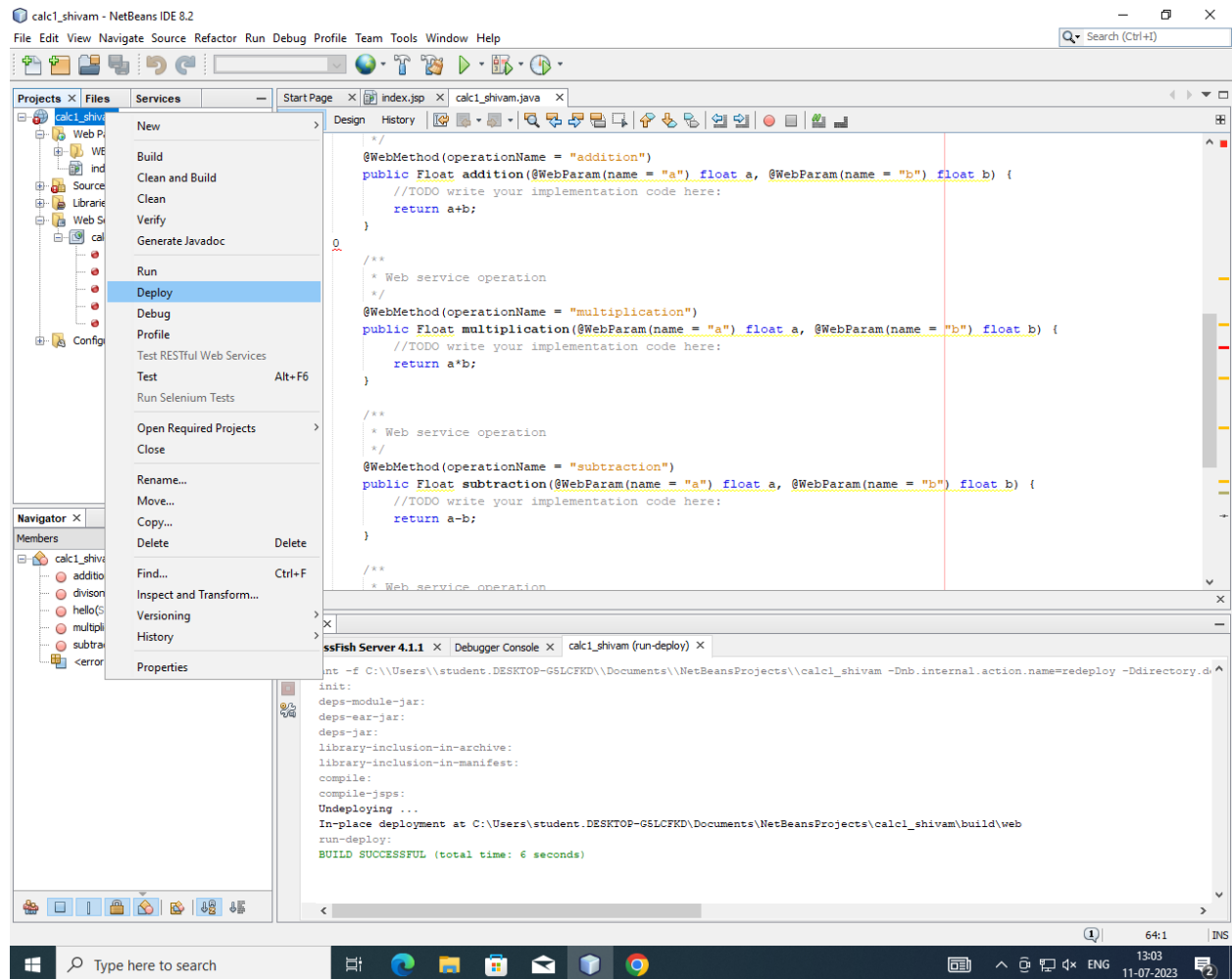


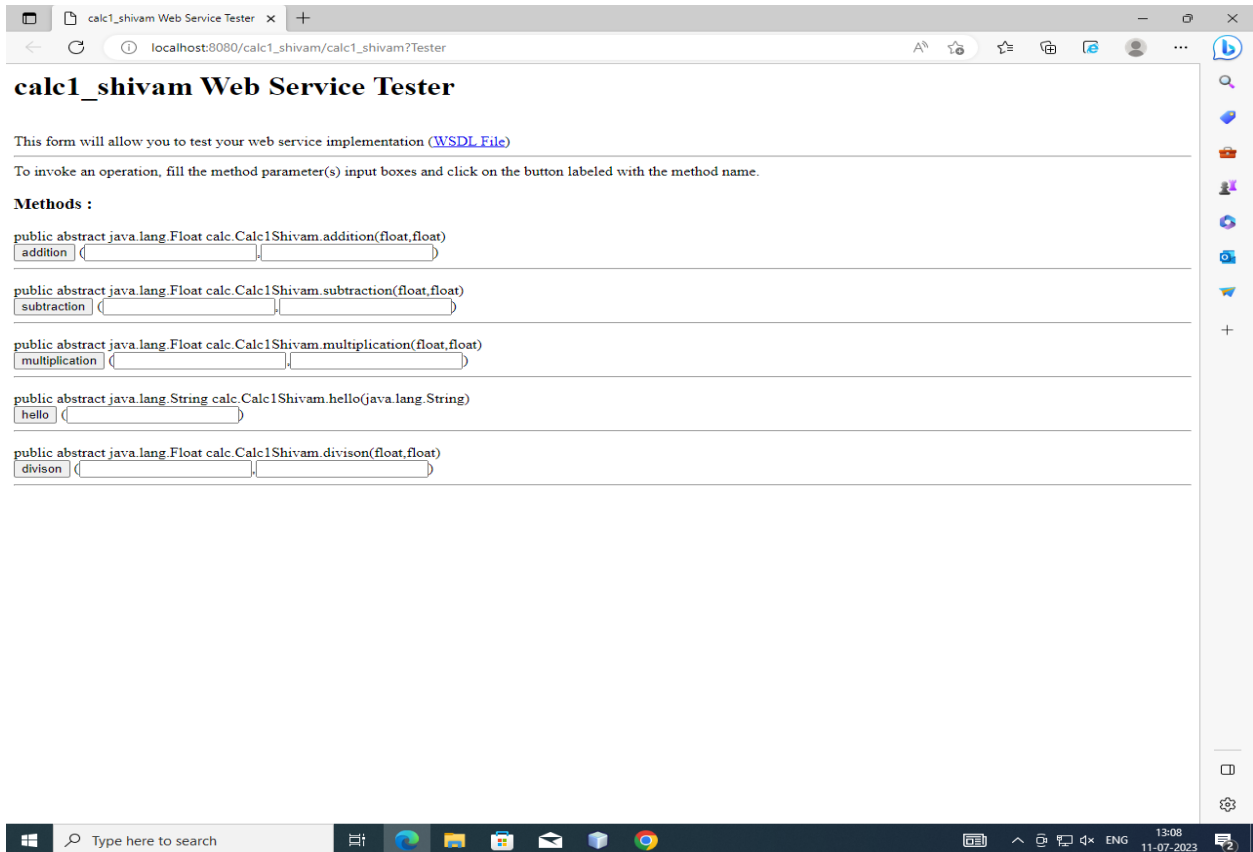
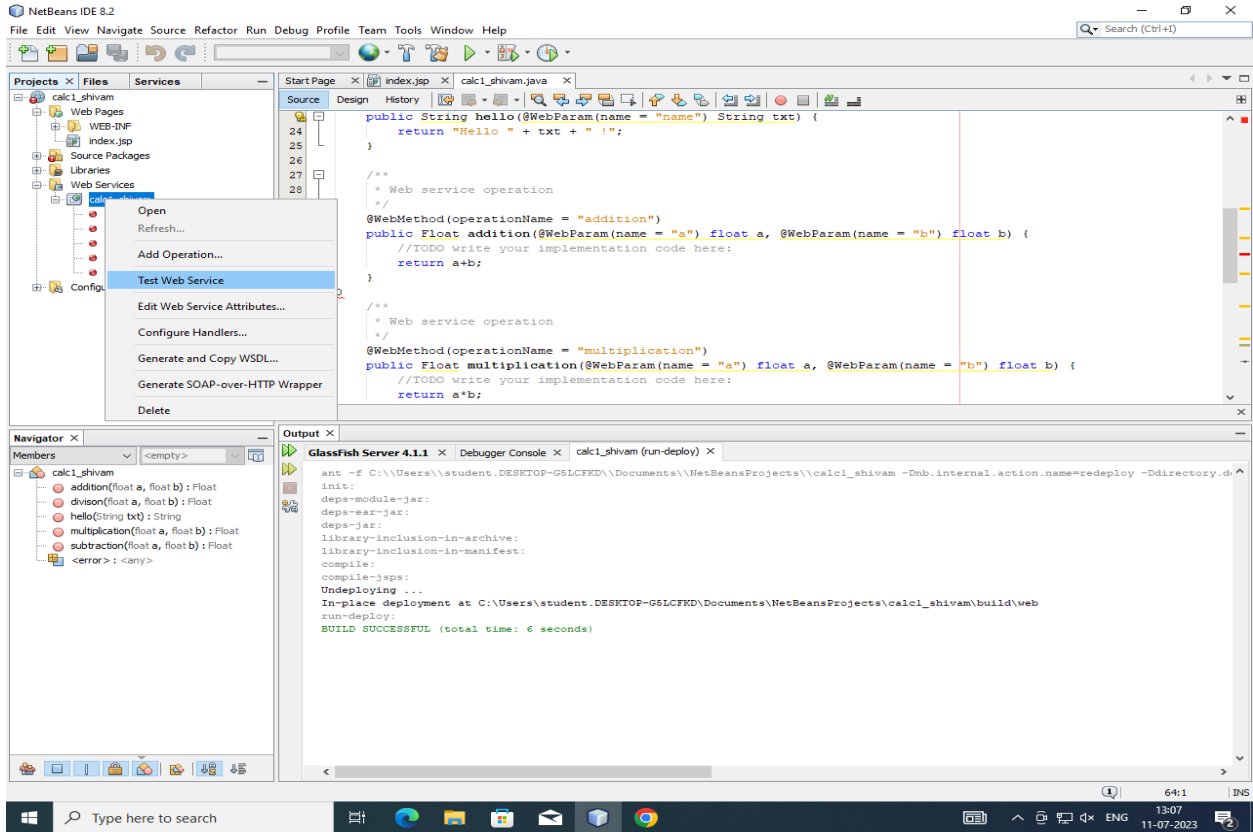


Now Right Click on main project name and click Deploy.

Then make sure in output we can see BUILD SUCCESSFULLY. After that in Web Services → Project name → Test web services.

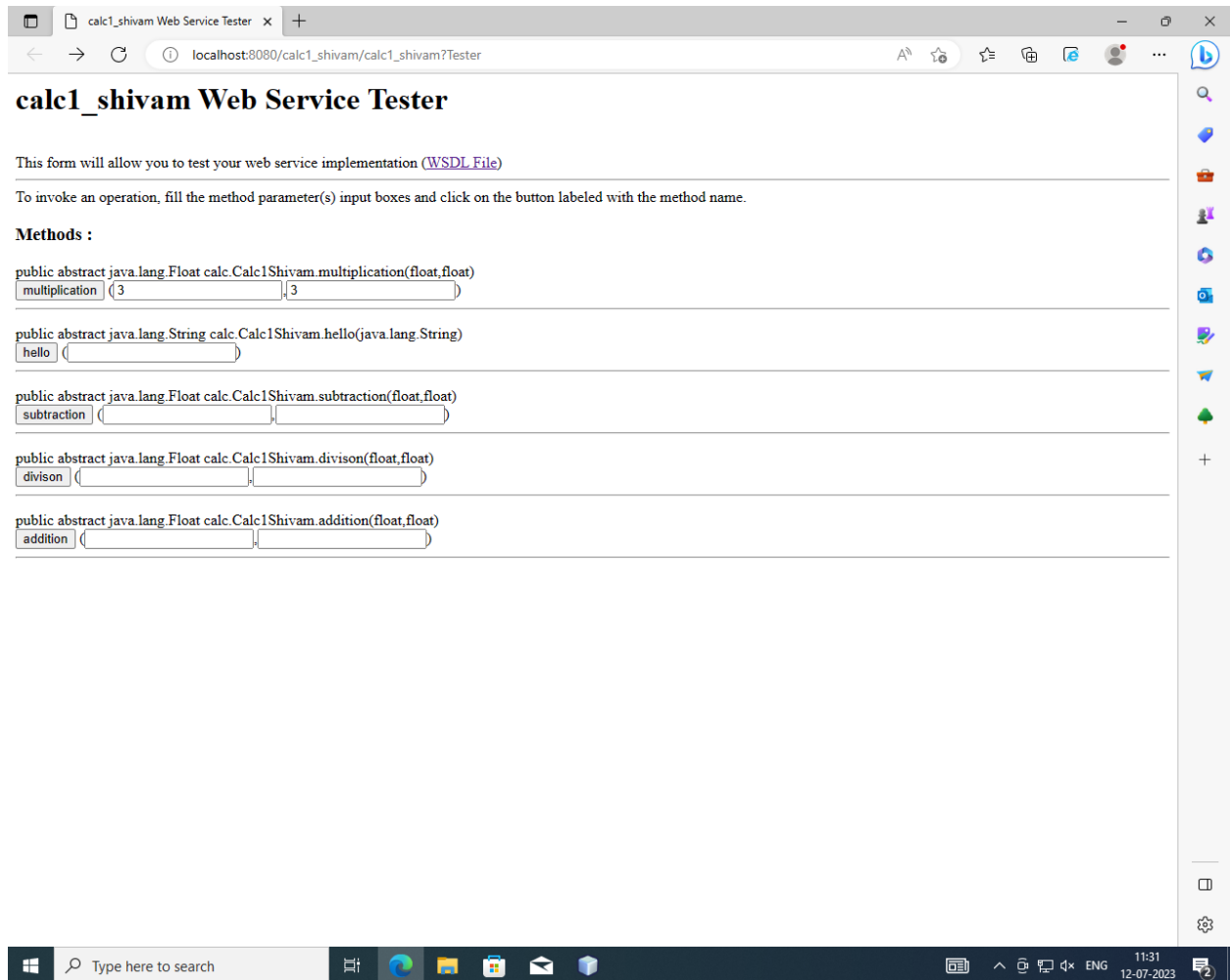
Now we can see output in browser.





Now we can test all the methods and will get output.

Then we can see WSDL file click on that and copy URL. so we can test our service in SOAP UI by that URL.



The screenshot shows a web browser window with the title "calc1\_shivam Web Service Tester". The address bar shows the URL "localhost:8080/calc1\_shivam/calc1\_shivam?Tester". The page content includes a heading "calc1\_shivam Web Service Tester" and a description: "This form will allow you to test your web service implementation (WSDL File)". Below this, it says "To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name." The "Methods :" section lists five methods, each with a label and input boxes:

- multiplication**: `public abstract java.lang.Float calc.Calc1Shivam multiplication(float,float)`. Input boxes contain "3" and "3".
- hello**: `public abstract java.lang.String calc.Calc1Shivam hello(java.lang.String)`. Input box is empty.
- subtraction**: `public abstract java.lang.Float calc.Calc1Shivam subtraction(float,float)`. Input boxes are empty.
- divison**: `public abstract java.lang.Float calc.Calc1Shivam divison(float,float)`. Input boxes are empty.
- addition**: `public abstract java.lang.Float calc.Calc1Shivam addition(float,float)`. Input boxes are empty.

The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system tray on the right shows the time "11:31" and date "12-07-2023".

Method invocation trace x Method invocation trace x +

localhost:8080/calci\_shivam/calci\_shivam?Tester

## multiplication Method invocation

Method parameter(s)

Type	Value
float	3
float	3

Method returned

java.lang.Float : "9.0"

SOAP Request

```
<?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:multiplication xmlns:ns2="http://calci/">
      <e>3.0</e>
      <b>3.0</b>
    </ns2:multiplication>
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?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:multiplicationResponse xmlns:ns2="http://calci/">
      <return>9.0</return>
    </ns2:multiplicationResponse>
  </S:Body>
</S:Envelope>
```

Type here to search

11:31 12-07-2023

localhost:8080/calci\_shivam/calci x +

localhost:8080/calci\_shivam/calci\_shivam?WSDL

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b150120.1832 JAXWS-API/
-->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b150120.1832 JAXWS-API/
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsdp="http://www.w3.org/ns/ws-policy"
  xmlns:wspl_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://calci/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://calci/" name="calci_shivam">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://calci/" schemaLocation="http://localhost:8080/calci_shivam/calci_shivam?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="multiplication">
    <part name="parameters" element="tns:multiplication"/>
  </message>
  <message name="multiplicationResponse">
    <part name="parameters" element="tns:multiplicationResponse"/>
  </message>
  <message name="division">
    <part name="parameters" element="tns:division"/>
  </message>
  <message name="divisionResponse">
    <part name="parameters" element="tns:divisionResponse"/>
  </message>
  <message name="hello">
    <part name="parameters" element="tns:hello"/>
  </message>
  <message name="helloResponse">
    <part name="parameters" element="tns:helloResponse"/>
  </message>
  <message name="subtraction">
    <part name="parameters" element="tns:subtraction"/>
  </message>
  <message name="subtractionResponse">
    <part name="parameters" element="tns:subtractionResponse"/>
  </message>
  <message name="addition">
    <part name="parameters" element="tns:addition"/>
  </message>
  <message name="additionResponse">
    <part name="parameters" element="tns:additionResponse"/>
  </message>
  <portType name="calci_shivam">
    <operation name="multiplication">
      <input wsam:Action="http://calci/calci_shivam/multiplicationRequest" message="tns:multiplication"/>
      <output wsam:Action="http://calci/calci_shivam/multiplicationResponse" message="tns:multiplicationResponse"/>
    </operation>
    <operation name="division">
      <input wsam:Action="http://calci/calci_shivam/divisionRequest" message="tns:division"/>
      <output wsam:Action="http://calci/calci_shivam/divisionResponse" message="tns:divisionResponse"/>
    </operation>
    <operation name="hello">
      <input wsam:Action="http://calci/calci_shivam/helloRequest" message="tns:hello"/>
      <output wsam:Action="http://calci/calci_shivam/helloResponse" message="tns:helloResponse"/>
    </operation>
    <operation name="subtraction">
      <input wsam:Action="http://calci/calci_shivam/subtractionRequest" message="tns:subtraction"/>
      <output wsam:Action="http://calci/calci_shivam/subtractionResponse" message="tns:subtractionResponse"/>
    </operation>
  </portType>
</definitions>
```

Type here to search

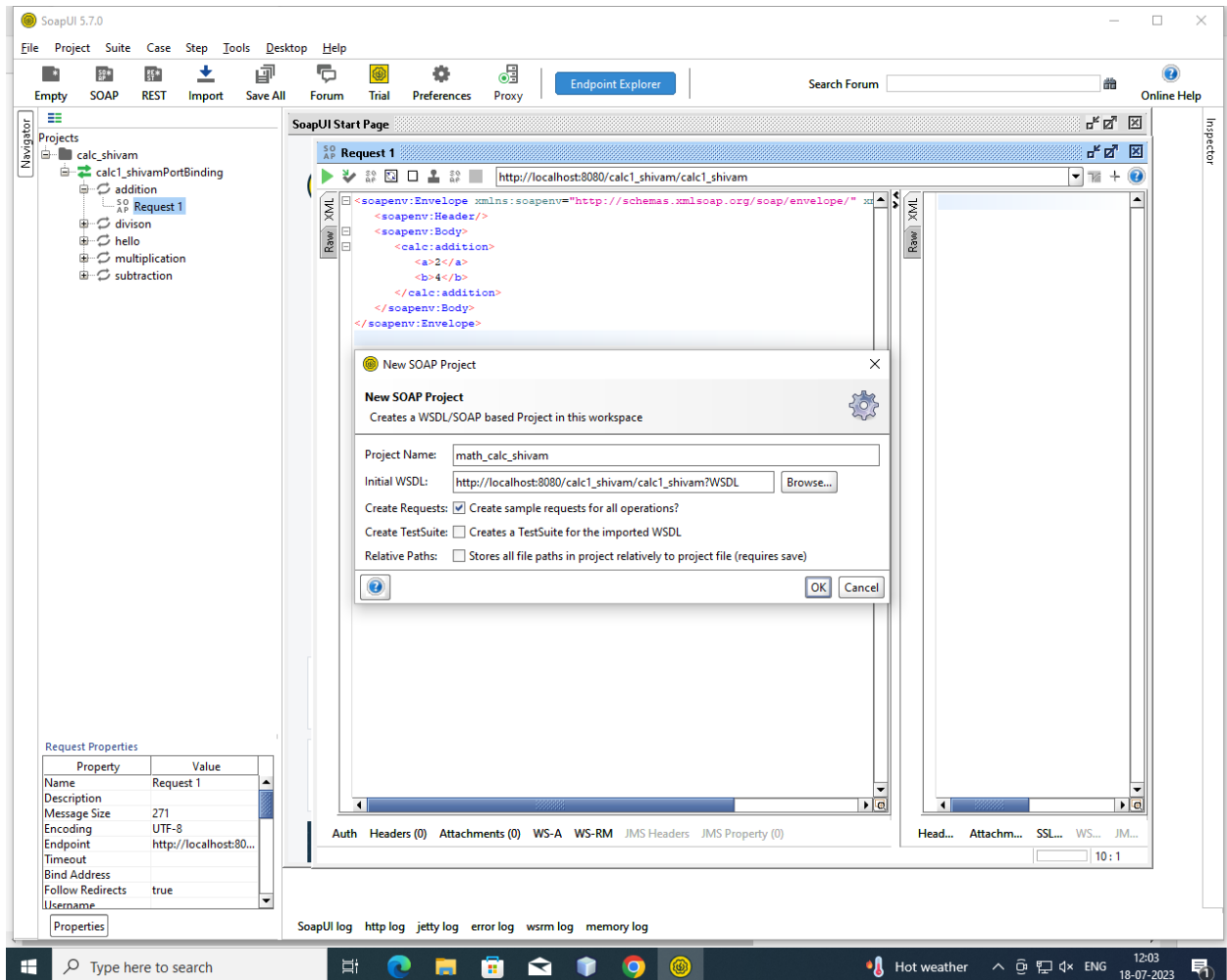
Morning news

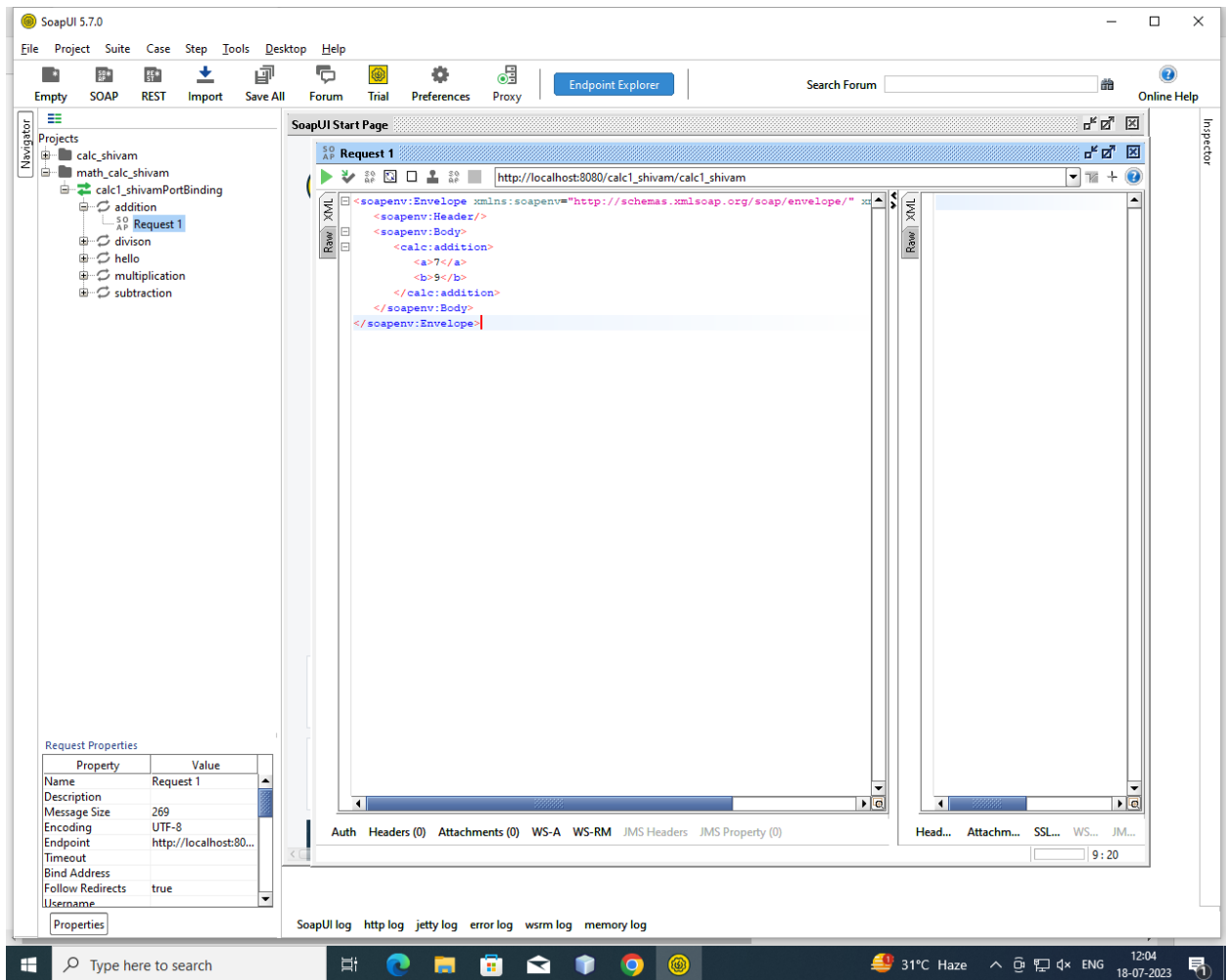
11:41 18-07-2023

Now we test our service in SOAP UI.

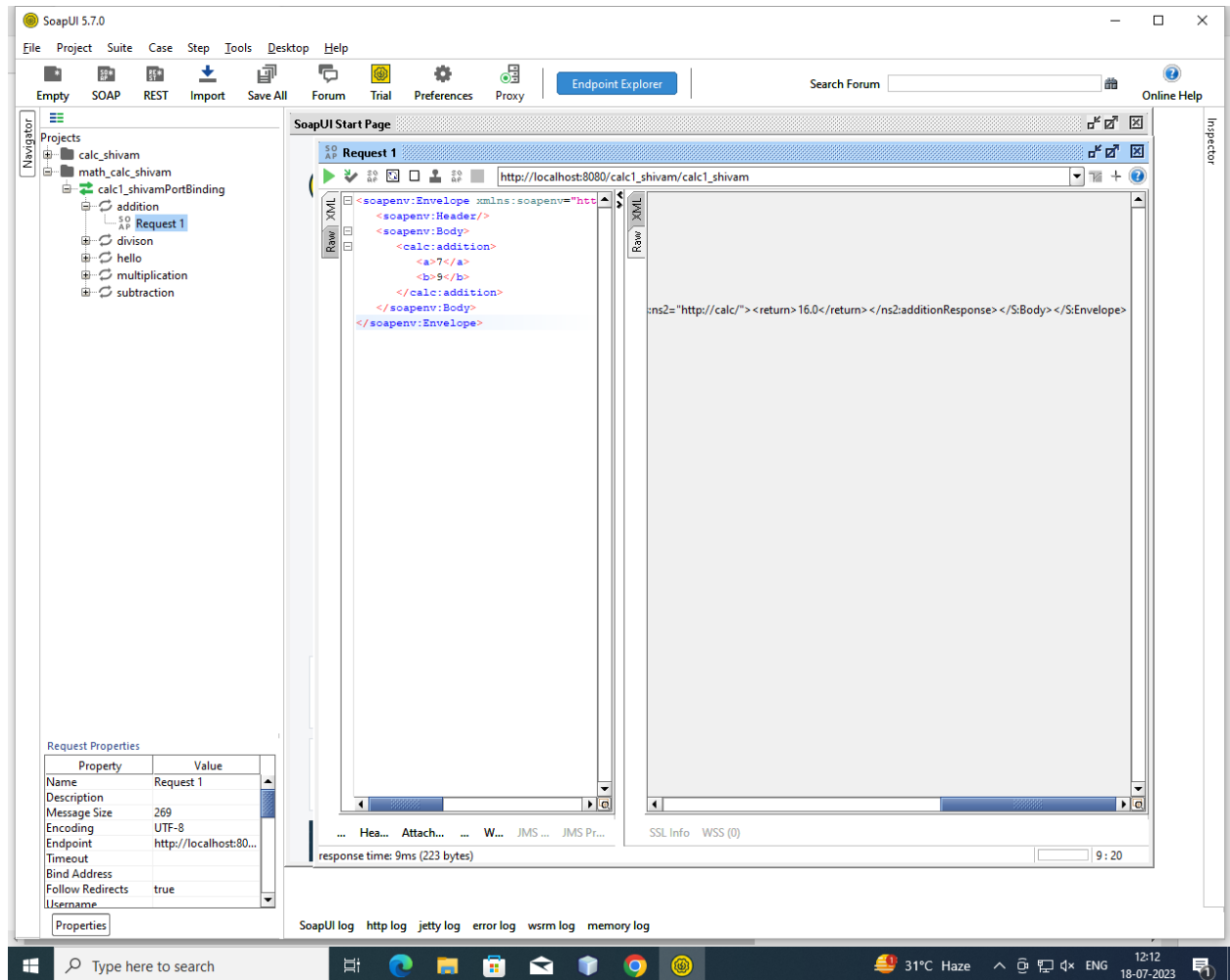
Open SOAP UI. Click on SOAP button and enter project name and in initial WSDL we have to give WSDL file URL. Then click Ok.

In File change a, b values and run the program and we can see output.so our service is working correctly.



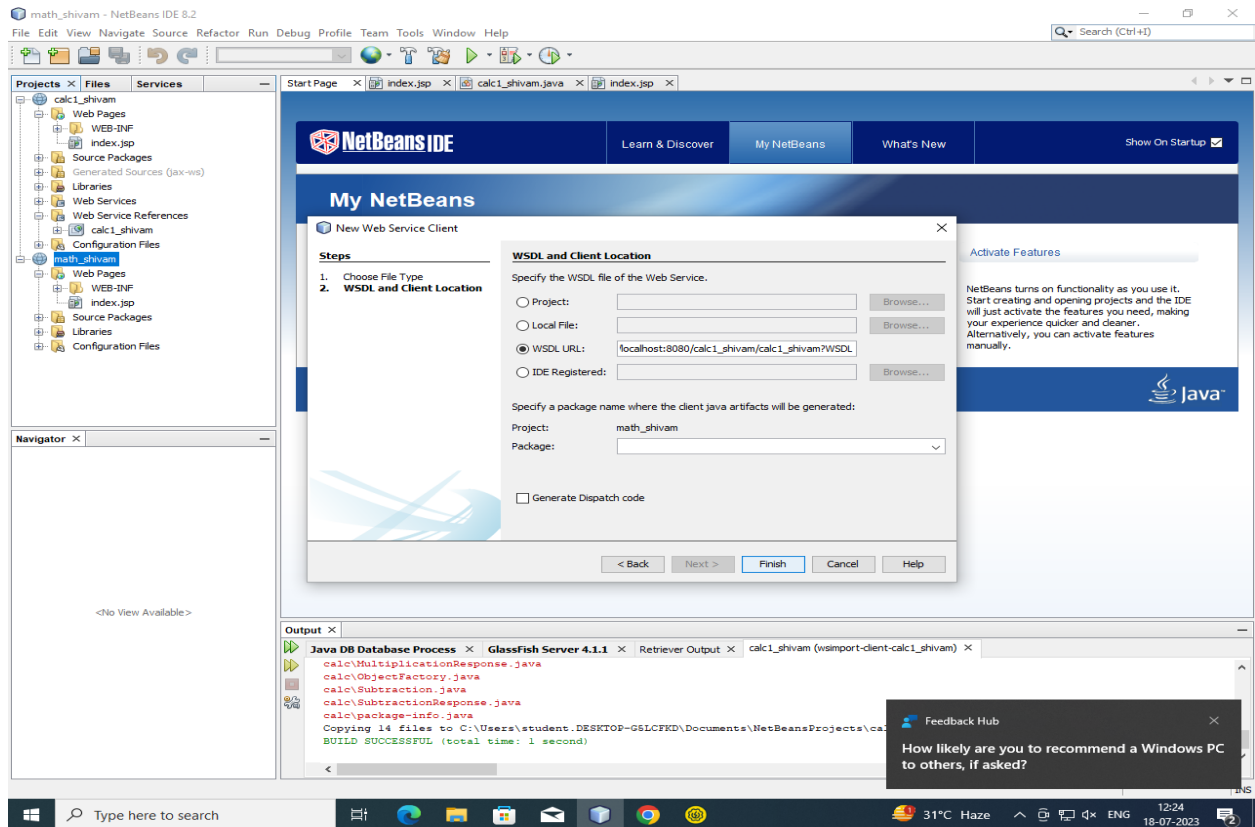
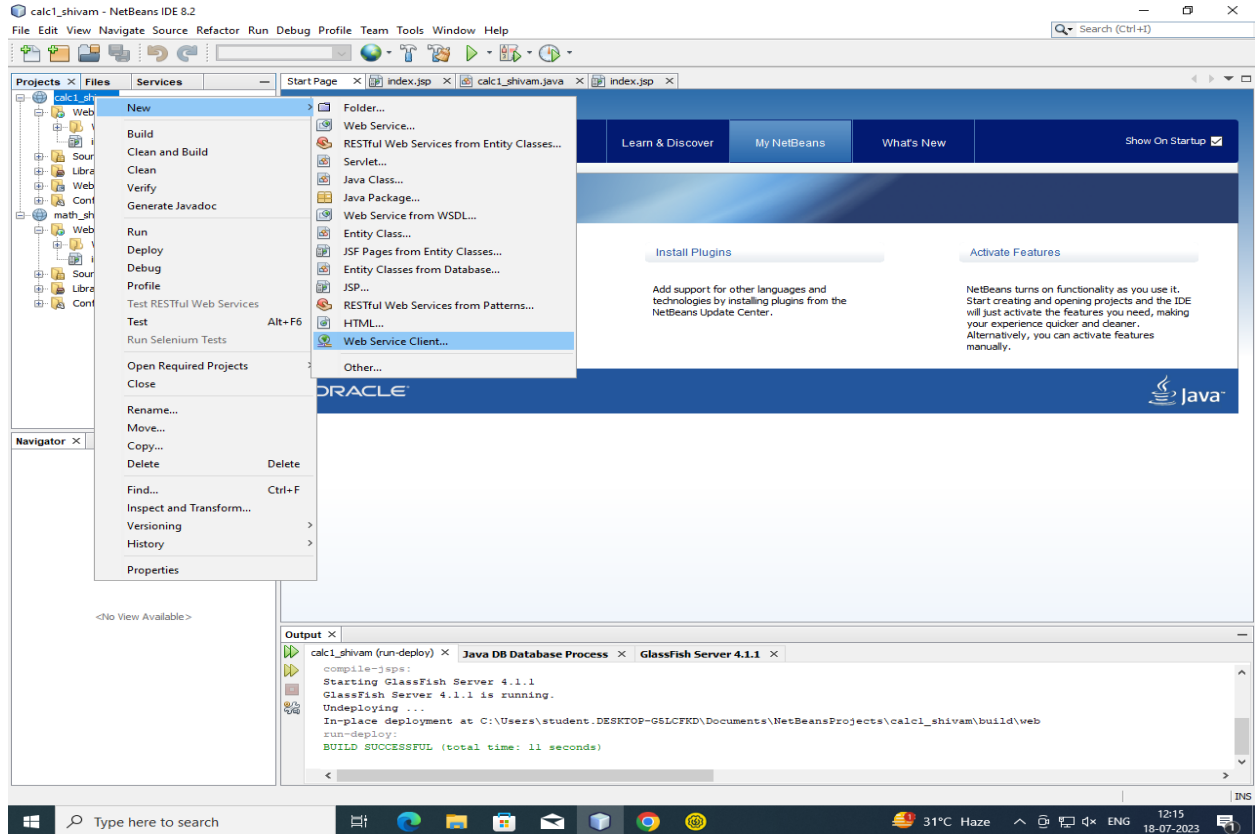


In right side we get our output : 16



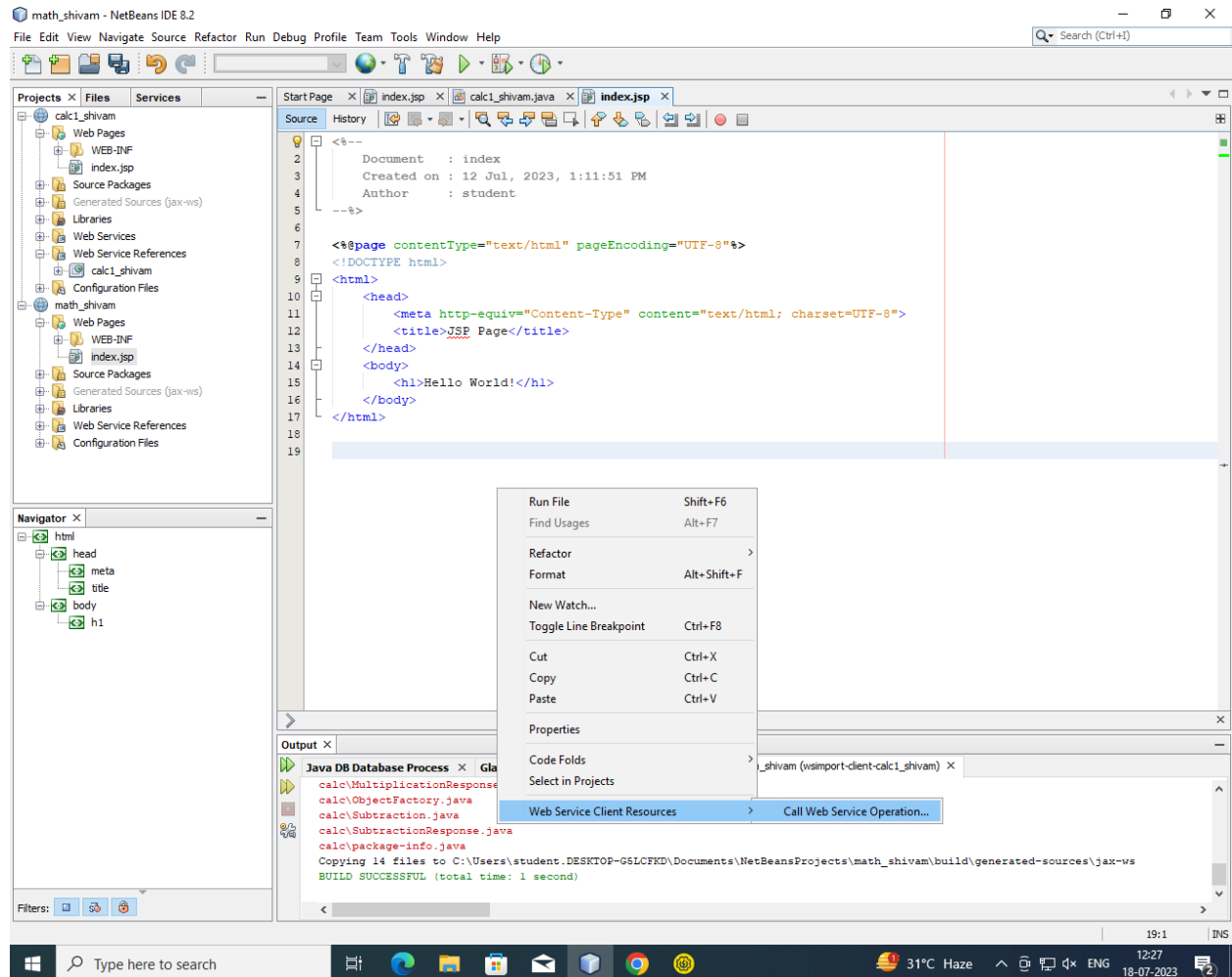
Now in Netbeans, Simply create another project.

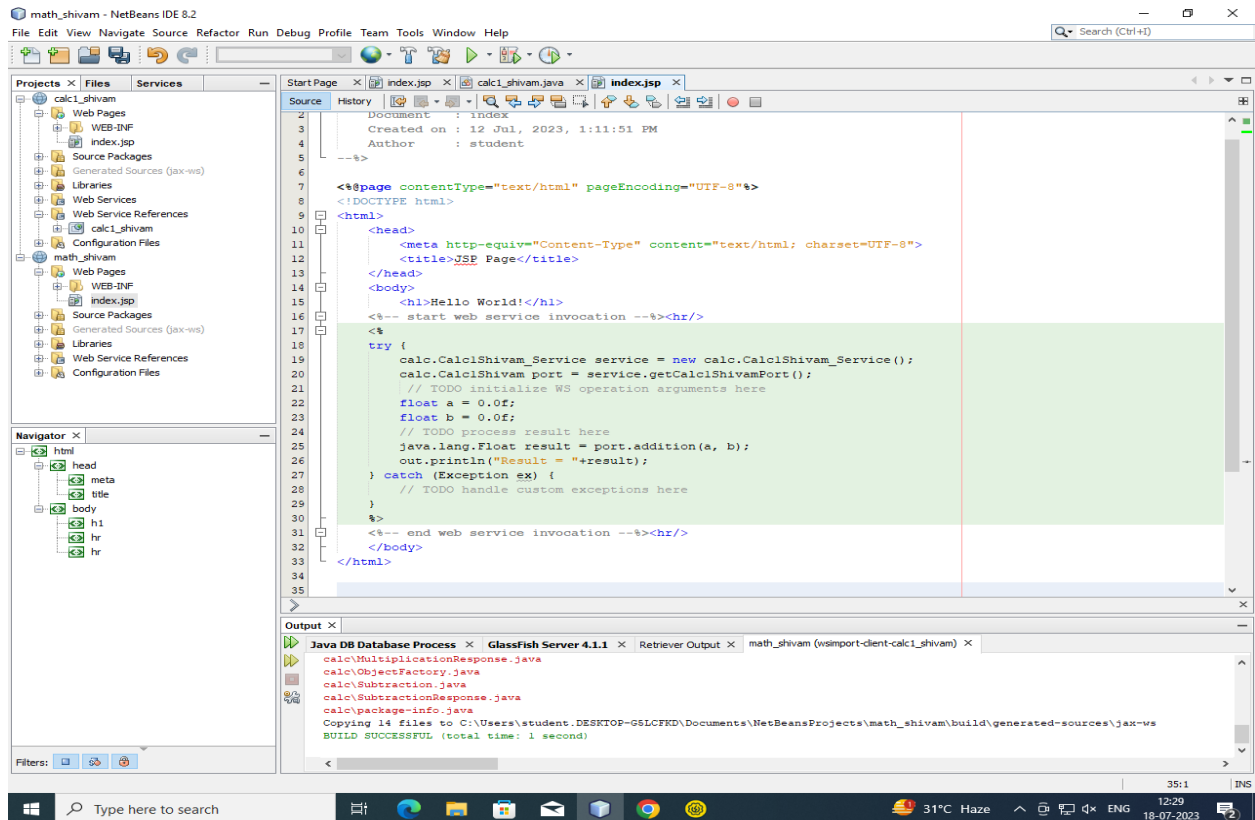
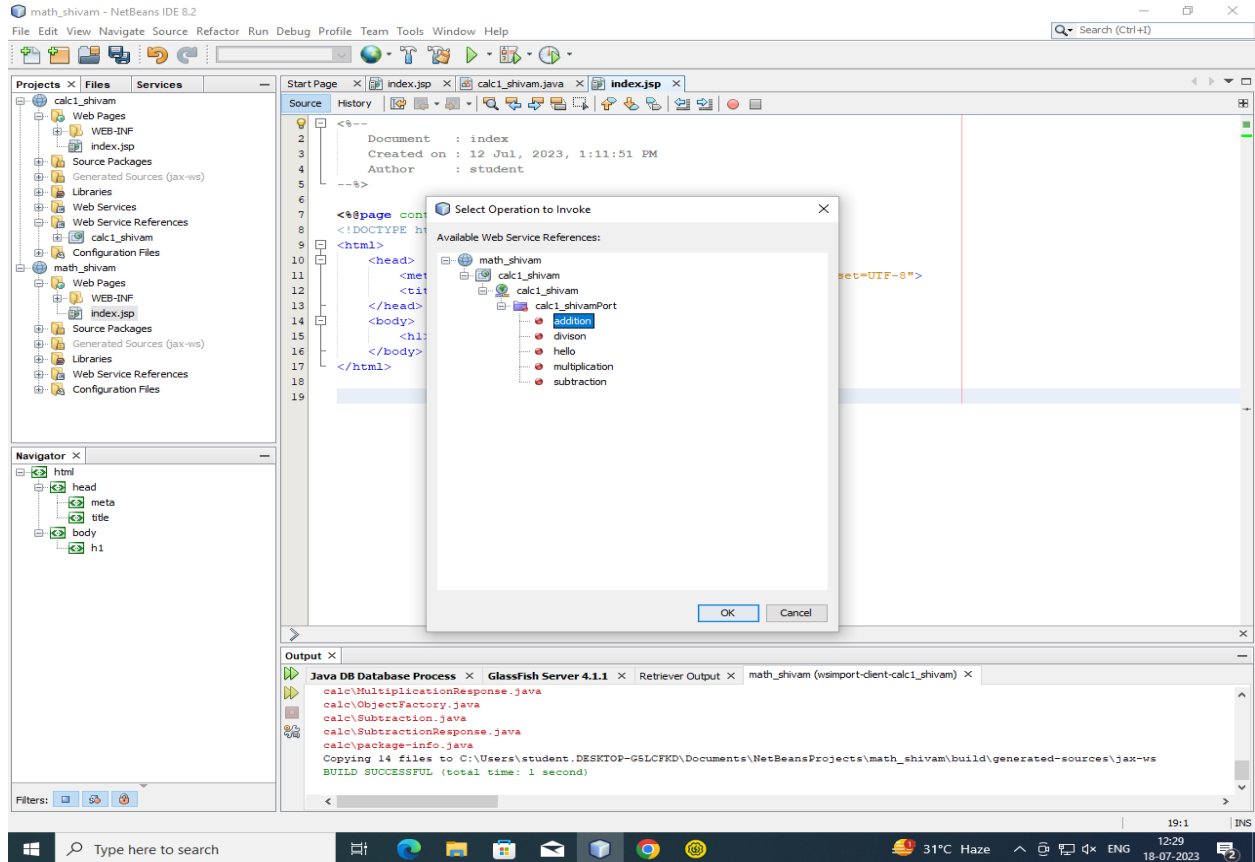
Right click on project name → New → web service client.and enter WSDL url that we have copied previously.





Now open index.jsp file and Right click → web service client resources → Call web service operation. After that click one of them operation click Ok. Then in index.jsp we will get that operation code automatically in java.





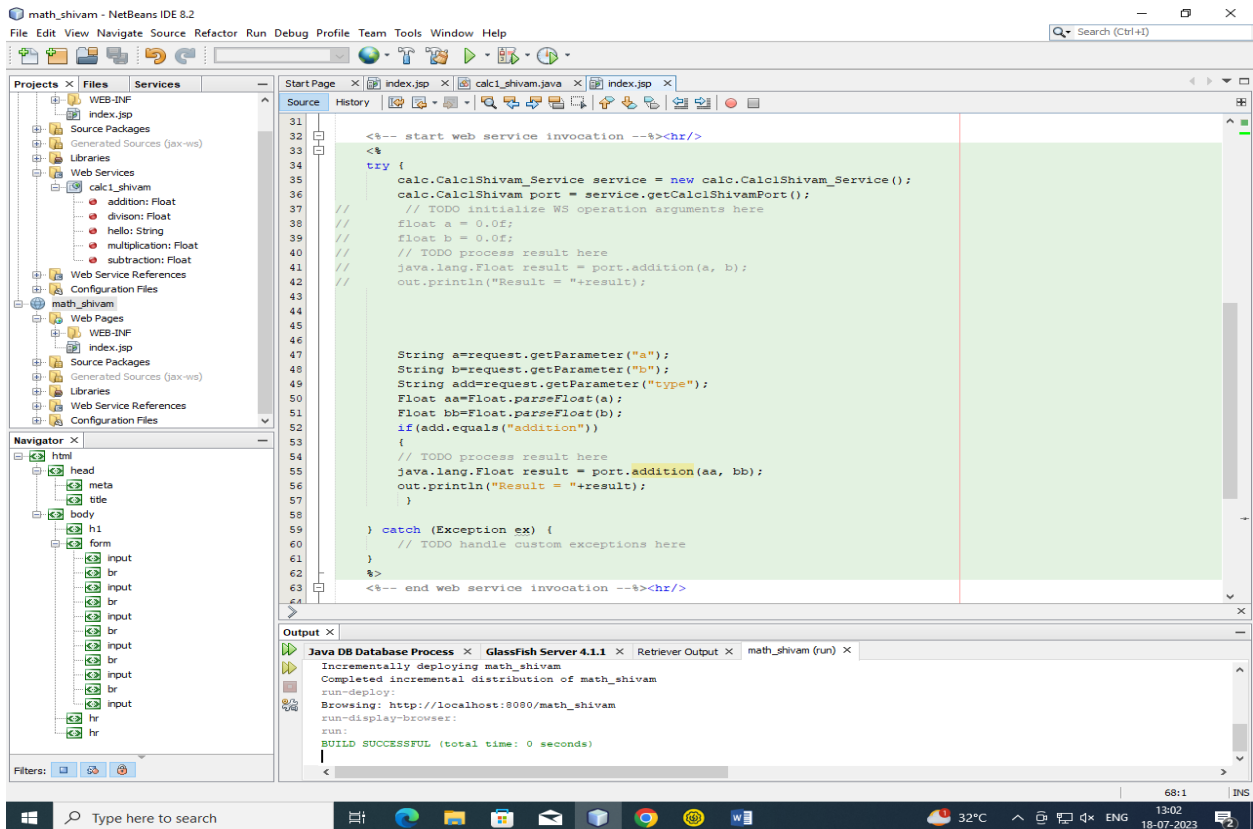
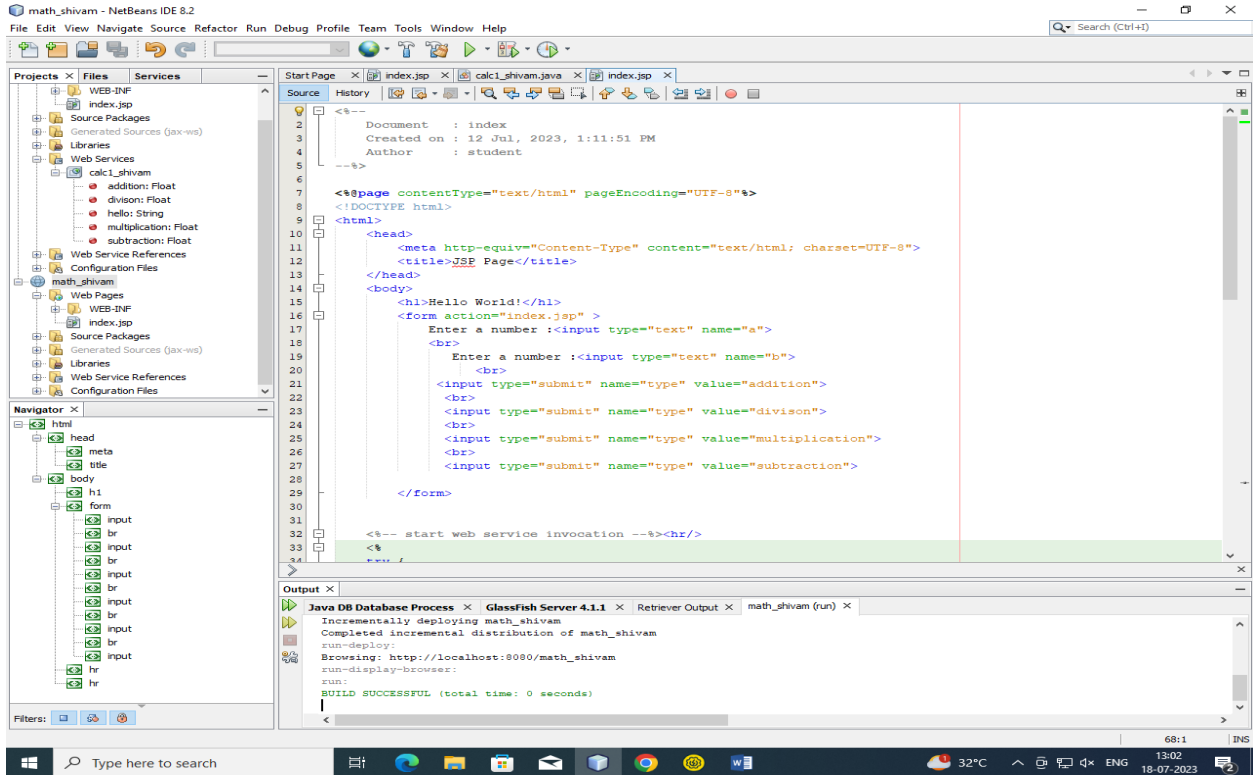
In index.jsp, Add Form part in Body tag.

```
<form action="index.jsp" >
Enter a number <input type="text" name="a"> <br>
Enter a number :<input type="text" name="b" <br>
<input type="submit" name="type" value="addition"> <br>
<br>
<input type="submit" name="type" value="divison">
<input type="submit" name="type" value="multiplication">
<input type="submit" name="type" value="subtraction">
</form>
```

Also in try block add this

```
calc. Calc1Shivam Service service = new calc. CalciShivam Service () ;
calc. CalciShivam port = service.getCalc1ShivamPort () ;
```

```
String a=request.getParameter ("a");
String b=request.getParameter ("b");
String add=request. getParameter ("type");
Float aa=Float.parseFloat (a) ;
Float bb=Float.parseFloat (b) ;
if (add.equals ("addition")) {
java.lang. Float result = port.addition (aa, bb);
out.println ("Result =" +result) ;
}
```



Now Right click on project name and click Deploy then after successfully deployed. Right click on project name and Click on run. And we will get output.

