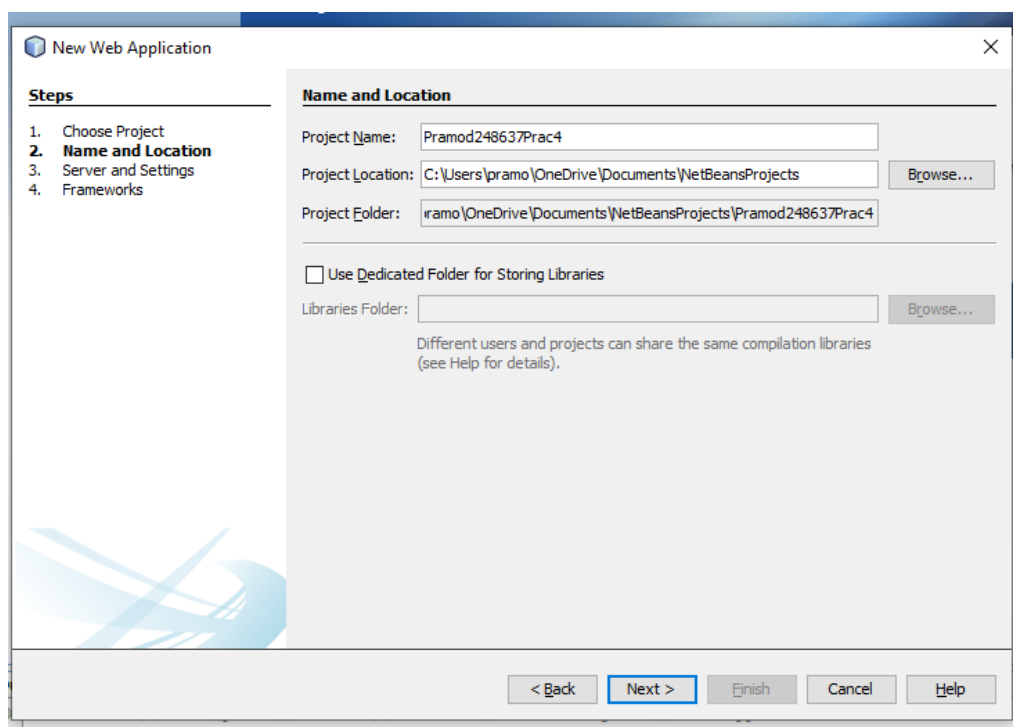
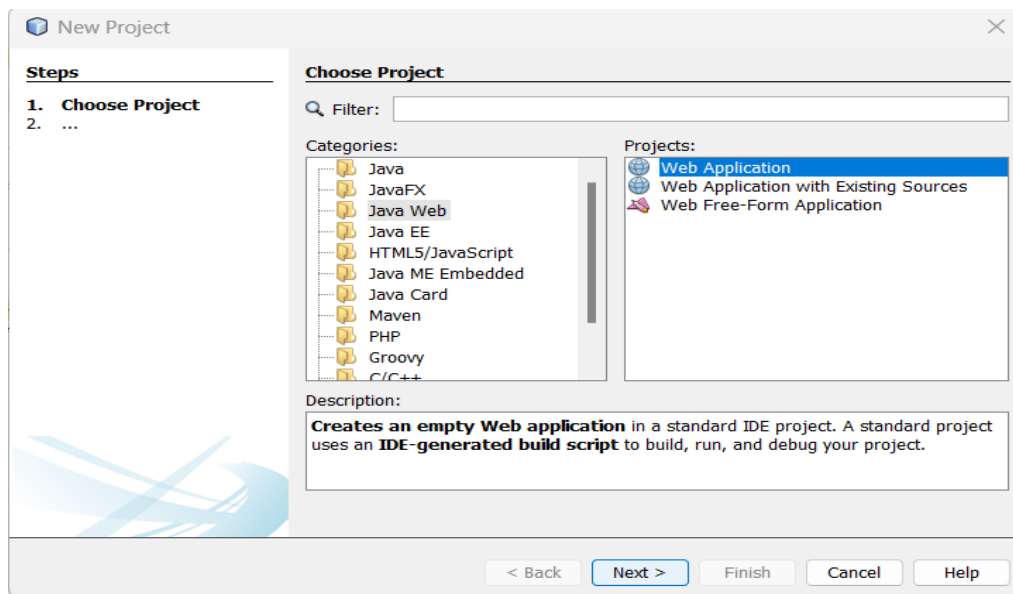


PRACTICAL 4

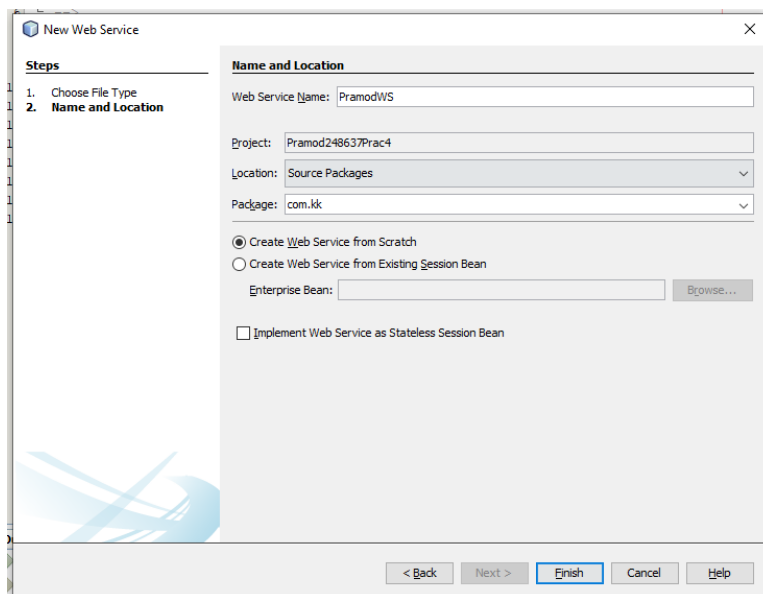
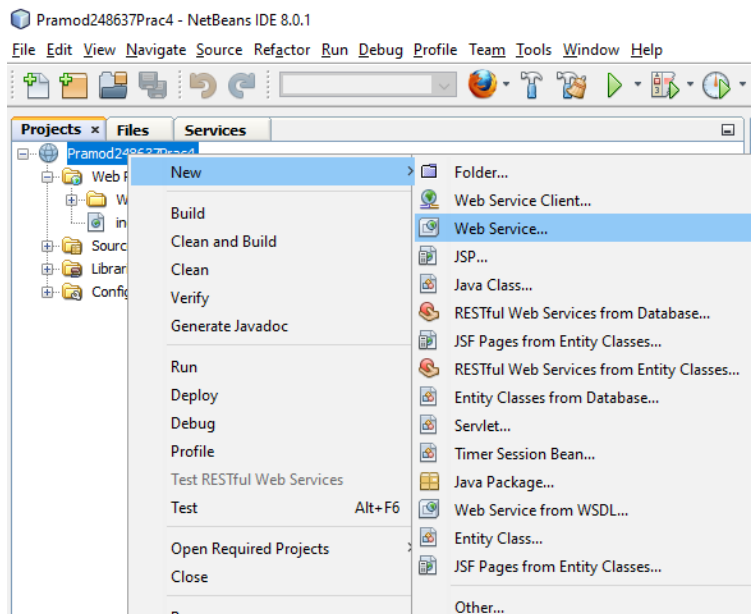
Aim: Show the implementation of web services.

Q1. Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.

Step 1: Create a new project. Click on web application → Name of project → Next → Finish.



Step 2: Once your project is successfully created. Expand your project, right click on project. Select New → Web Service. Enter the name of new Web Service as operation & package name as com.kk. Click on Finish options.



Step 3: Once your web service Created. Remove the selected part from Operation.java

```

/*
 * 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.kk;

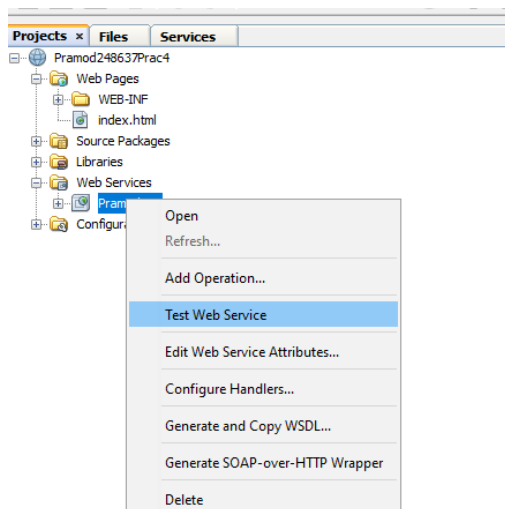
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

/**
 *
 * @author pramo
 */
@WebService(serviceName = "PramodWS")
public class PramodWS {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "Hello " + txt + " !";
    }
}

```

Step 4: Now go to the project palette. Expand your project, Go to the web services → Operation. Right click on Web service, select Test Web Service.



Step 5: Add Operations. The first operation is Addition.

Operation name is Add. The return type will be integer. Add some parameter.

Now here I am taking two parameters.

Parameter name is a & type will be integer.

Parameter name is b & type will be integer. After that, click on OK button.

Add Operation

Name:

Return Type:

Parameters Exceptions

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

Step 6: Now the declaration of variable.

Int c = a+b;

Return c;

```

1  /*
2   * To change this license header, choose License Headers in Project Property
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package com.kk;
7
8   import javax.ws.WebService;
9   import javax.ws.WebMethod;
10  import javax.ws.WebParam;
11
12  /**
13   *
14   * @author pramo
15   */
16  @WebService(serviceName = "PramodWS")
17  public class PramodWS {
18
19      /**
20       * This is a sample web service operation
21       */
22
23      /**
24       * Web service operation
25       */
26      @WebMethod(operationName = "Add")
27      public int Add(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
28          int c = a + b;
29          return c;
30      }
31  }

```

Step 7: Now repeat, the Step 4 & 5 for creation of Subtraction, multiplication & division Operation.

Add Operation

Name:

Return Type:

Parameters Exceptions

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

The 'Add Operation' dialog box shows the 'Name' field set to 'Multi' and the 'Return Type' set to 'int'. The 'Parameters' tab is active, displaying a table with two parameters: 'a' and 'b', both of type 'int'. The 'Final' column has checkboxes, with 'b' selected. The 'Add', 'Remove', 'Up', and 'Down' buttons are visible on the right. The 'OK' and 'Cancel' buttons are at the bottom.

Name	Type	Final
a	int	<input type="checkbox"/>
b	int	<input checked="" type="checkbox"/>

The 'Add Operation' dialog box shows the 'Name' field set to 'Div' and the 'Return Type' set to 'int'. The 'Parameters' tab is active, displaying a table with two parameters: 'a' and 'b', both of type 'int'. The 'Final' column has checkboxes, with 'b' selected. The 'Add', 'Remove', 'Up', and 'Down' buttons are visible on the right. The 'OK' and 'Cancel' buttons are at the bottom.

Name	Type	Final
a	int	<input type="checkbox"/>
b	int	<input checked="" type="checkbox"/>

Step 8: Now, write the following code for subtraction, multiplication & division method.

Subtraction method:

```
int c = a-b;
```

```
return c;
```

Multiplication method:

```
int c = a*b;
```

```
return c;
```

Division method:

```
int c = a/b;
```

```
return c;
```

```

@WebMethod(operationName = "Sub")
public int Sub(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    //TODO write your implementation code here:
    return a-b;
}

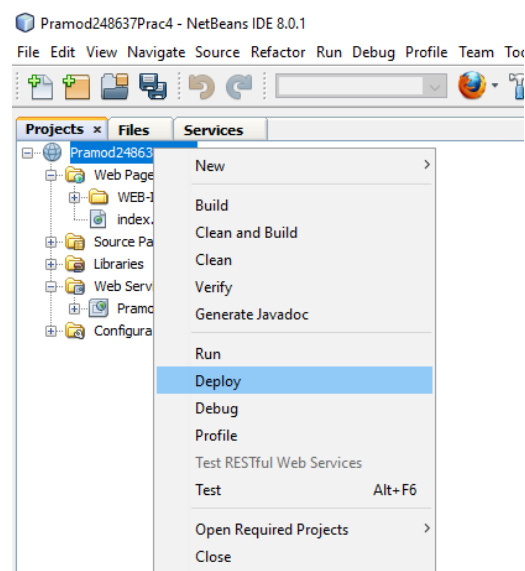
@WebMethod(operationName = "Multi")
public int Multi(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    //TODO write your implementation code here:
    return a*b;
}

@WebMethod(operationName = "Div")
public int Div(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    //TODO write your implementation code here:
    return a+b;
}
/**
 * This is a sample web service operation
 */
}

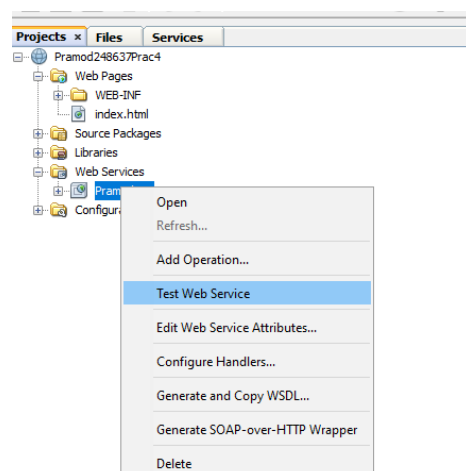
```

Step 9: Deploy Web Service

Right click on Web Application and deploy



Step 10: Now test your web service.



PramodWS Web Service Tester

localhost:8080/Pramod248637Prac4/PramodWS?T

PramodWS 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 label

Methods :

public abstract int com.kk.PramodWS.add(int,int)

add (12 , 32)

public abstract int com.kk.PramodWS.mul(int,int)

mul (42 , 21)

public abstract int com.kk.PramodWS.div(int,int)

div (43 , 23)

public abstract int com.kk.PramodWS.sub(int,int)

sub (312 , 43)

add Method invocation

mul Method invocation

div Method invocation

sub Method invocation

Method parameter(s)

Method parameter(s)

Method parameter(s)

Method parameter(s)

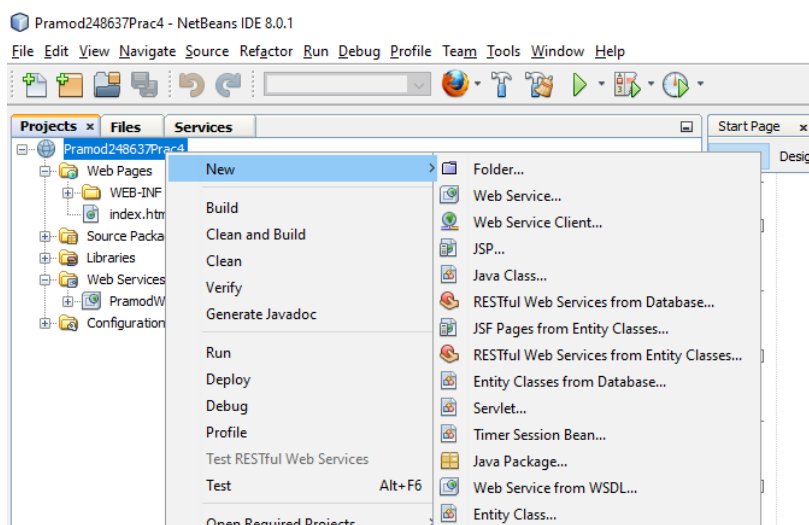
Method returned

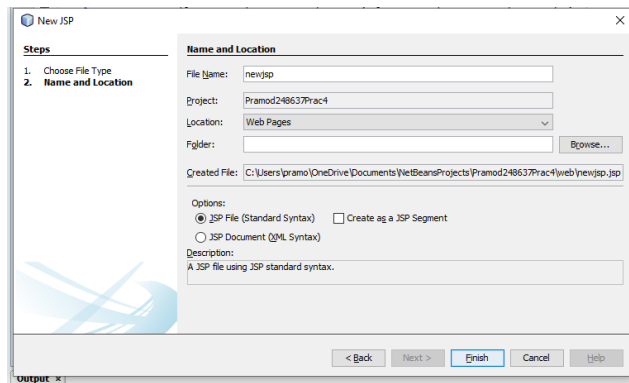
Method returned

Method returned

Method returned

Step 11: Now right click on project. Select New → JSP page. Give the file name & click on Finish.





Step 11: Write the following code in JSP page.

```
<%
```

```
com.kk.PramodWS a1 = new com.kk.PramodWS();

int d = a1.Add(3, 6);

int e = a1.Sub(45, 26);

int f = a1.Mul(23, 3);

int g = a1.Div(64, 6);

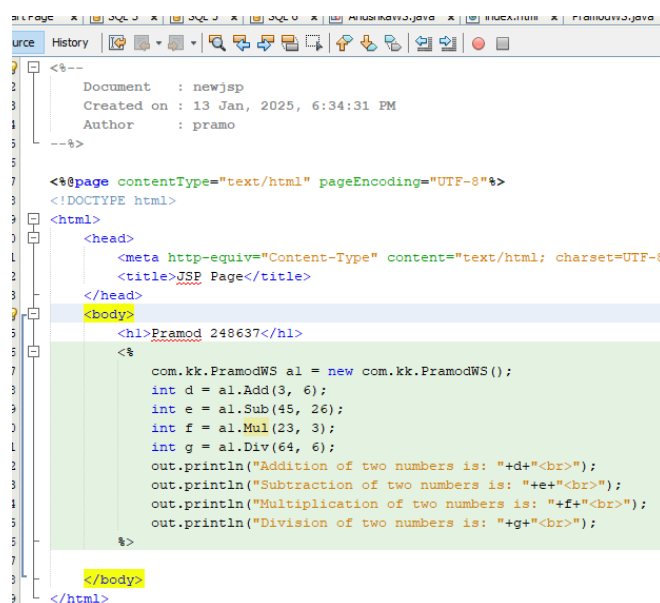
out.println("Addition of two numbers is: "+d+"<br>");

out.println("Subtraction of two numbers is: "+e+"<br>");

out.println("Multiplication of two numbers is: "+f+"<br>");

out.println("Division of two numbers is: "+g+"<br>");
```

```
%>
```



Step 12: Now right click on JSP page page. Click on Run file.

Pramod 248637

Addition of two numbers is: 9

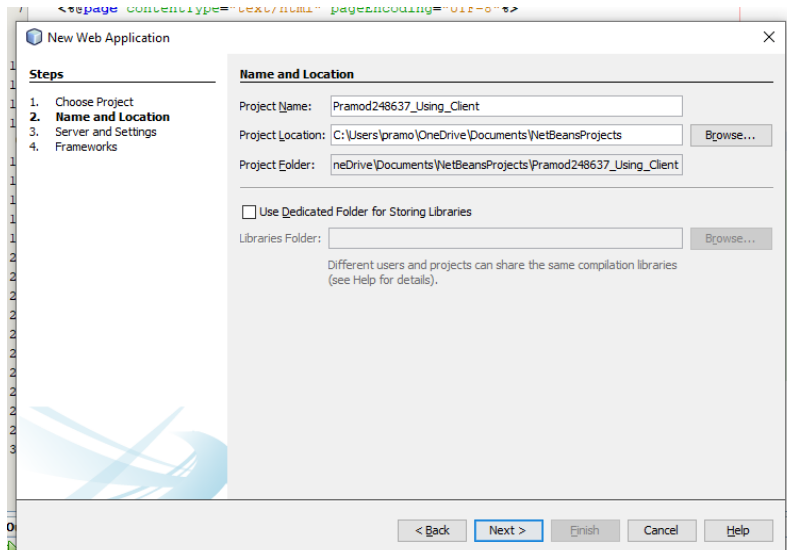
Subtraction of two numbers is: 19

Multiplication of two numbers is: 69

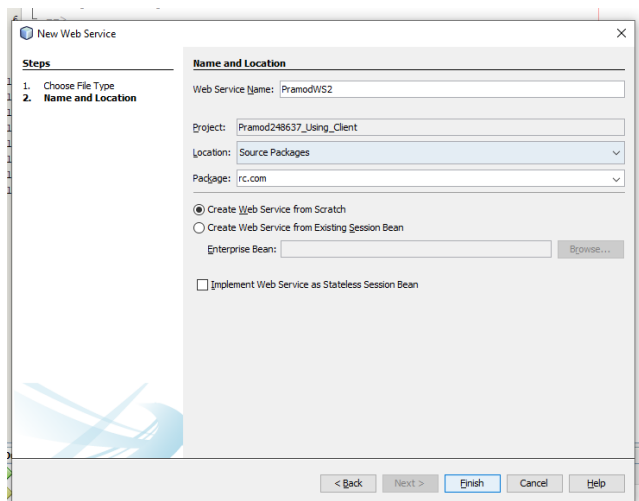
Division of two numbers is: 10

Q2. Write a JAX-WS web service server & client to perform the following operations. Define a servlet/ JSP that consumes the web service.

Step 1: Create a new project. New project → Java Web → Web Application.



Step 2: Right click on newly created project. Create a new web service. The service name is PramodWS2 & package name is rc.com.



Step 3: Remove the selected content from PramodWS2.java

```

    /** This is a sample web service operation
    */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "Hello " + txt + " !";
    }
}

```

Step 4: Write the following code in deleted content.

@WebMethod (operationName = "Square")

public double square (double val) {

return val*val;

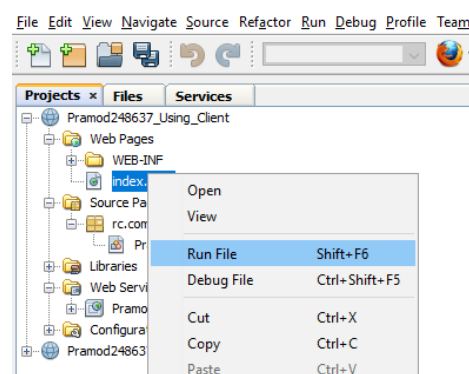
}

```

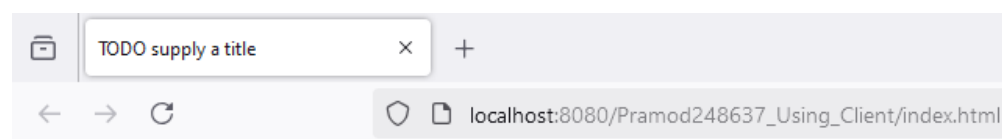
20
21
22 /**
23  * This is a sample web service operation
24  */
25 @WebMethod(operationName = "Square")
26 public double square(double val) {
27     return val*val;
28 }
29

```

Step 5: Now go to the palate. Right click on index.html in project. Click on run file.

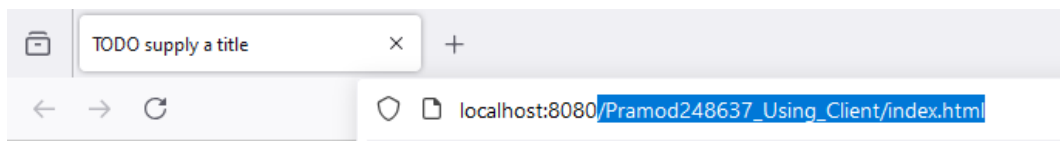


The following window will open after run your index.html file.

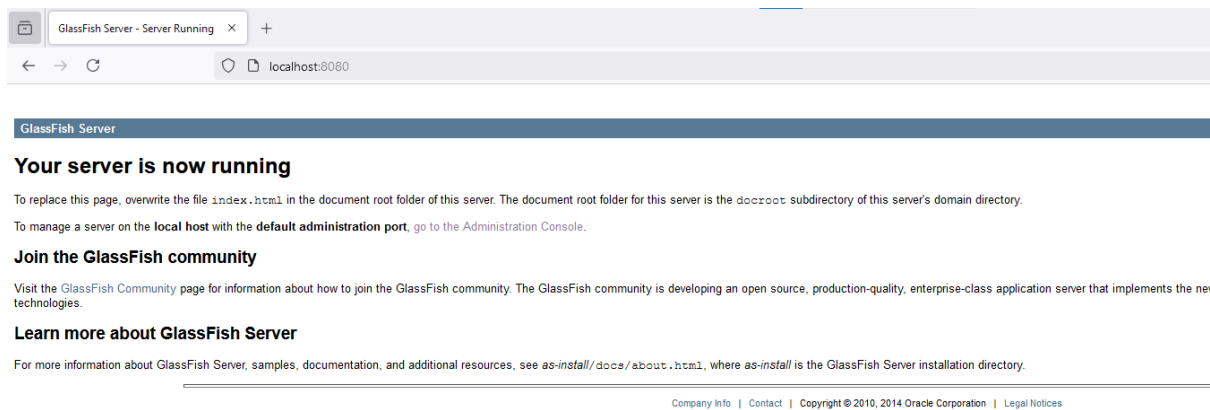


TODO write content

Step 6: Remove the selected part from link.



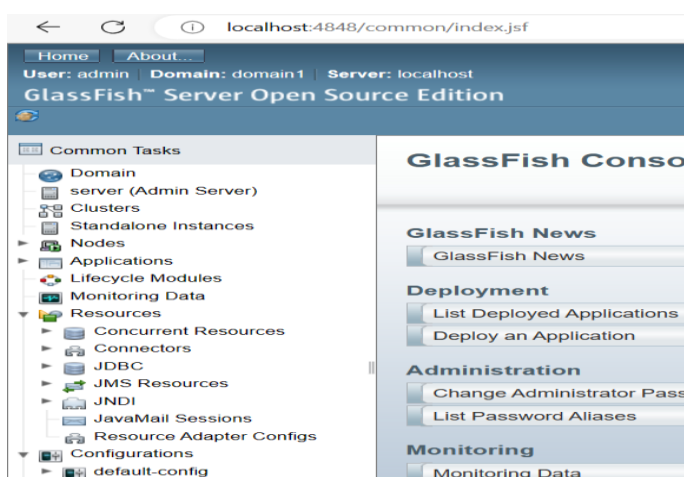
Now, the below window will open.



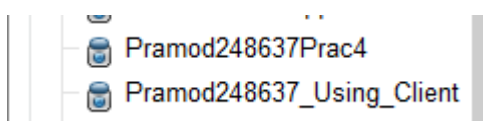
Step 7: Click on go to the administration console.

port, go to the Administration Console.

The following window will open.



Step 8: In left side panel, expand the Application task. Click on your project.



Step 9: Go to the general. Scroll down the page, there is one table which name is Modules & Components. In action column, click on View Endpoint.

Action
Launch
View Endpoint

Step 10: Click on [/Pramod248637_Using_Client/PramodWS2?wsdl](#) link.

Web Service Endpoint Information

View details about a web service endpoint.

Application Name:	Pramod248637_Using_Client
Tester:	/Pramod248637_Using_Client/PramodWS2?Tester
WSDL:	/Pramod248637_Using_Client/PramodWS2?wsdl
Endpoint Name:	PramodWS2
Service Name:	PramodWS2
Port Name:	PramodWS2Port
Deployment Type:	109
Implementation Type:	SERVLET
Implementation Class Name:	rc.com.PramodWS2
Endpoint Address URI:	/Pramod248637_Using_Client/PramodWS2
Namespace:	http://com.rc/

Step 11: When you click on link. The View WSDL window will open

Web Service Endpoint Informati × View WSDL × +

← → ↺ localhost:4848/common/applications/webServiceWsdlsjsf?ap

View WSDL

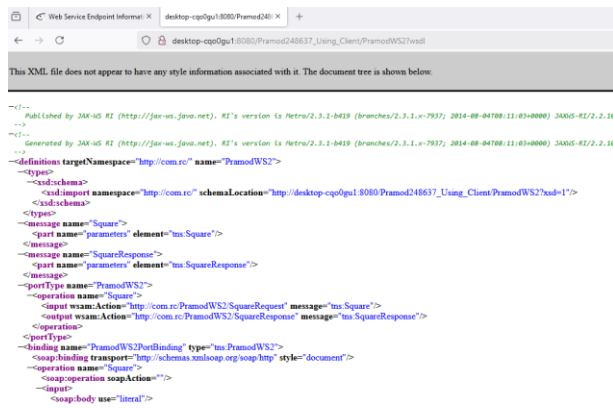
If the server or listener is not running, the link may not work. In this case, check the status of the server instance. After launching the

Application Name: Pramod248637_Using_Client

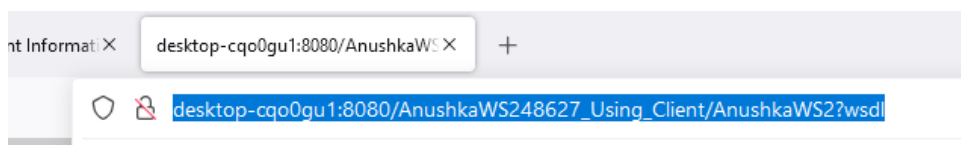
Links:

- [server] http://DESKTOP-CQ00GU1:8080/Pramod248637_Using_Client/PramodWS2?wsdl
- [server] https://DESKTOP-CQ00GU1:8181/Pramod248637_Using_Client/PramodWS2?wsdl

Step 12: Click on first link which will provide the access of server.

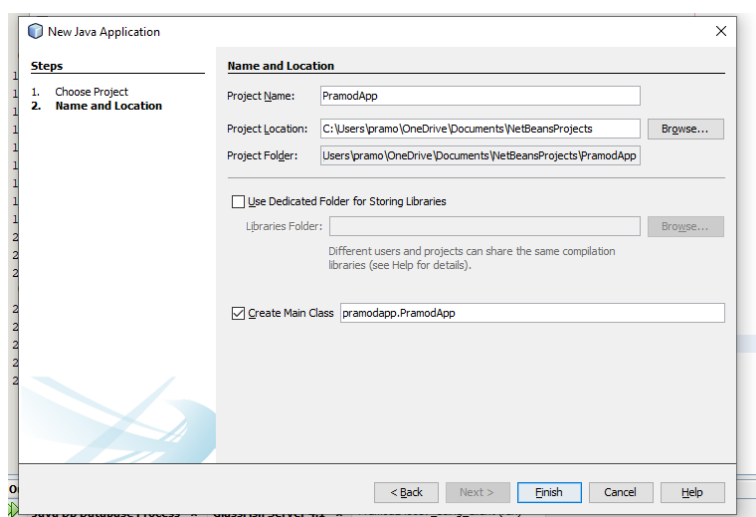
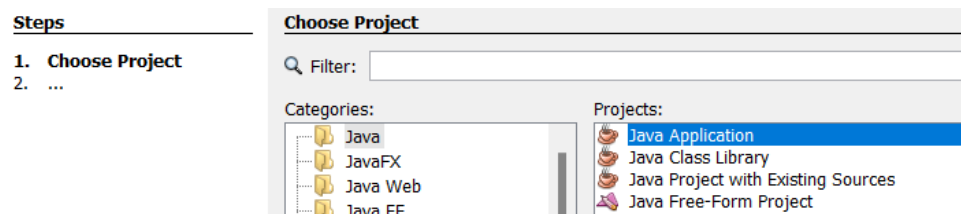


Step 13: Copy the link from search engine to notepad.



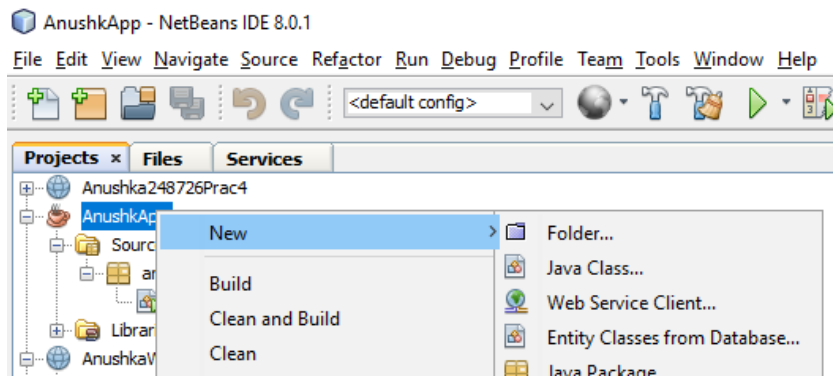
Now minimize the Microsoft edge window.

Step 13: Create a new Java project.

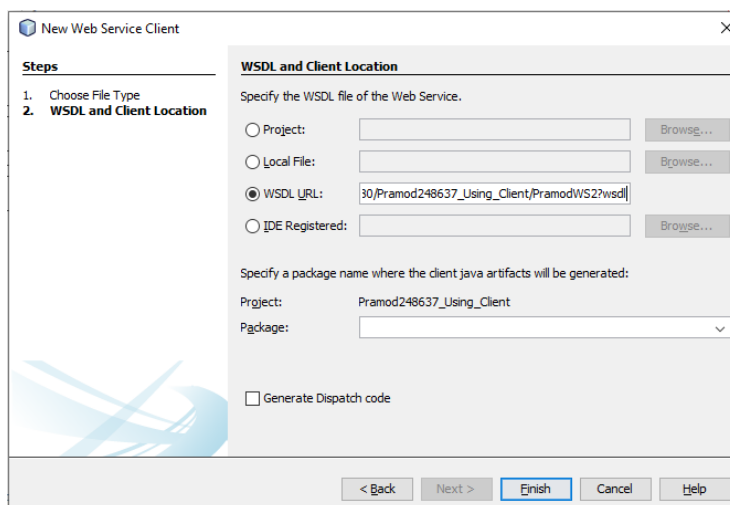


Step 14: In Java Application project, create a new web service client.

Right click on project → New → Web Service Client



Step 15: In new web service client. Select wsdl URL. Paste the copied link.



Step 16: Open JavaApplication.java. Type the following code.

```
PramodWS2_Service service = new PramodWS2_Service();
```

```
final PramodWS2 proxy = service.getPramodWS2Port();
```

```
System.out.println(proxy.square(7));
```

```

/* @author pramo
 */
public class PramodApp {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        PramodWS2_Service service = new PramodWS2_Service();
        final PramodWS2 proxy = service.getPramodWS2Port();
        System.out.println(proxy.square(7));
    }

}

```

Step 17: Now import the service packages.

```
L  */  
package pramodapp;  
import rc.com.PramodWS2;  
import rc.com.PramodWS2_Service;
```

Step 18: Run the Java application project.

