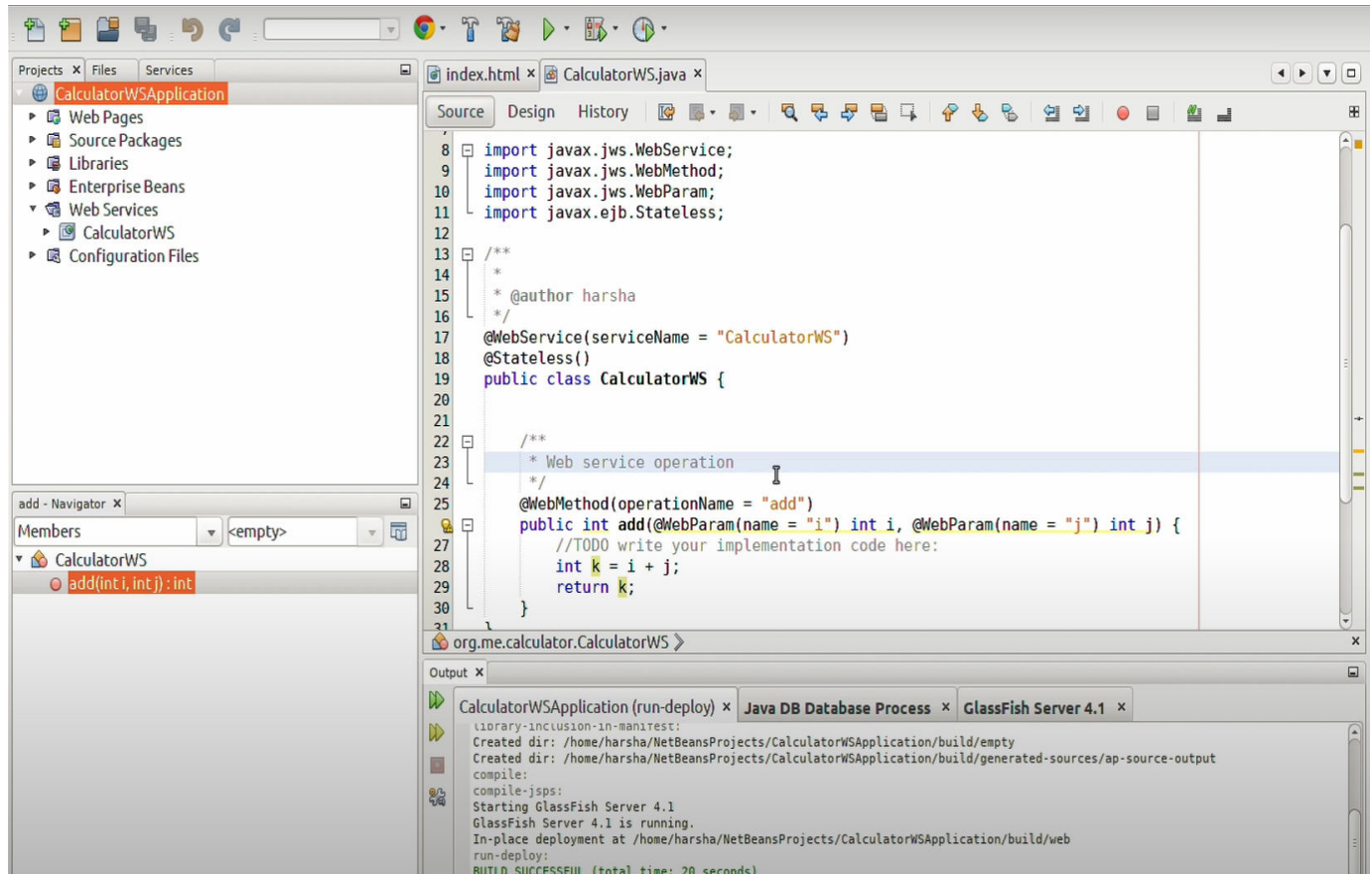
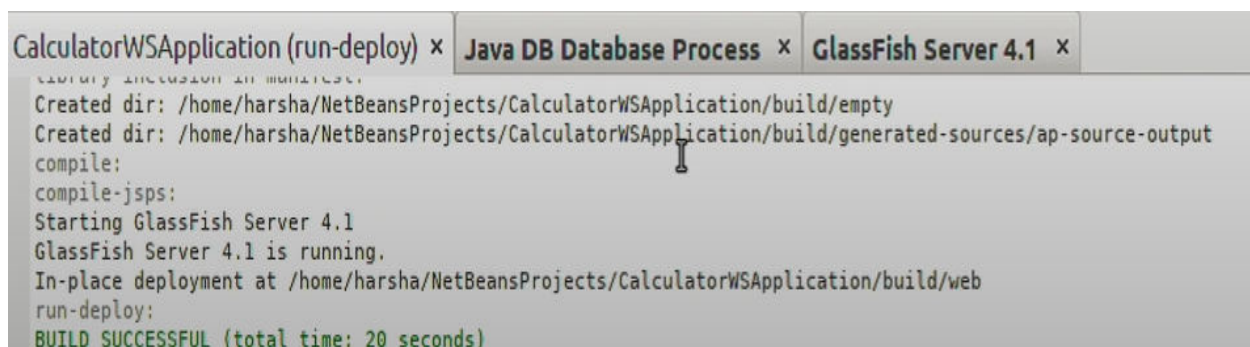


web service

Creating web service:



Deploying Web Services:



Assignment 7

Web Service to perform Addition:

localhost:8080/CalculatorWS/CalculatorWS?Tester

CalculatorWS 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 int org.me.calculator.CalculatorWS.add(int,int)

add (5 7)

localhost:8080/CalculatorWS/CalculatorWS?Tester

add Method invocation

Method parameter(s)

Type	Value
int	5
int	7

Method returned

int : "12"

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:add xmlns:ns2="http://calculator.me.org/">
      <i>5</i>
      <j>7</j>
    </ns2:add>
  </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:addResponse xmlns:ns2="http://calculator.me.org/">
      <return>12</return>
    </ns2:addResponse>
  </S:Body>
```

Consumer Code:

```

12 public class CalculatorWS_Client_Application {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19     }
20
21     private static int add(int i, int j) {
22         org.me.calculator.CalculatorWS_Service service = new org.me.calculator
23         org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
24         return port.add(i, j);
25     }
26
27
28
29
30
31 }
32

```

Distributed application to consume the web service

