

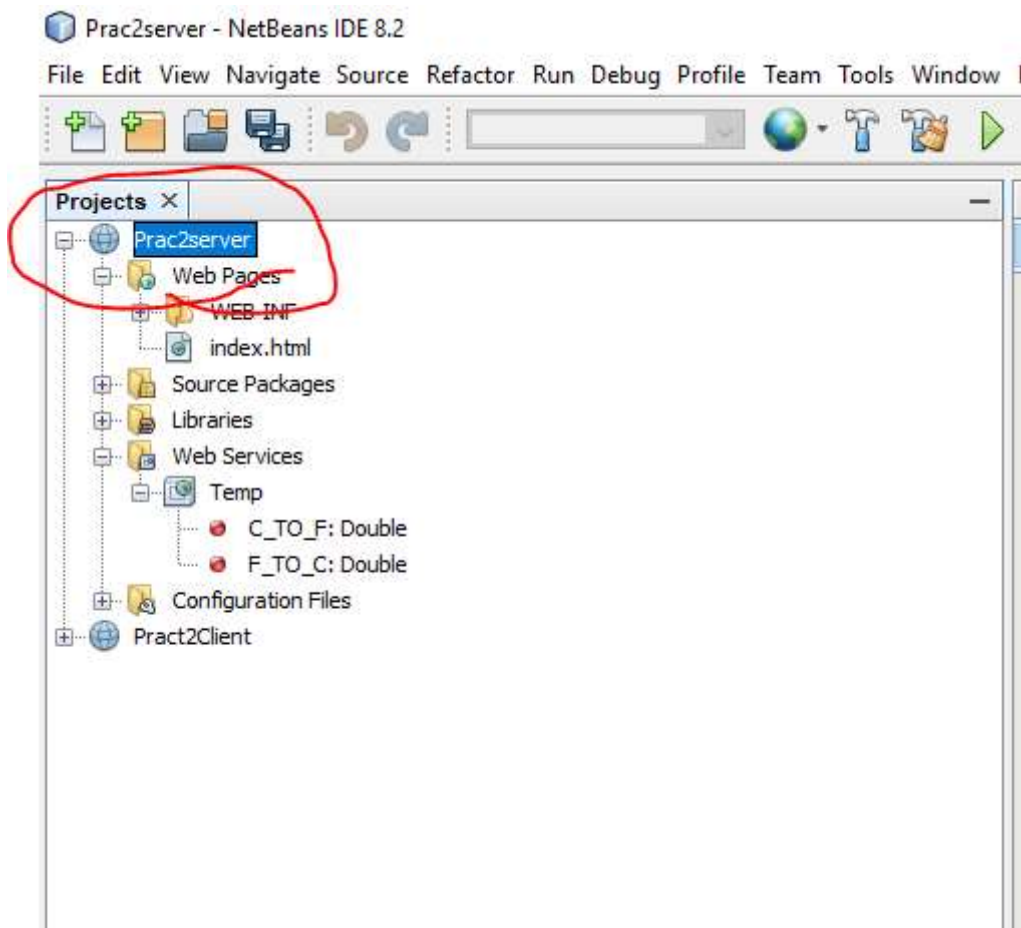
Practical No.2

Aim:- Create a Simple SOAP Web Service.

Steps : Client Side

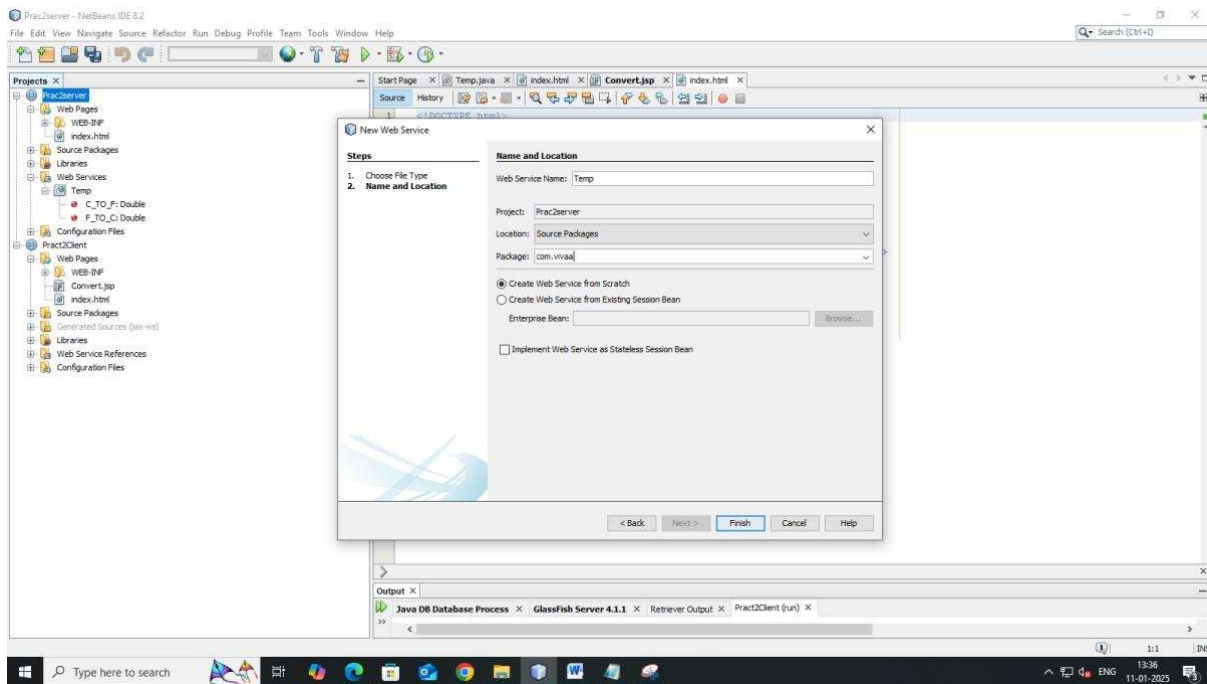
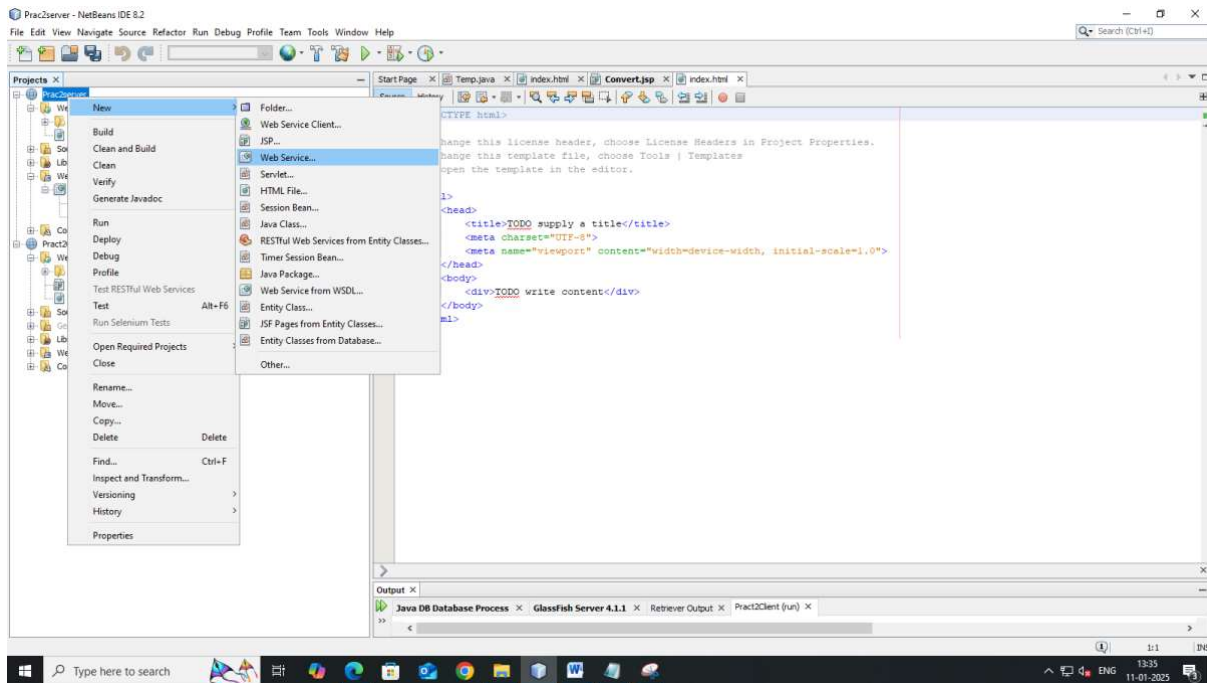
1)Open Netbeans 8.2

2)Click file > New Project > Select Java Web > next > name project name(Prac2Server) > next > next > Finish



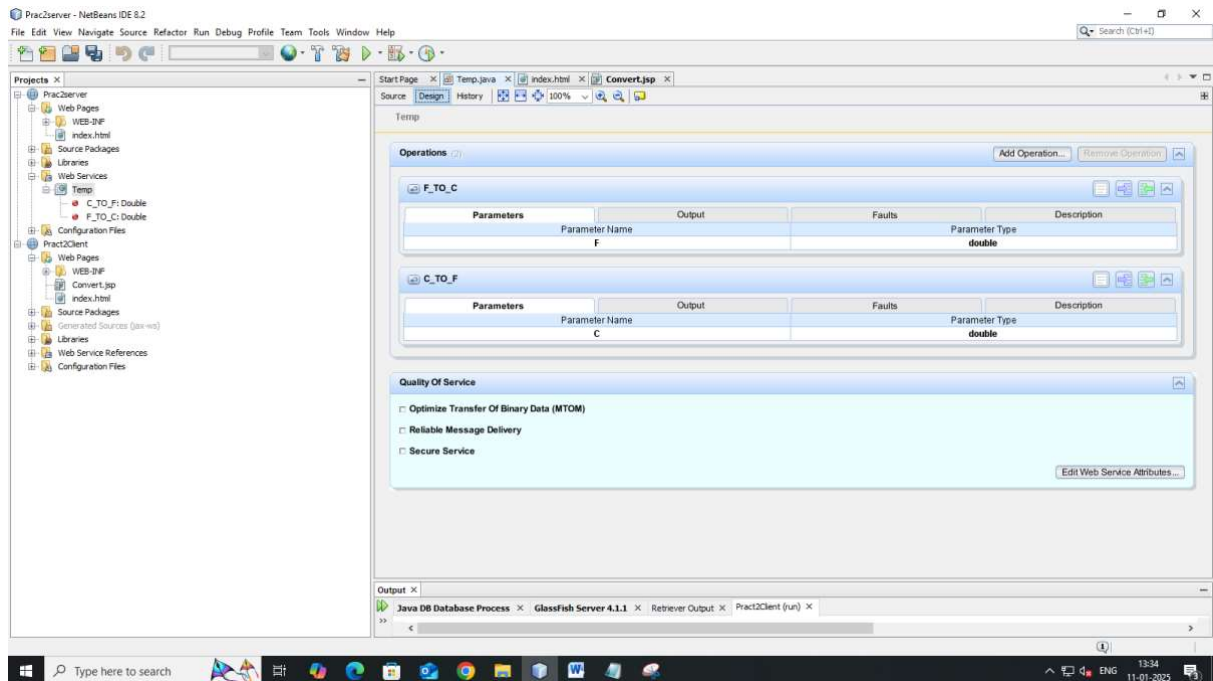
3)To create Web service :

Click prj name(Prac2Server) > new > web service > web service name(Temp) and package name(com.viva) > Finish



Here we created 2 Operations

- a) F_To_C
- b) C_To_F



We added the following code for Conversion of temperatures from Celcius to Farenheit and vice versa.

Code:

```
/*
```

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

```
*/
```

```
package com.viva;
```

```
import javax.jws.WebService;
```

```
import javax.jws.WebMethod;
```

```
import javax.jws.WebParam;
```

```
/**
```

```
*
```

```
* @author student
```

```
*/
```

```
@WebService(serviceName = "Temp")
```

```
public class Temp {
```

```
/**
```

```
* Web service operation
```

```
*/
```

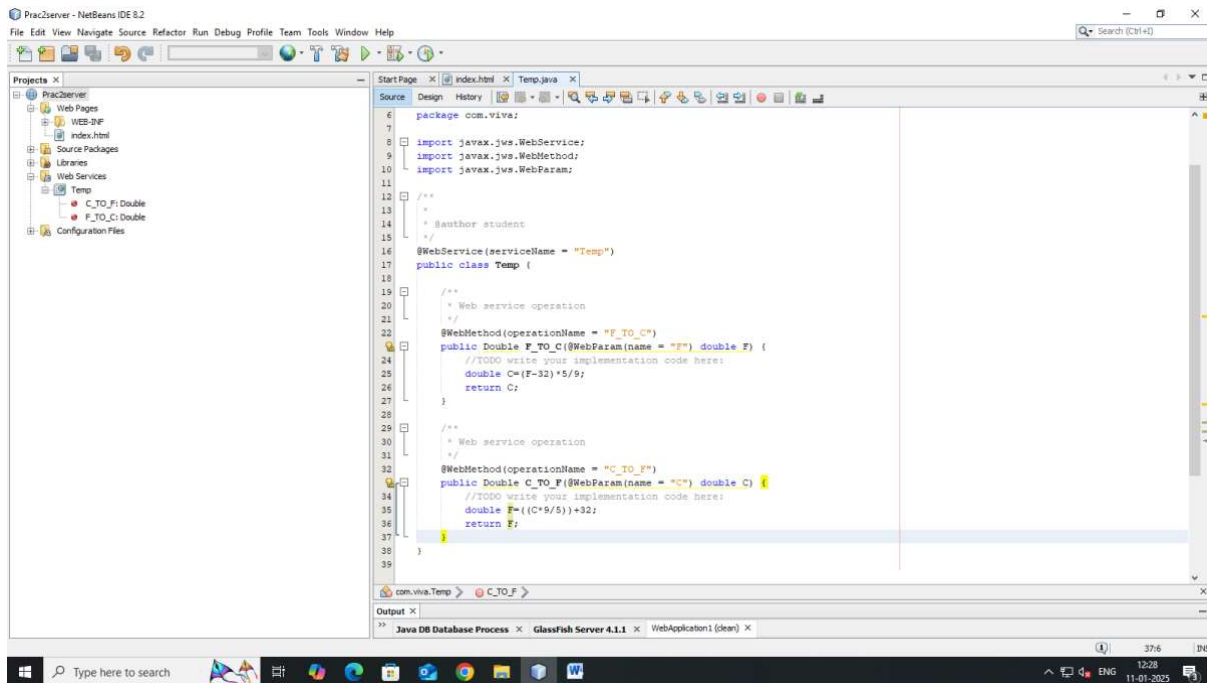
```
@WebMethod(operationName = "F_TO_C")
```

```
public Double F_TO_C(@WebParam(name = "F") double F) {
```

```
//TODO write your implementation code here:
double C=(F-32)*5/9;
return C;
}

/**
 * Web service operation
 */
@WebMethod(operationName = "C_TO_F")
public Double C_TO_F(@WebParam(name = "C") double C) {
    //TODO write your implementation code here:
    double F=((C*9/5))+32;
    return F;
}
}

<html>
<head>
<title>TODO supply a title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<form action="Convert.jsp" method="post">
<input type="text" name="t1">
<input type="submit" value="show">
</form>
</body>
</html>
```



Build and deploy

The image shows a development environment with NetBeans IDE and a web browser. The NetBeans IDE window is titled "Prac2server - NetBeans IDE 8.2" and displays a Java source file named "Temp.java". The code defines a web service package "com.viva" with a class "Temp" that implements a method "F_TO_C". The IDE also shows a project explorer on the left and an output window at the bottom right displaying the build process for "Glassfish Server 4.1.1".

The web browser window is titled "Temp Web Service Tester" and shows the URL "localhost:8080/Prac2server/TempTester". The page content includes a title "Temp Web Service Tester", a description of the form's purpose, and two input fields for testing the "F_TO_C" and "F_TO_C" methods.

```
package com.viva;

import javax.ws.rs.WebService;
import javax.ws.rs.WebMethod;
import javax.ws.rs.WebParam;

/**
 *
 * @author student
 */
@WebService(serviceName = "Temp")
public class Temp {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "F_TO_C")
    public Double F_TO_C(@WebParam(name = "F") double F) {
        //TODO write your implementation code here:
        double C=(F-32)*5/9;
        return C;
    }
}
```

Java DB Database Process x Glassfish Server 4.1.1 x WebApplication1 (clean) x

```
ant -f D:\hashita\WebApplication1 clean
init:
undeploy-clean:
deploy-clean:
do-clean:
check-clean:
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Temp Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.Double com.viva.Temp.cTOF(double)
cTOF (10)

public abstract java.lang.Double com.viva.Temp.fTOC(double)
fTOC (25)

The image displays two screenshots of a web browser window showing SOAP method invocation traces. The browser's address bar indicates the URL is `localhost:8080/Prac2server/TempTTester`.

Top Screenshot: cTOF Method invocation

Method parameter(s)

Type	Value
double	10

Method returned

java.lang.Double : "50.0"

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:C_TO_F xmlns:ns2="http://viva.com/">
      <C>10.0</C>
    </ns2:C_TO_F>
  </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:C_TO_FResponse xmlns:ns2="http://viva.com/">
      <return>50.0</return>
    </ns2:C_TO_FResponse>
  </S:Body>
</S:Envelope>
```

Bottom Screenshot: fTOC Method invocation

Method parameter(s)

Type	Value
double	25

Method returned

java.lang.Double : "-3.5555555555555559"

SOAP Request

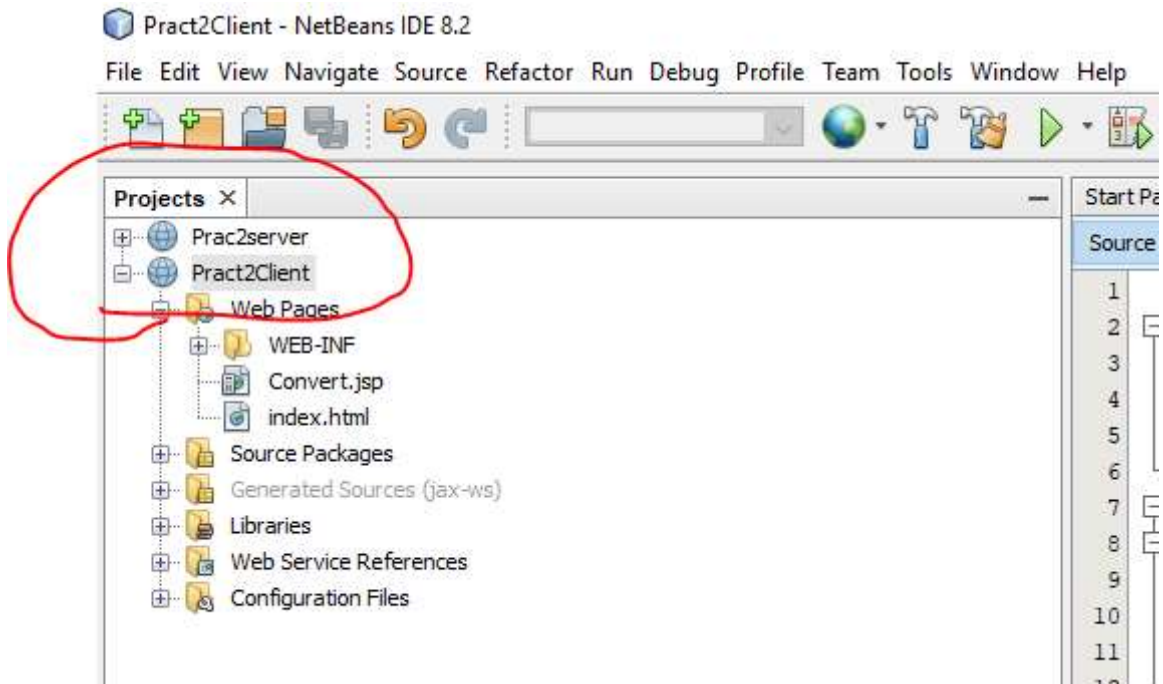
```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:F_TO_C xmlns:ns2="http://viva.com/">
      <F>25.0</F>
    </ns2:F_TO_C>
  </S:Body>
</S:Envelope>
```

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:F_TO_CResponse xmlns:ns2="http://viva.com/">
      <return>-3.5555555555555559</return>
    </ns2:F_TO_CResponse>
  </S:Body>
</S:Envelope>
```

Client Side :

1)Click file > New Project > Select Java Web > next > name project name(Prac2Client) > next > next > Finish



2) Add the following code in index.html of (Prac2Client)

```
<body>
```

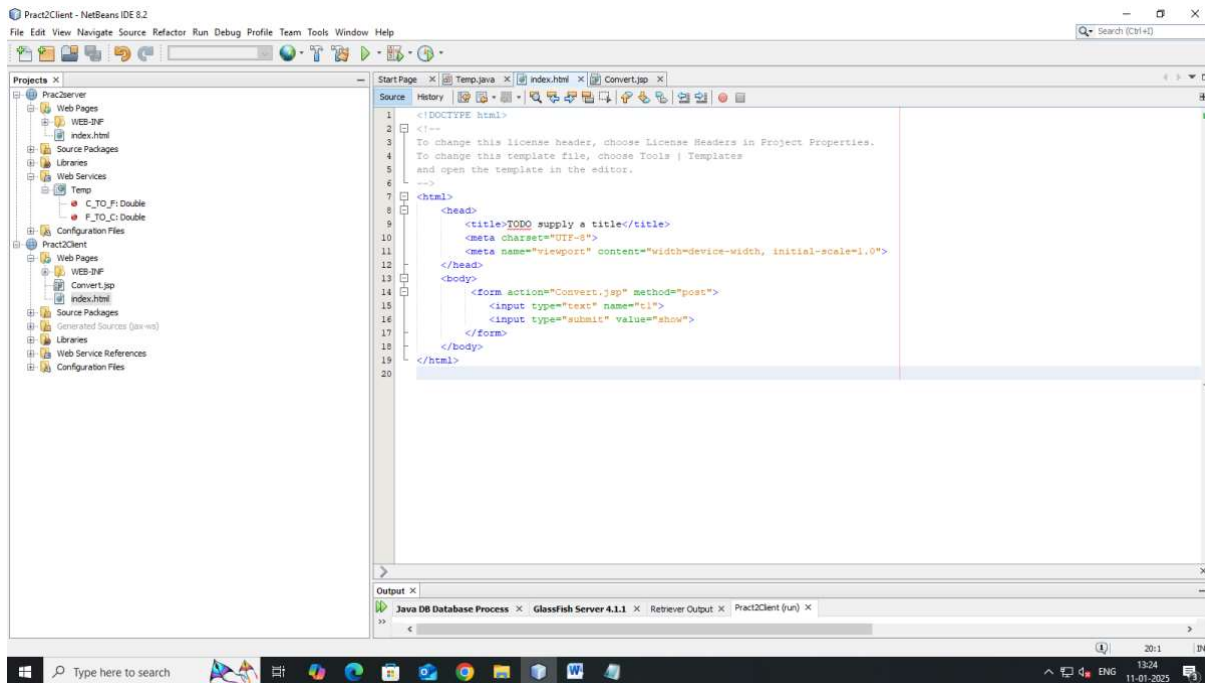
```
  <form action="Convert.jsp" method="post">
```

```
    <input type="text" name="t1">
```

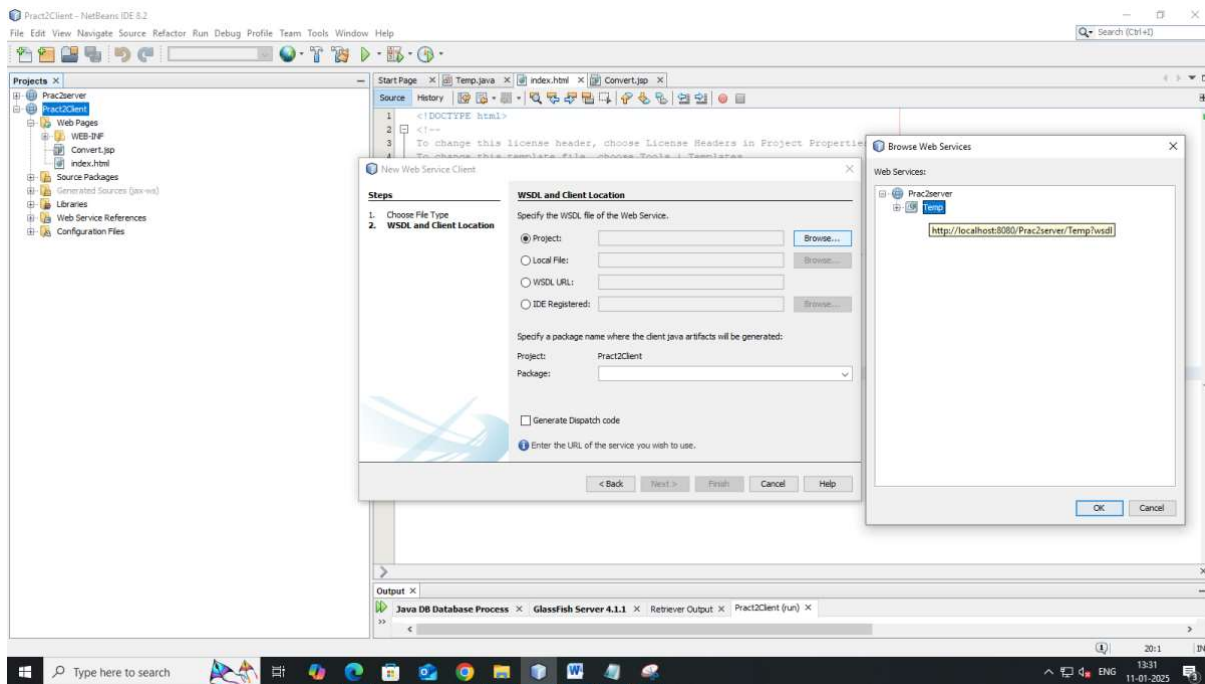
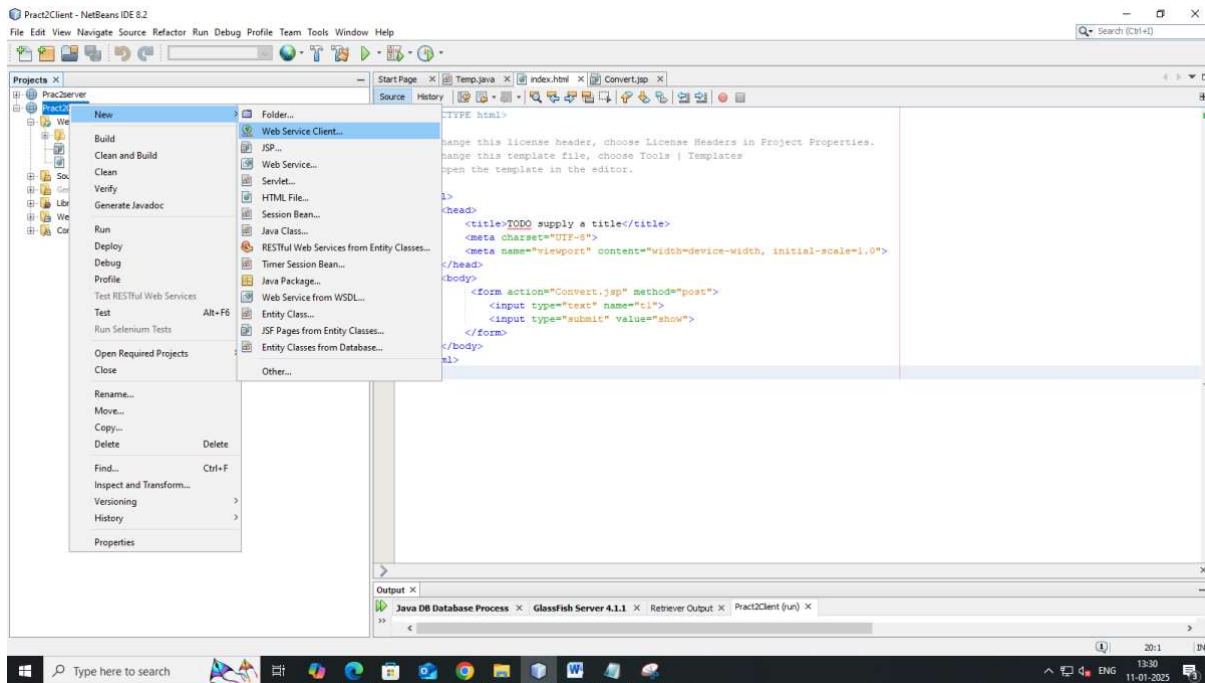
```
    <input type="submit" value="show">
```

```
  </form>
```

```
</body>
```

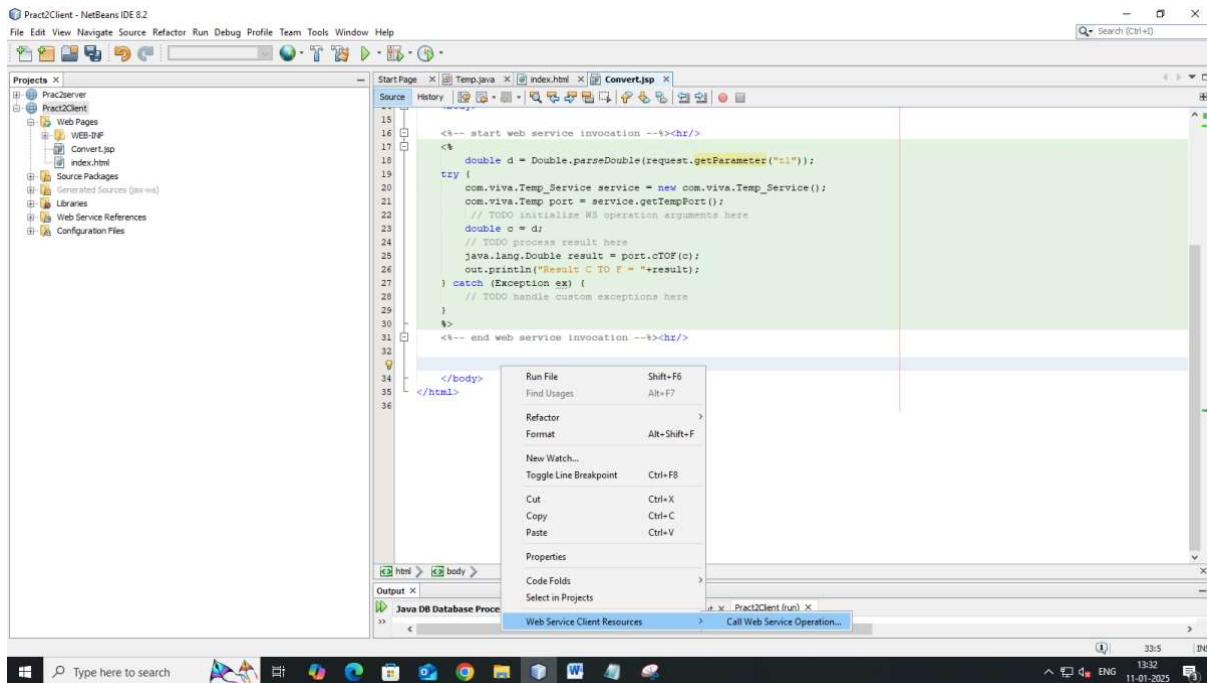



3) Click on Prac2Client > new > web service client > browse the the temp

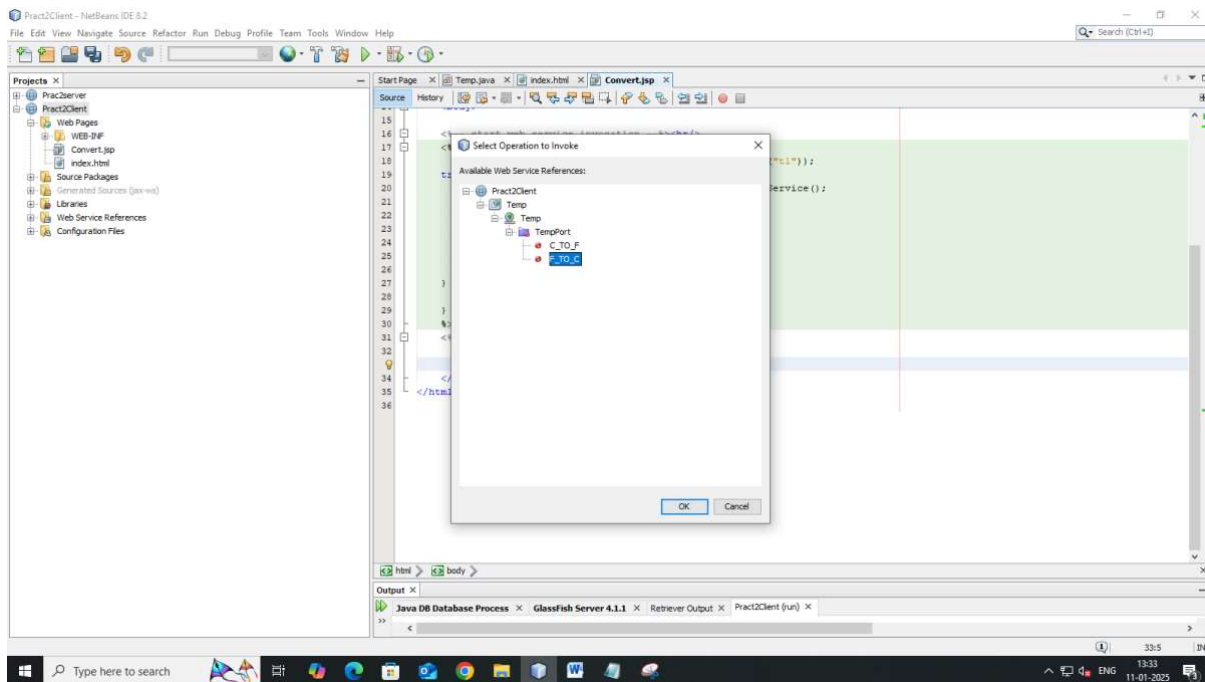
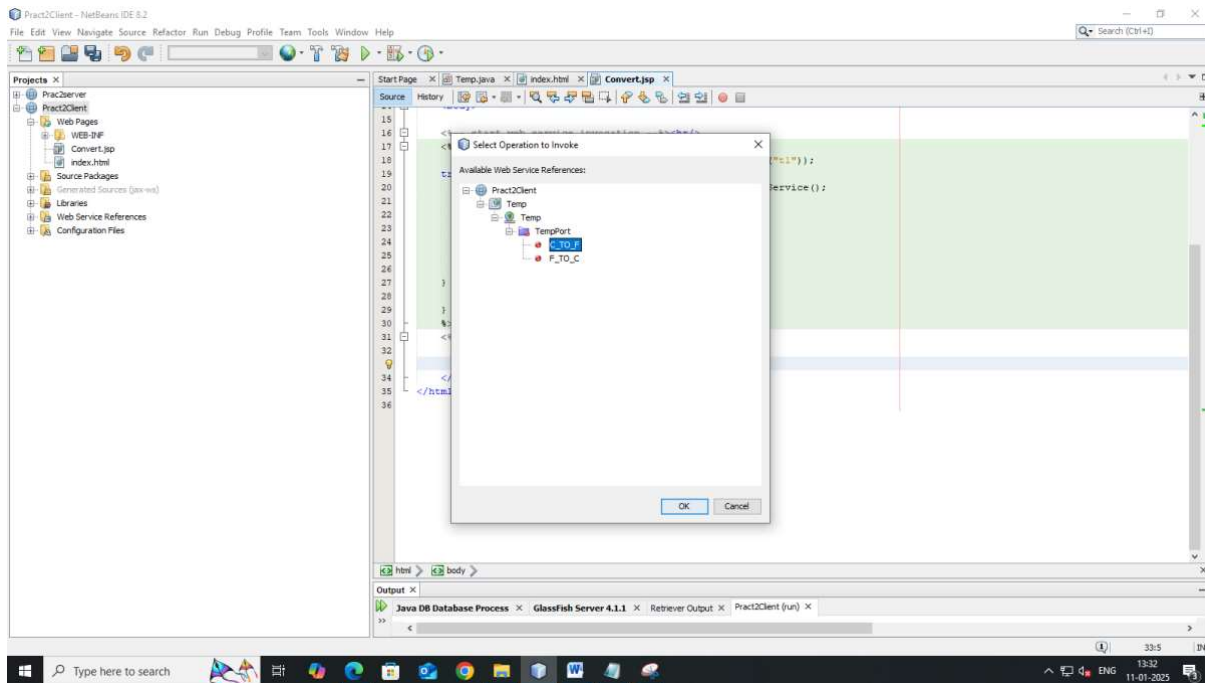


4)create a jsp page(Convert.jsp) (Remove the hello world)

Right click in the body area > web service client resources >call web service operation



Add both (F_TO_C) and (C_TO_F)



```
<%
```

```
double d = Double.parseDouble(request.getParameter("t1"));
```

```
try {
```

```
com.viva.Temp_Service service = new com.viva.Temp_Service();
```

```
com.viva.Temp port = service.getTempPort();
```

```
// TODO initialize WS operation arguments here
```

```
        double c = d;

        // TODO process result here

        java.lang.Double result = port.cTOF(c);

        out.println("Result C TO F = "+result);
    } catch (Exception ex) {

        // TODO handle custom exceptions here

    }

    try {

        com.viva.Temp_Service service = new com.viva.Temp_Service();

        com.viva.Temp port = service.getTempPort();

        // TODO initialize WS operation arguments here

        double f = d;

        // TODO process result here

        java.lang.Double result = port.fTOC(f);

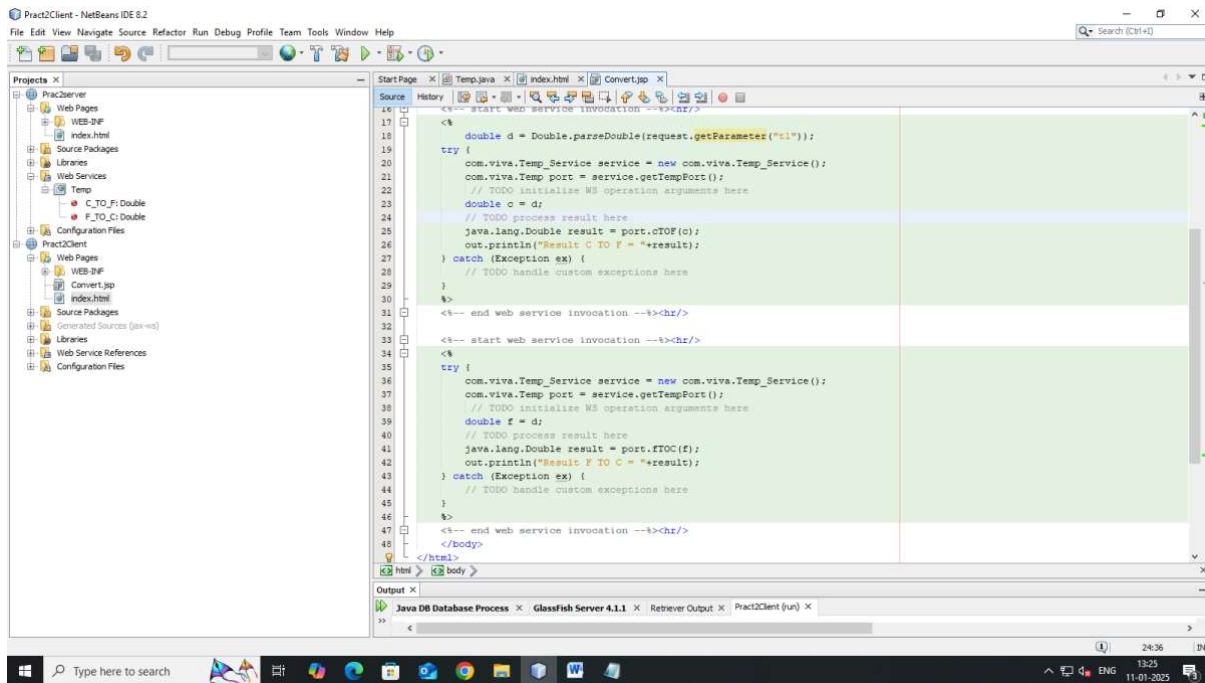
        out.println("Result F TO C = "+result);
    } catch (Exception ex) {

        // TODO handle custom exceptions here

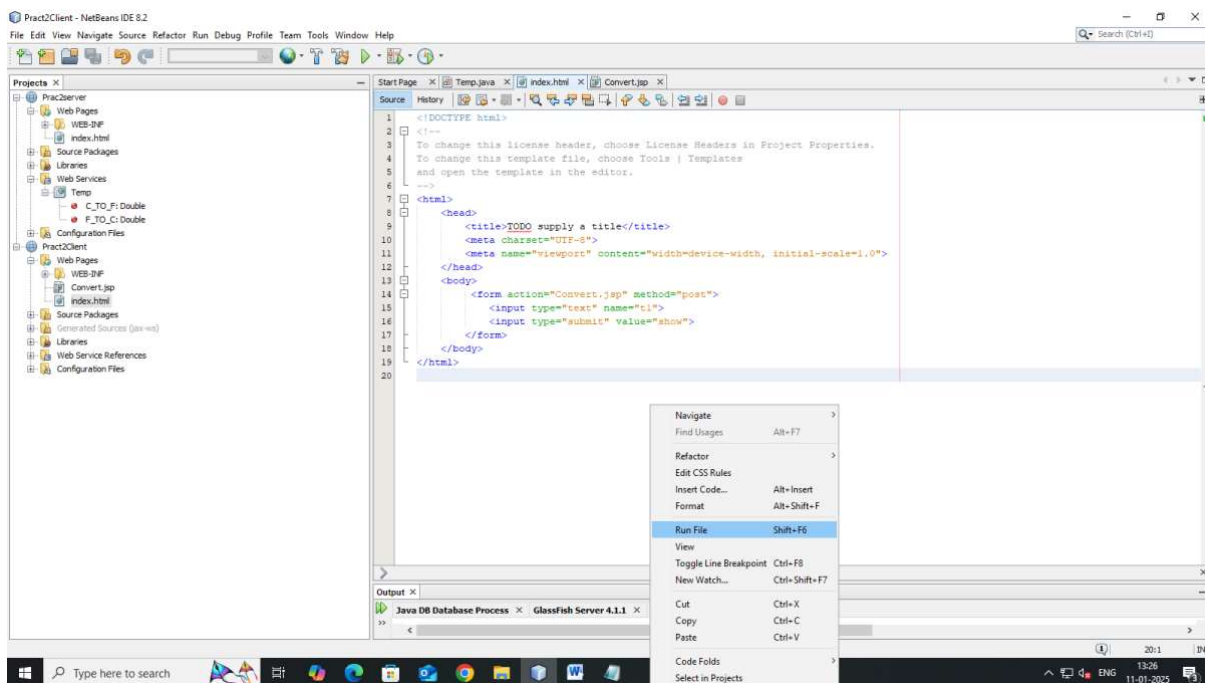
    }

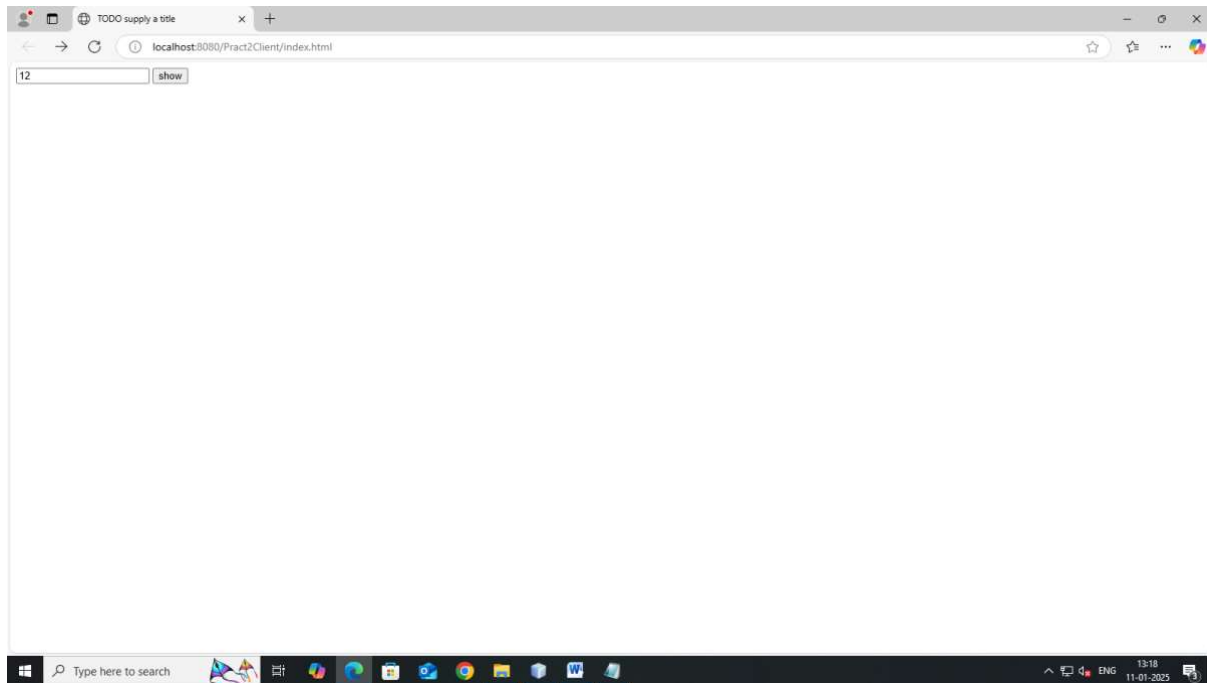
</body>

</html>
```



4) Run the index .html file (right click the and run)





Output:

