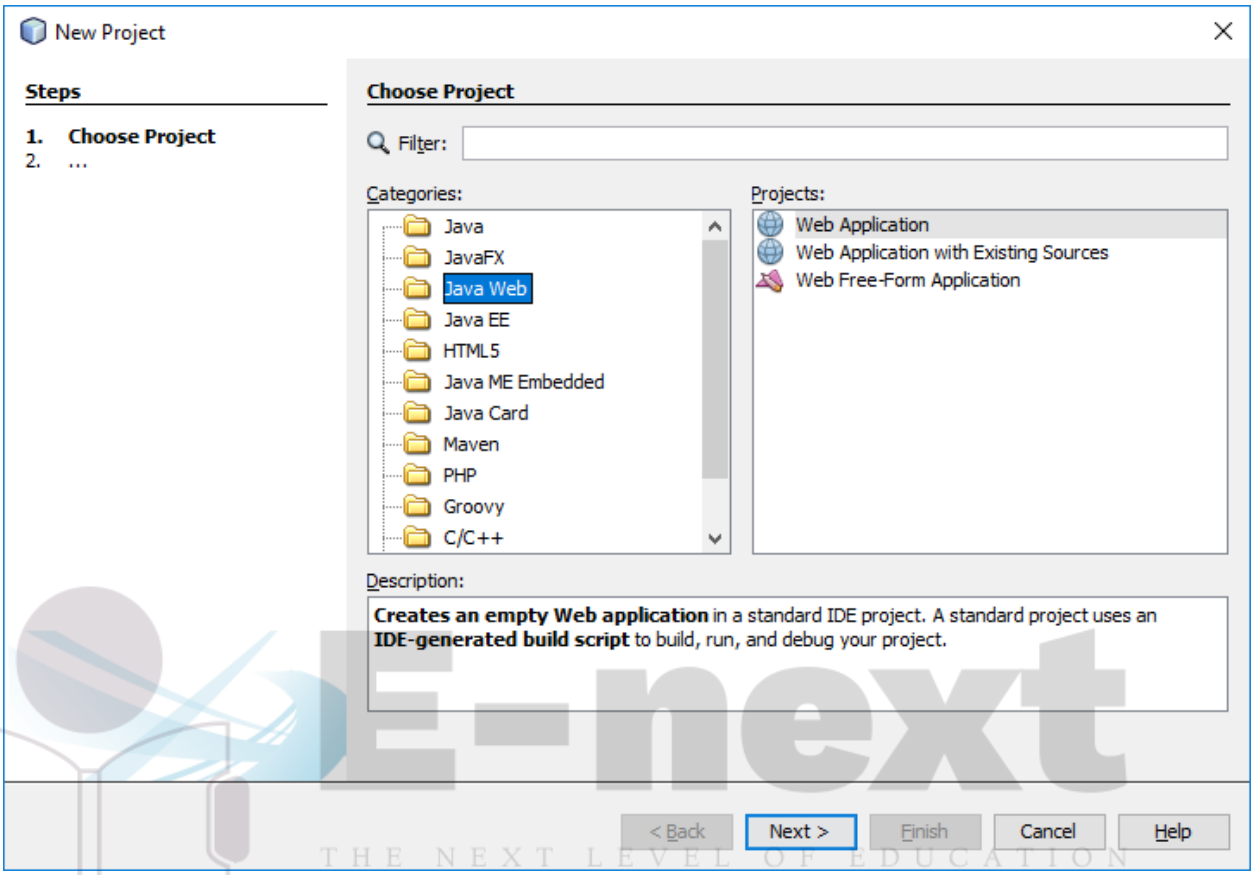
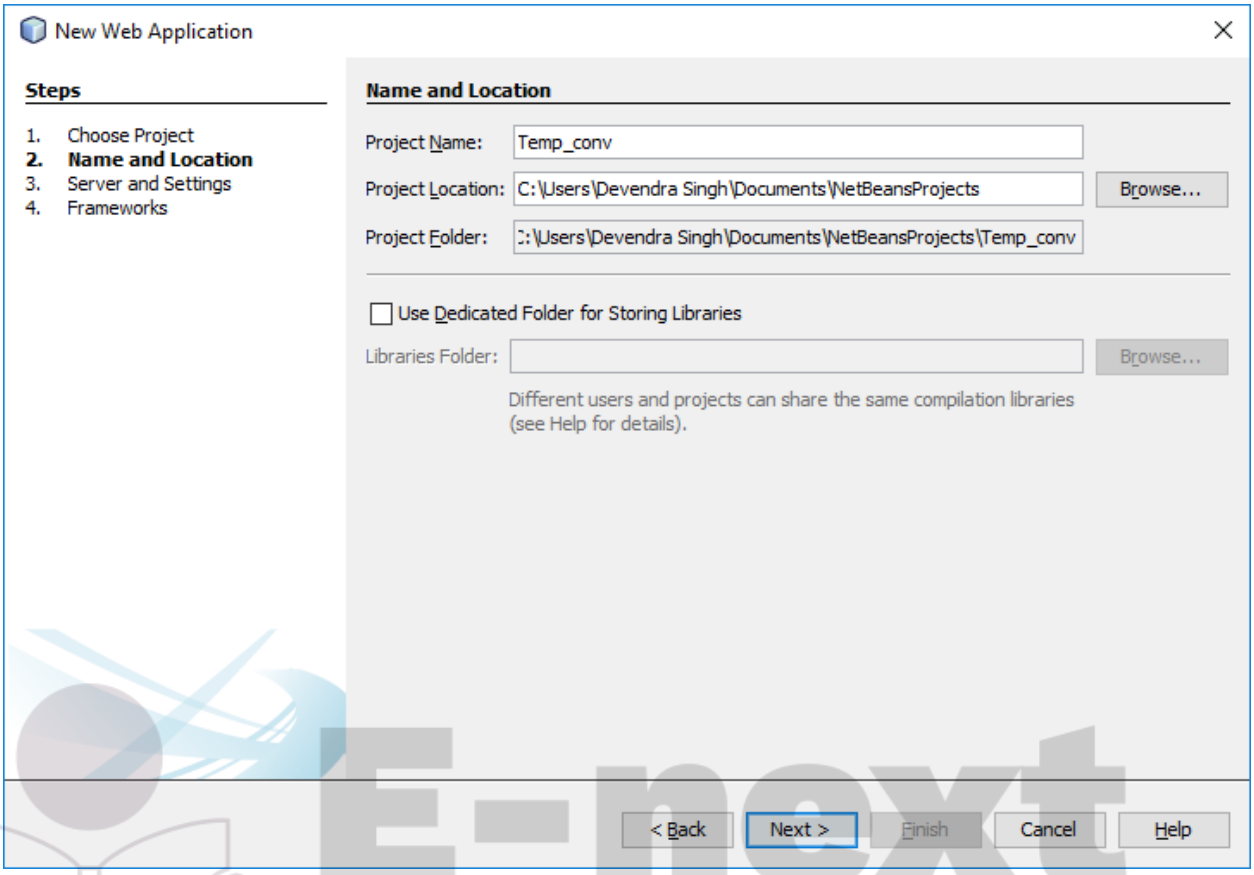
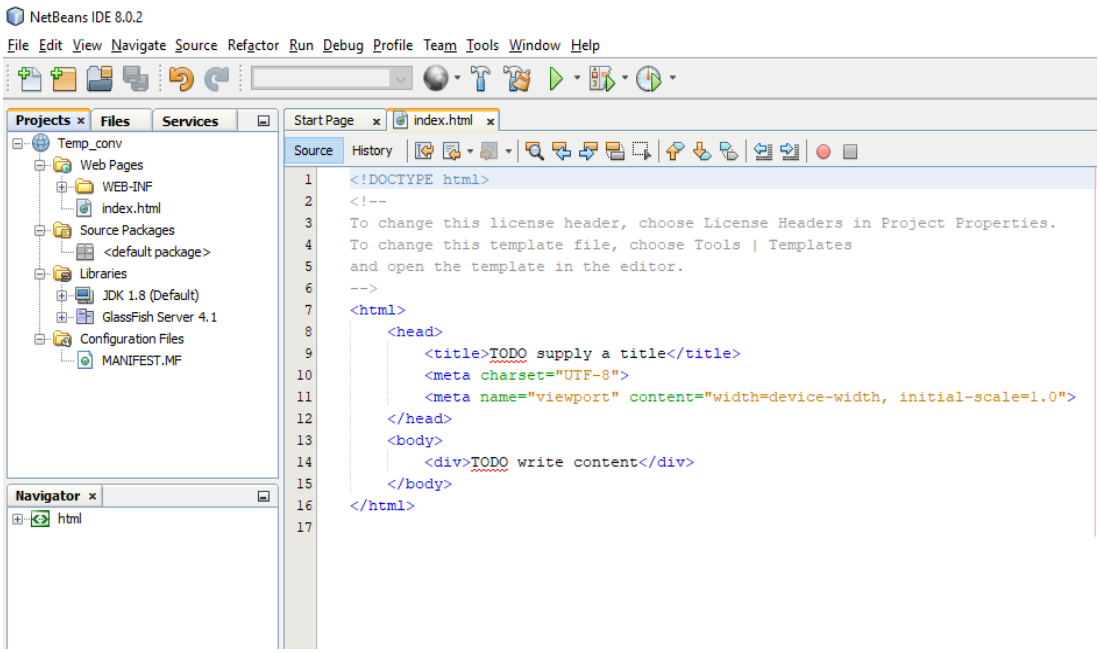
1. Go to **File -> New Project**. Select **Java Web in categories** and **Web Application in Projects. Click on Next** to create web based project.



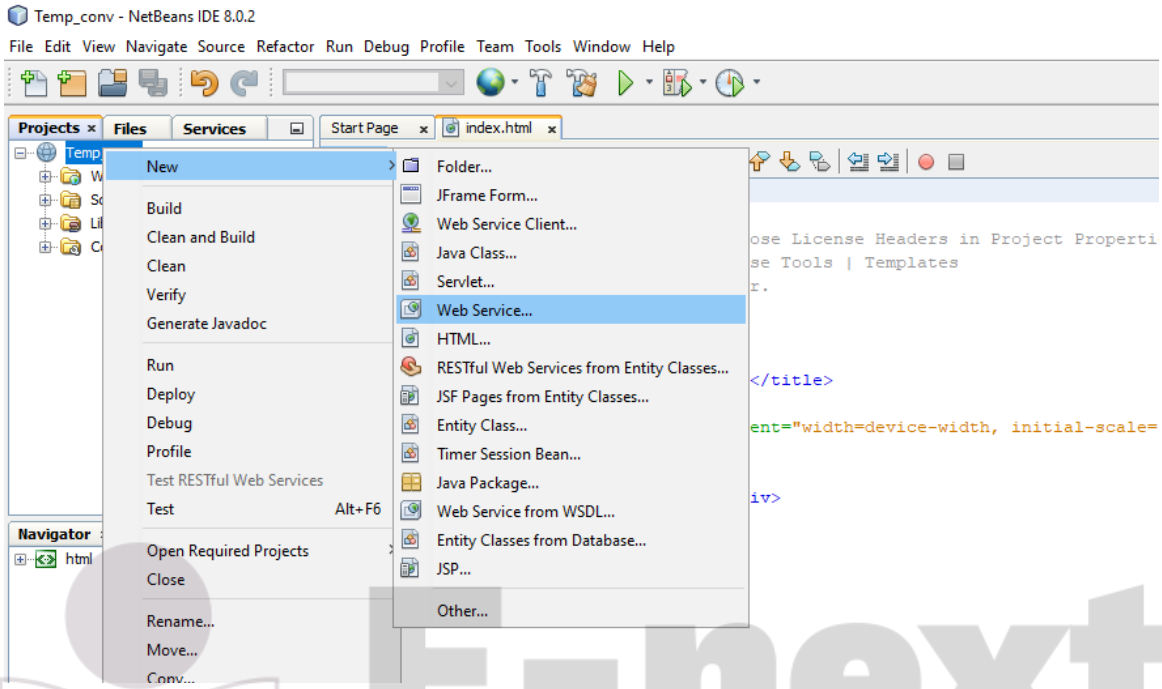
2. Enter a project name whatever you want and then click on **Next**. On next page click **Finish**.



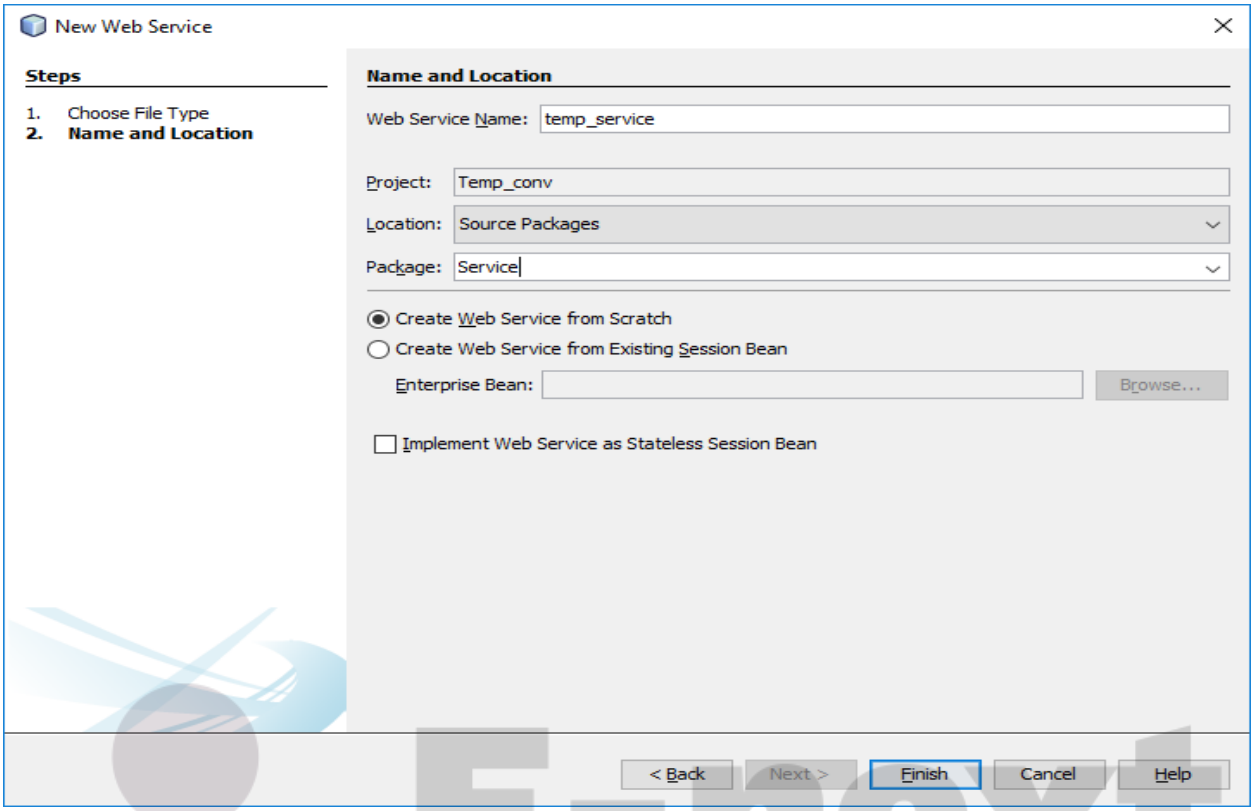
3. After completion of project creation process a window will appear like below.



4. Create web service.
   Right click on Project -> New -> Web Service

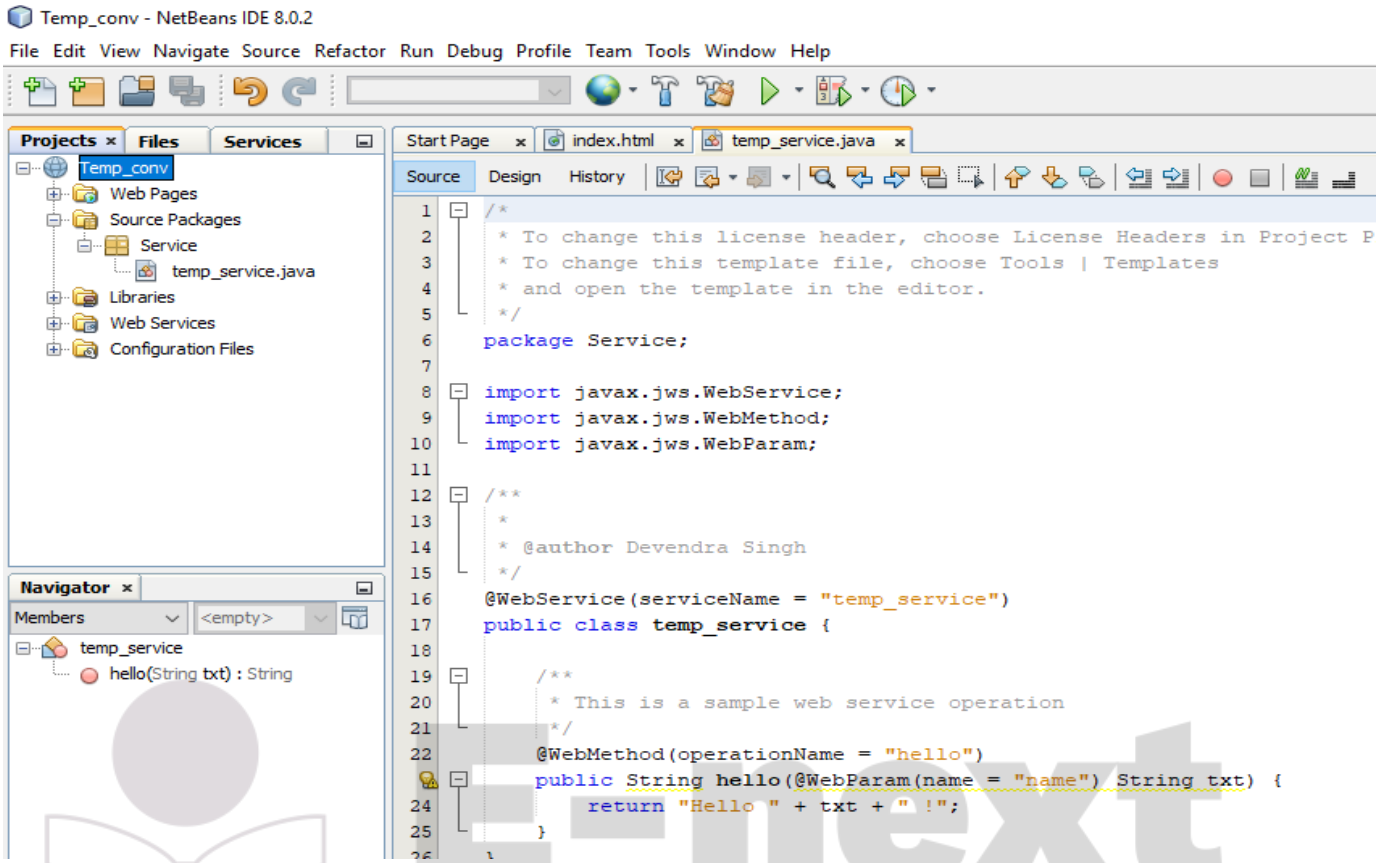


5. **Enter a Web Service Name and package name** and then **click on Finish** to create a Web Service.

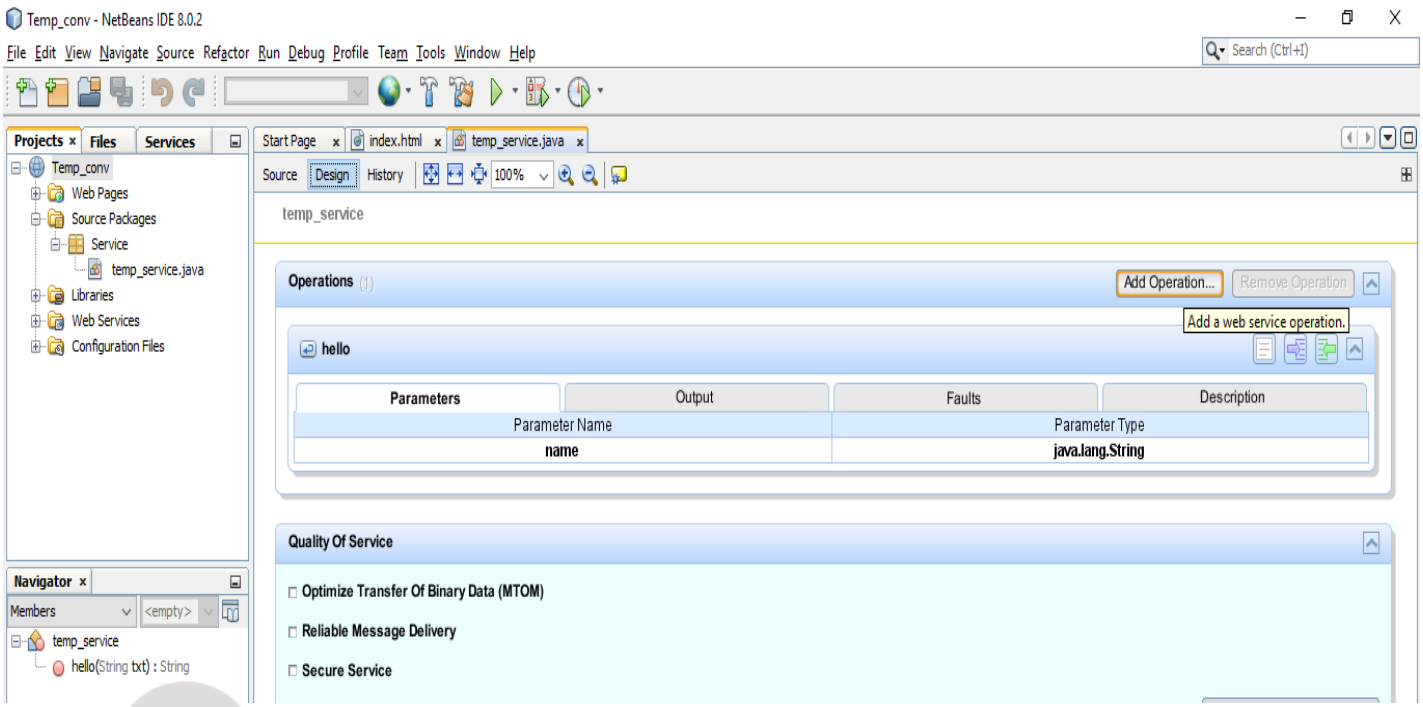


6. As you can see in following pic; In **Source Packages** there is a package **Service** which contains the service file **temp_service.java**. Open this file by double click on it, So that we can add two operation that will convert temperature from celcious to farhenheit and vice-versa.

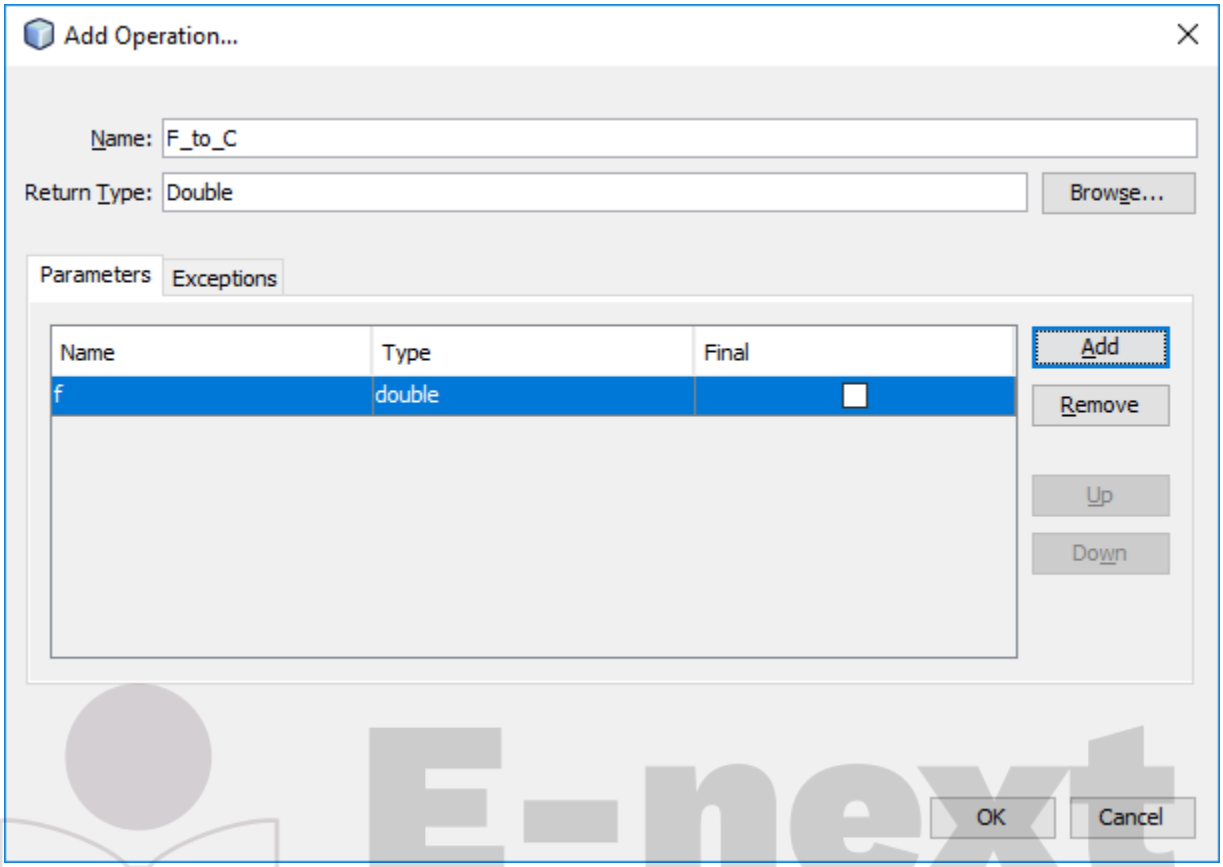
Go to design mode by click on **Design** .



7. **Click on Add Operation** to add operation.



8. Give Operation name **F_to_C** and return type as **Double**. So this method will return value in Double data type. After that click on **Add** button to give parameters for method. Give its name as **f** and type as **Double** and then click on **OK** button. Your one operation is now successfully created.

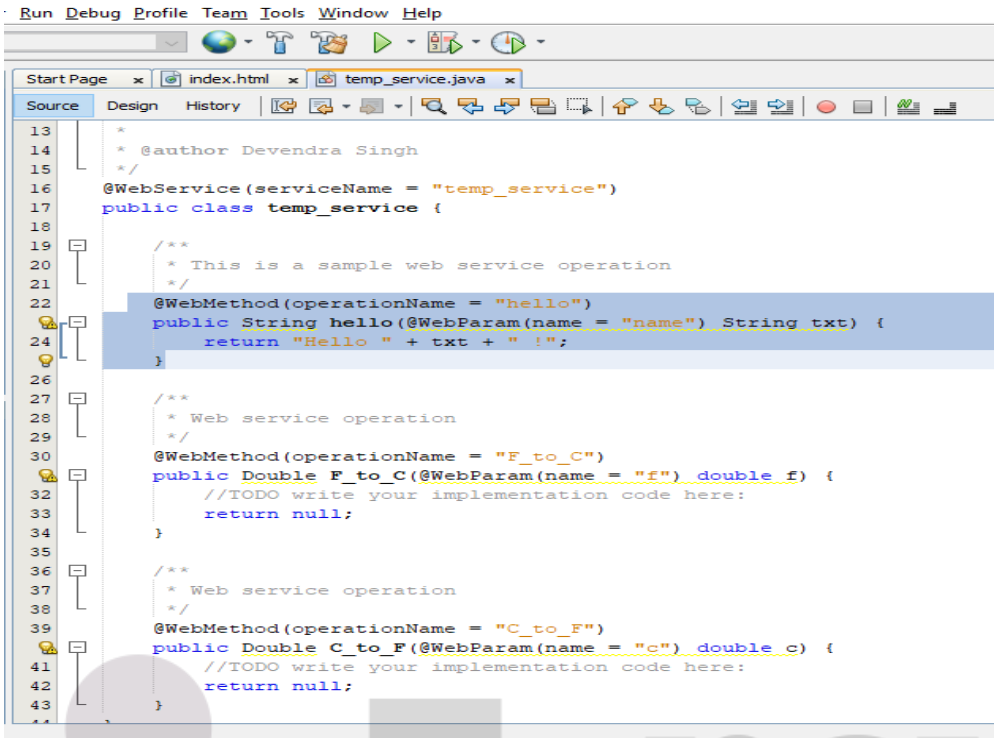


9. **Repeat step 7 & 8** to create second operation. But this time replace some above entered data with following data.

   F_to_C -> C_to_F

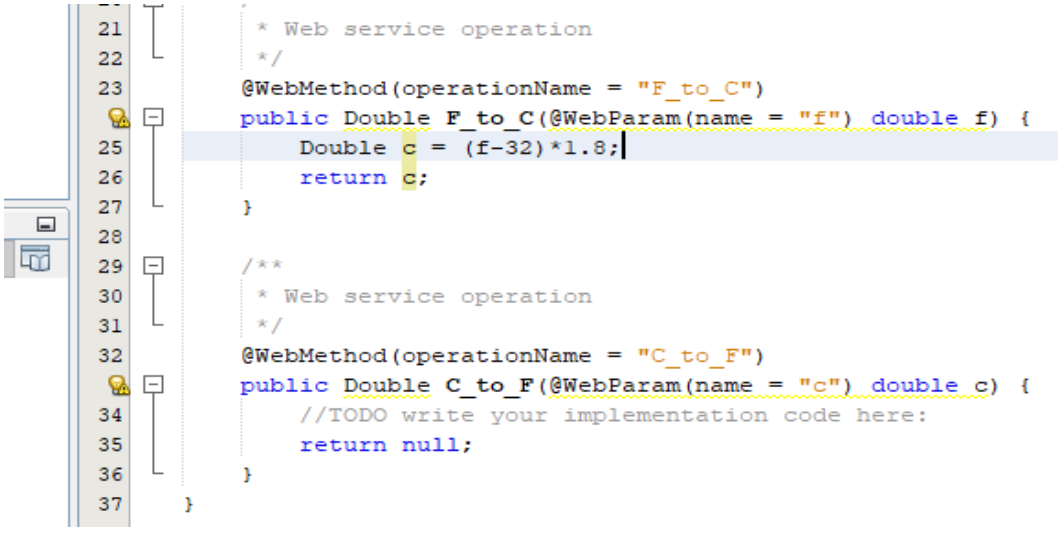
   f -> c

10. Now go to source mode by click on **Source** and **delete the selected** segment of code. Because it is default operation and we don't need this.



11. Now **replace the selected area with following code** to convert Fahrenheit to Celsius.


```
Double c = (f-32)*1.8;
 return c;
```

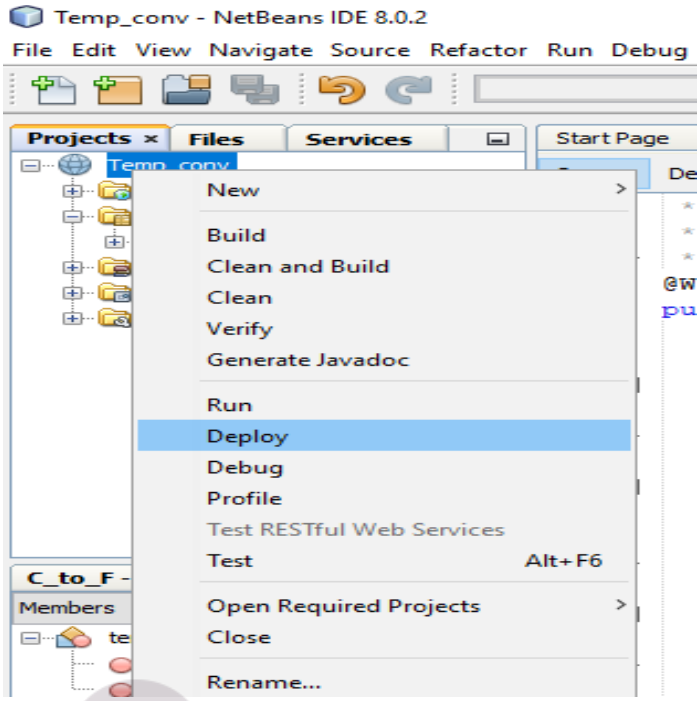


12. Now **replace the selected area with following code** to convert Celsius to Fahrenheit and then press Ctrl+S to save.

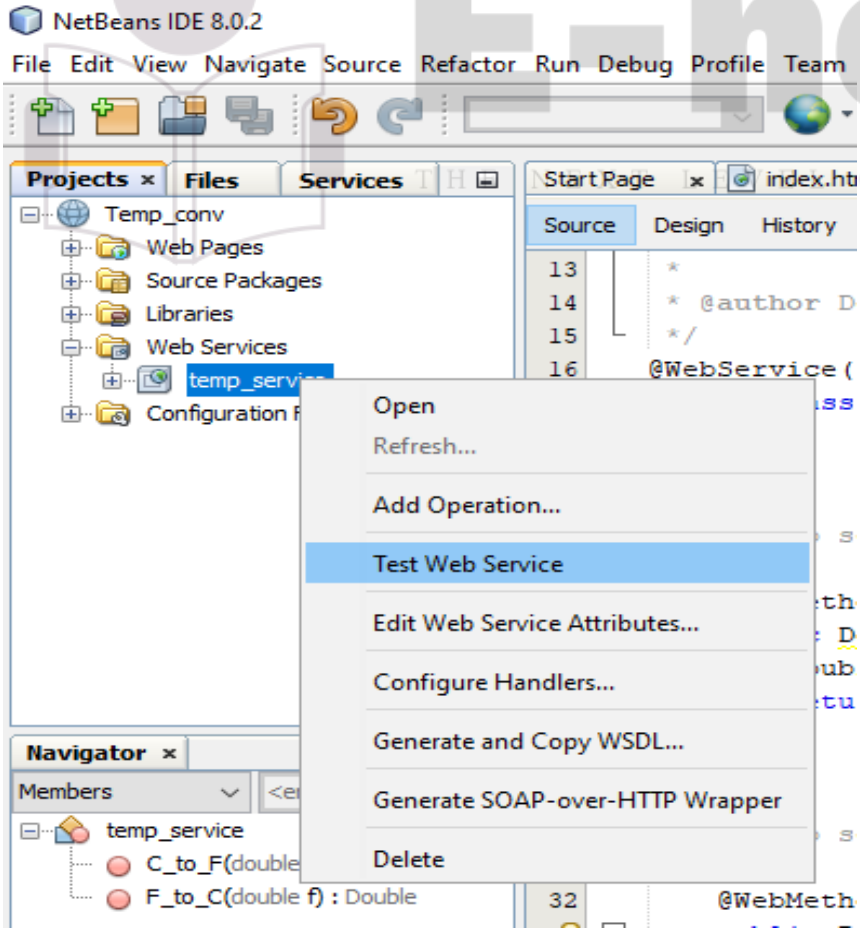
```
Double f = (c*1.8)+32;
return f;
```

```
18
19
20      /**
21       * Web service operation
22       */
23      @WebMethod(operationName = "F_to_C")
24      public Double F_to_C(@WebParam(name = "f") double f) {
25          Double c = (f-32)*1.8;
26          return c;
27      }
28
29      /**
30       * Web service operation
31       */
32      @WebMethod(operationName = "C_to_F")
33      public Double C_to_F(@WebParam(name = "c") double c) {
34          //TODO write your implementation code here:
35          return null;
36      }
37  }
38
```

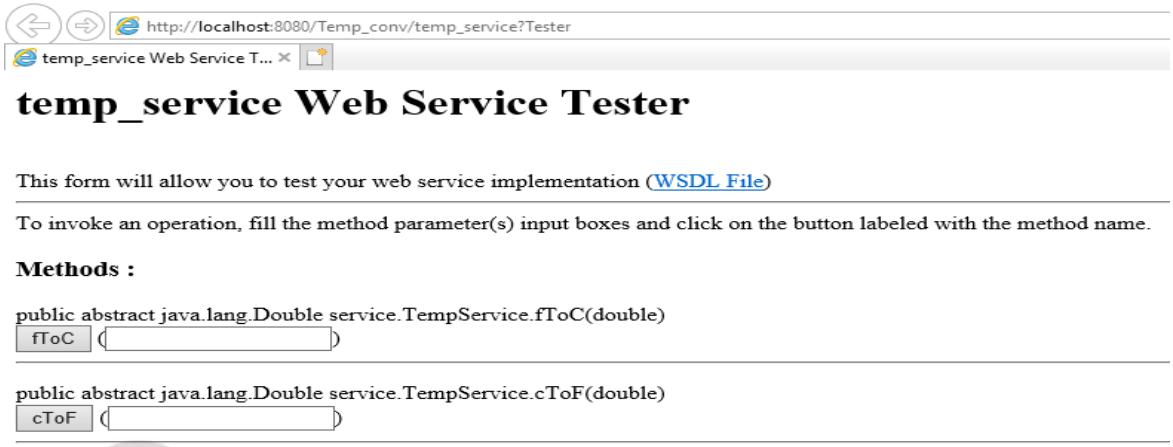
13. Now **right click on project name** and **click on Deploy** to deploy your project.



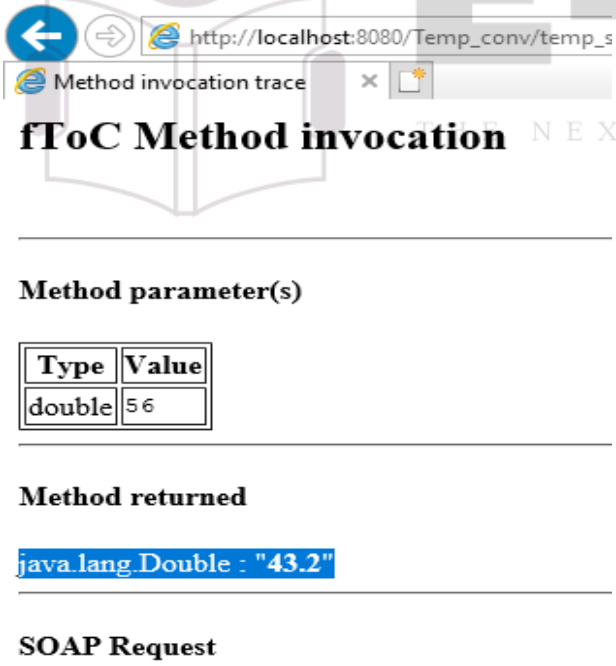
14. To test your web service follow the following process as in picture.



15. Following window will open in in browser. Now if you will enter a numeric data into first box and you will click on first button it will convert the entered data into Celsius.

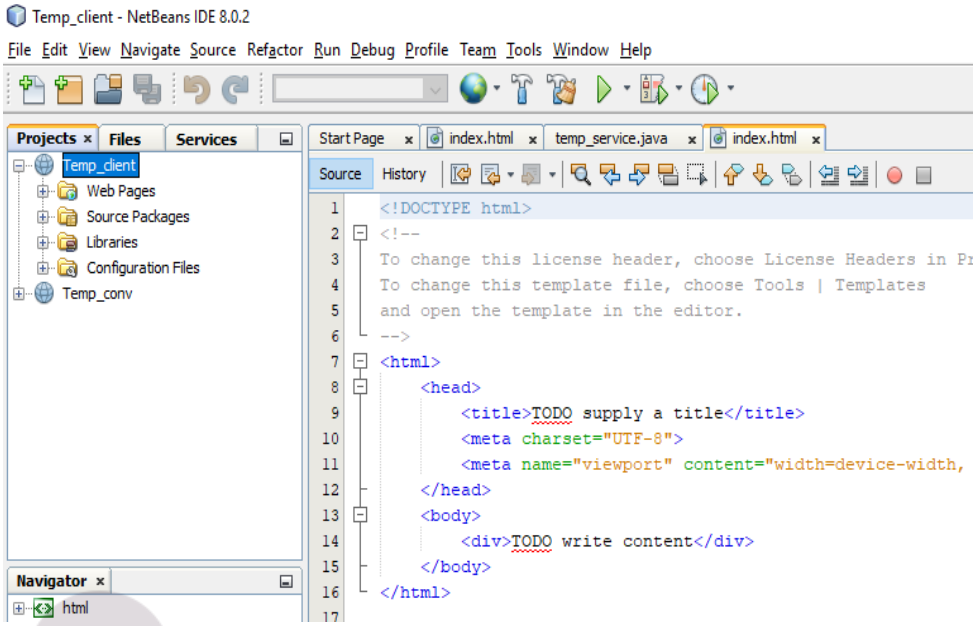


16. Selected value is in celsius of 56.



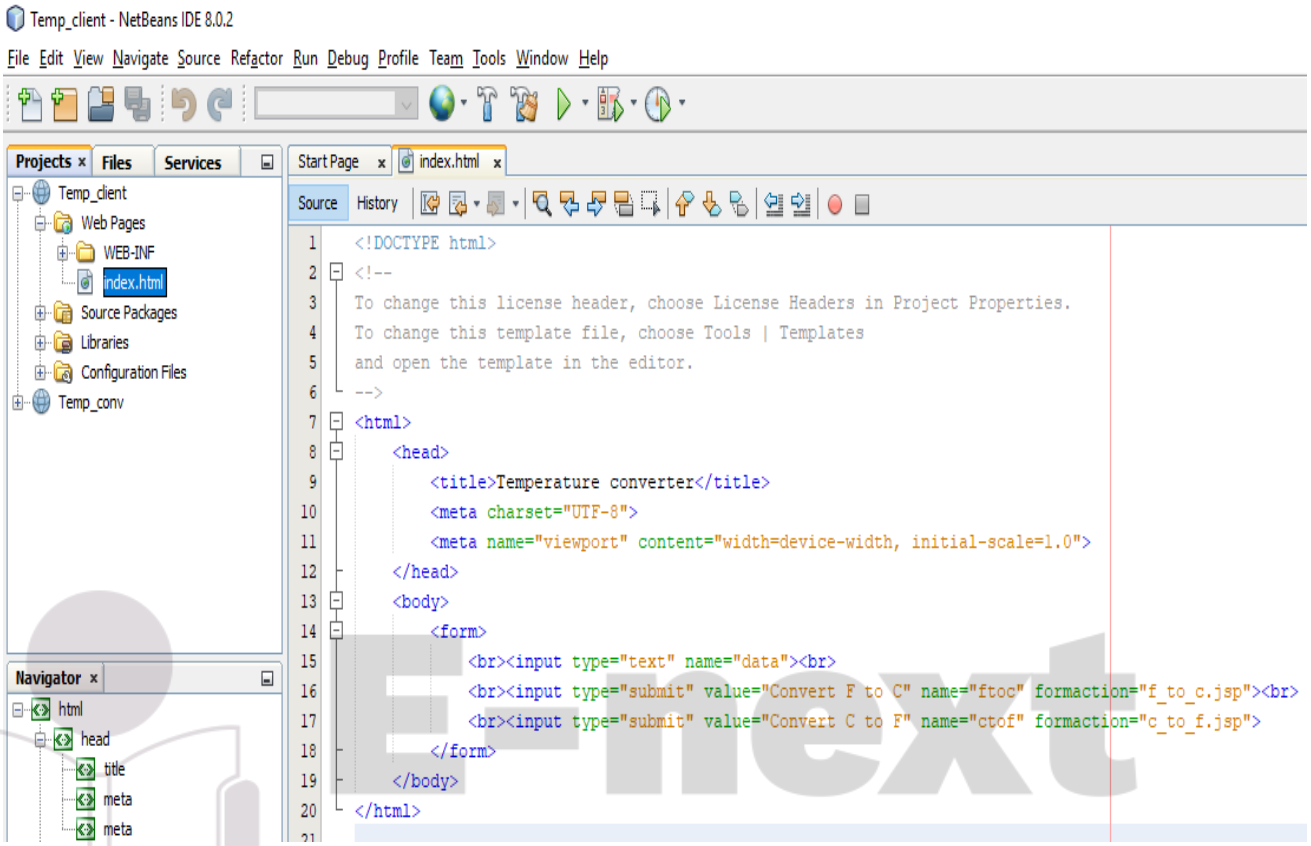
17. Similarly second textbox and button will convert the numeric value into Fahrenheit. Now to consume this web service we are creating a client.

18. Now create a new web application project with **name** as **Temp_client**.



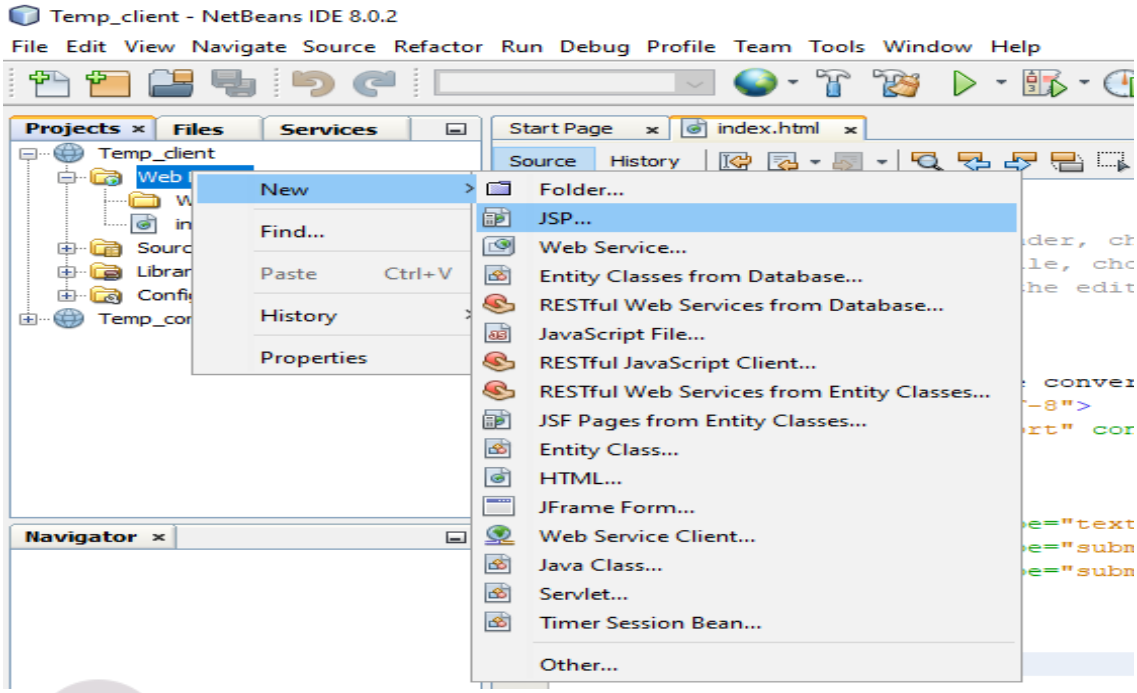
19. Now **open the index.html** page of Temp_client and write the following code into that.

```
<html>
  <head>
    <title>Temperature converter</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  </head>
  <body>
    <form>
      <br><input type="text" name="data"><br>
      <br><input type="submit" value="Convert F to C" name="ftoc"
formaction="f_to_c.jsp"><br>
      <br><input type="submit" value="Convert C to F" name="ctof"
formaction="c_to_f.jsp">
    </form>
  </body>
</html>
```

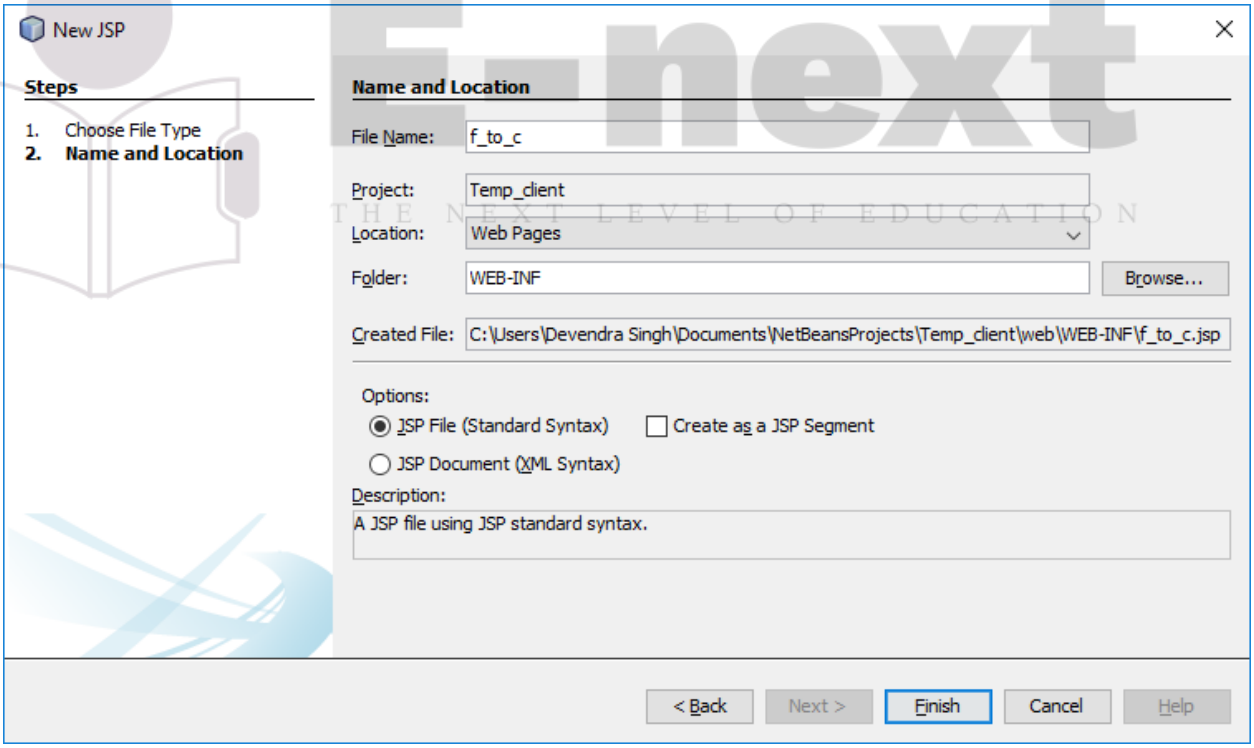


20. Now create two jsp pages for both submit button.

Right click on Web Pages -> New -> JSP

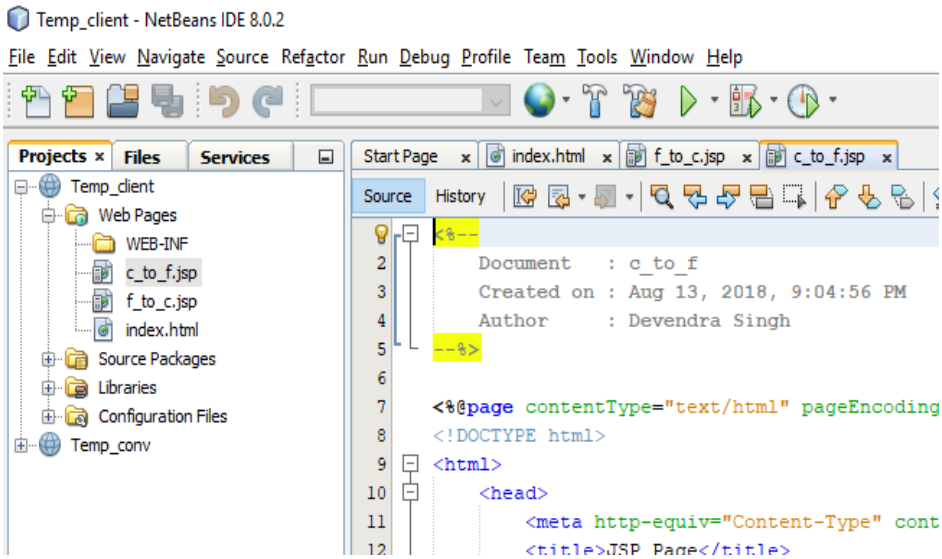


21. Enter file name f_to_c and then **click on Finish**.

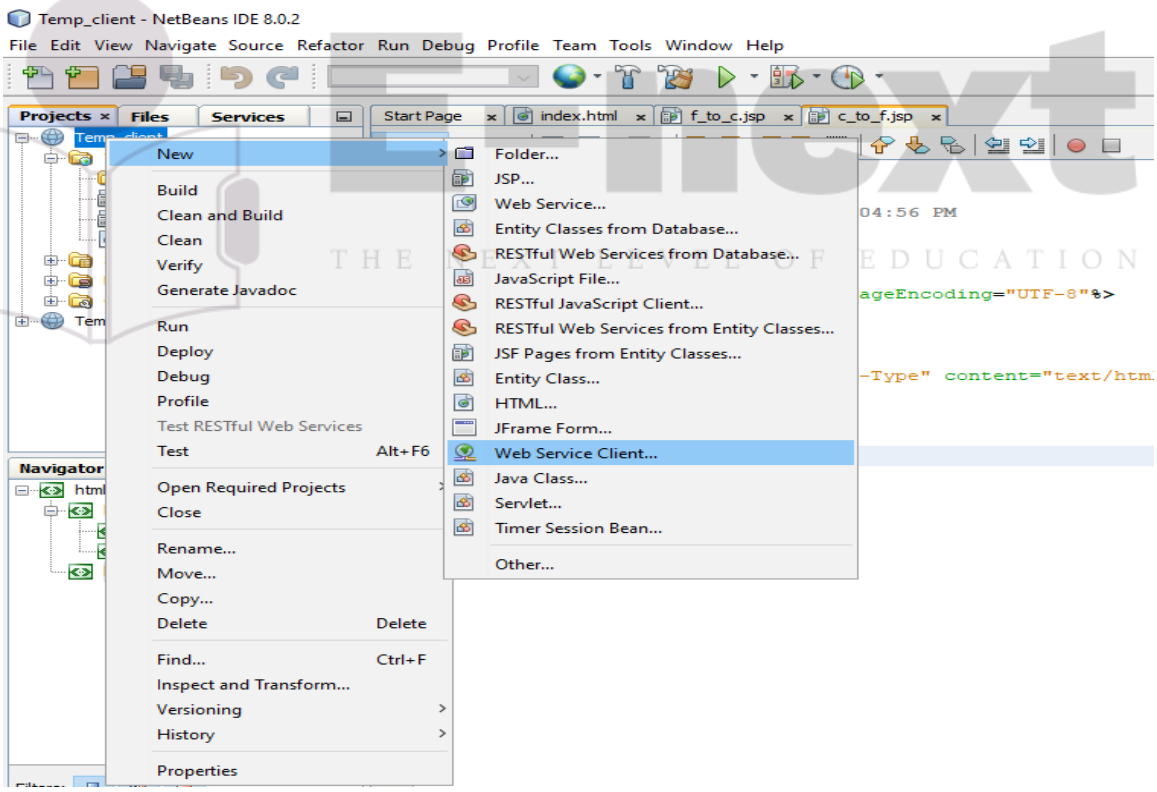


22. Now **repeat the step number 20 & 21.** But **give the File Name as c_to_f**.

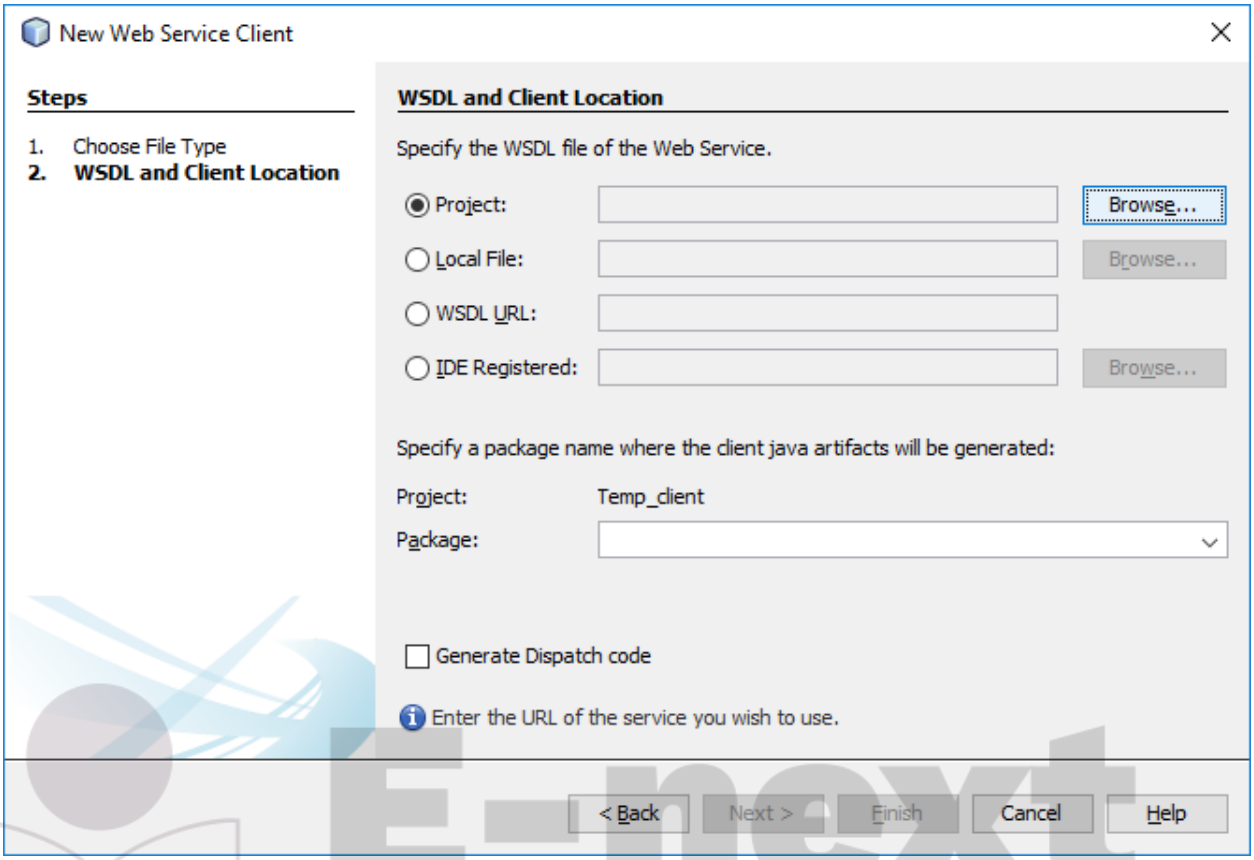
23. Now you have created two jsp files.



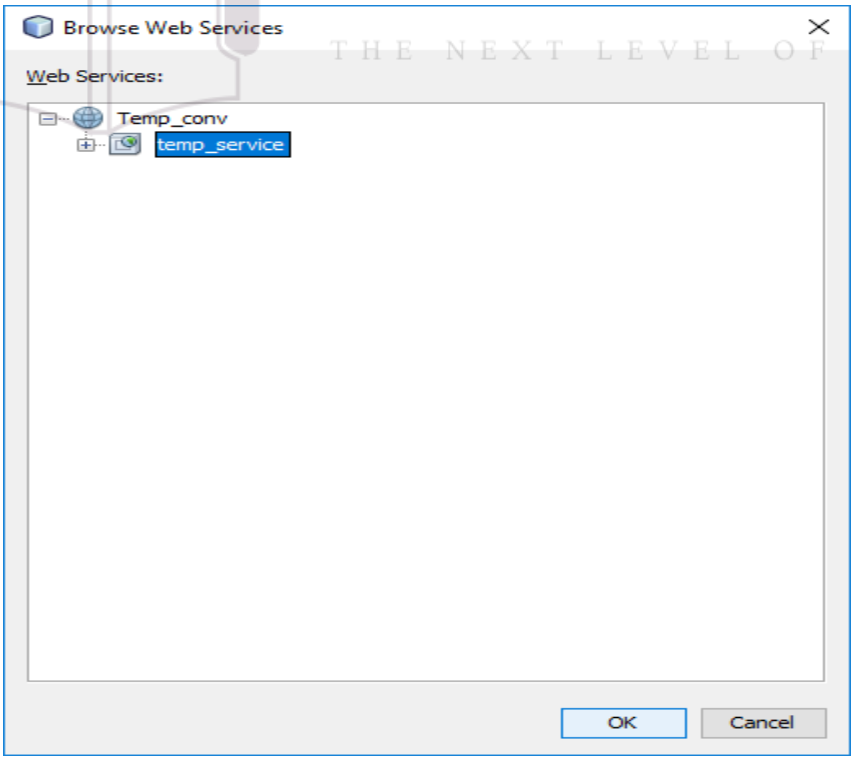
24. Right click on Temp_client and **select Web Service Client** as below.



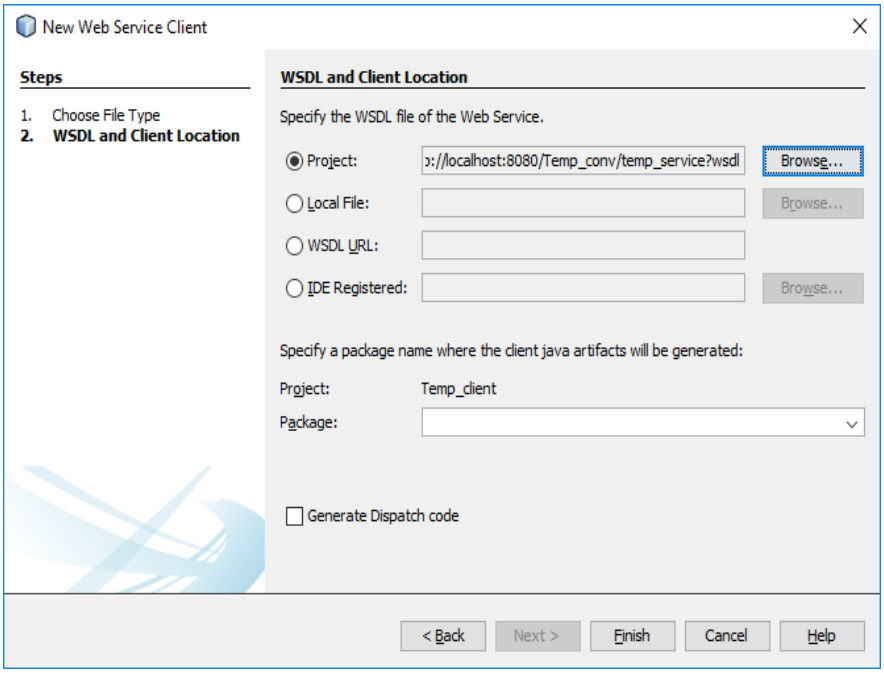
25. Click on Browse.



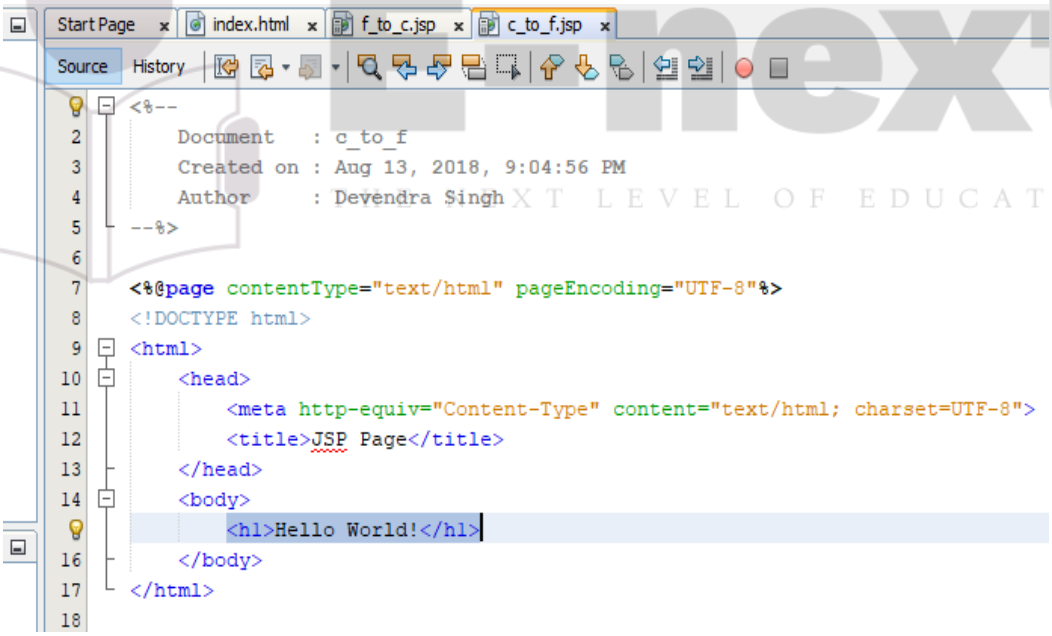
26. New window will appear and then select **temp_service** and click on **OK**.



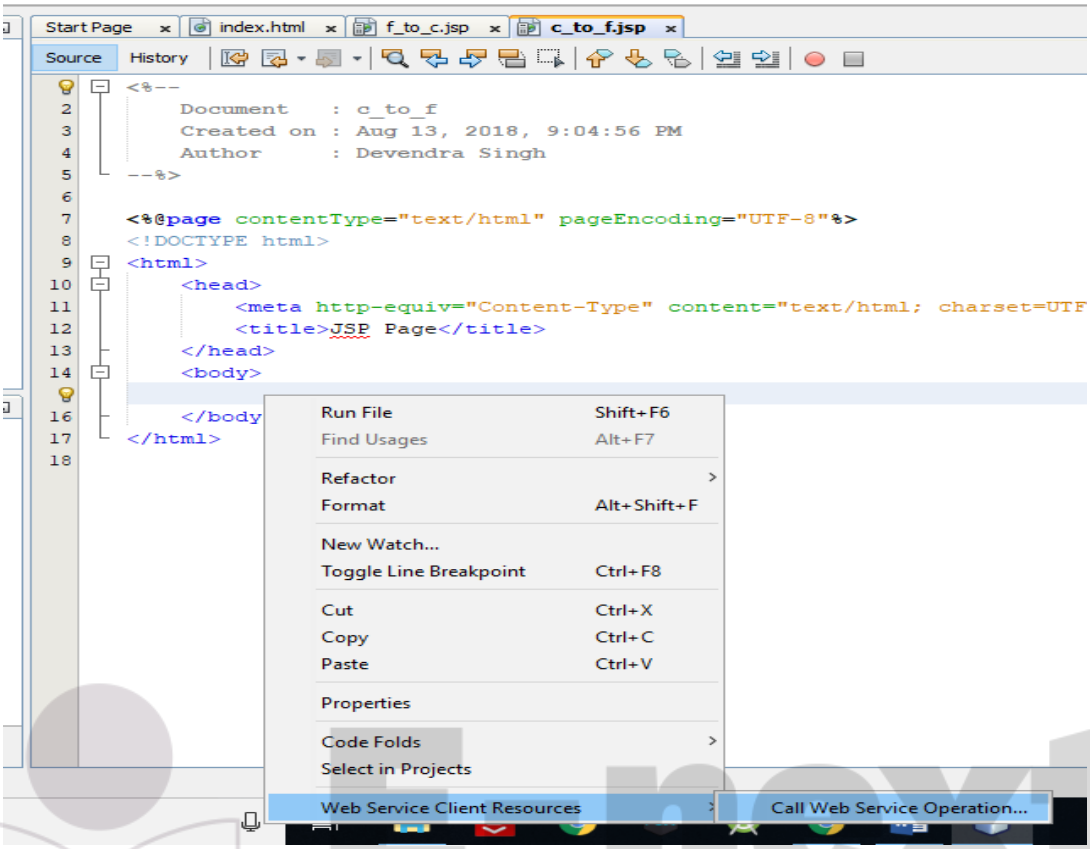
27. Click on Finish.



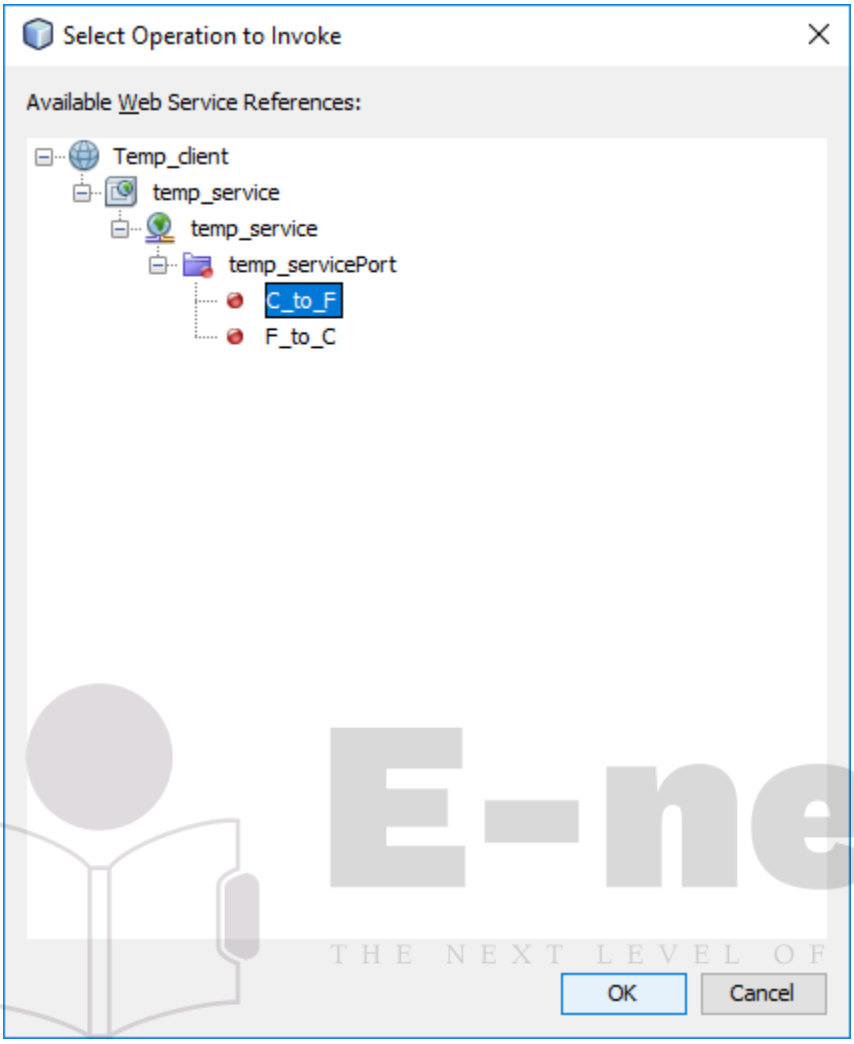
28. Now **open the c_to_f.jsp file** and **delete the selected line**.

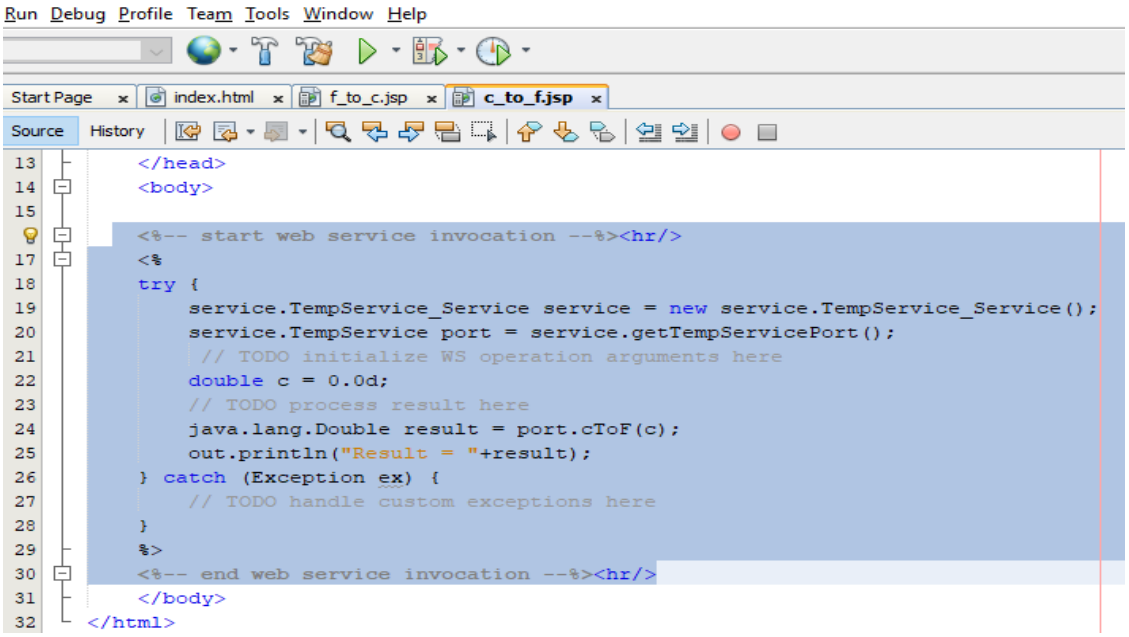


29. Now **right click in between the body section** and **select Call Web Service Operatin** as below.

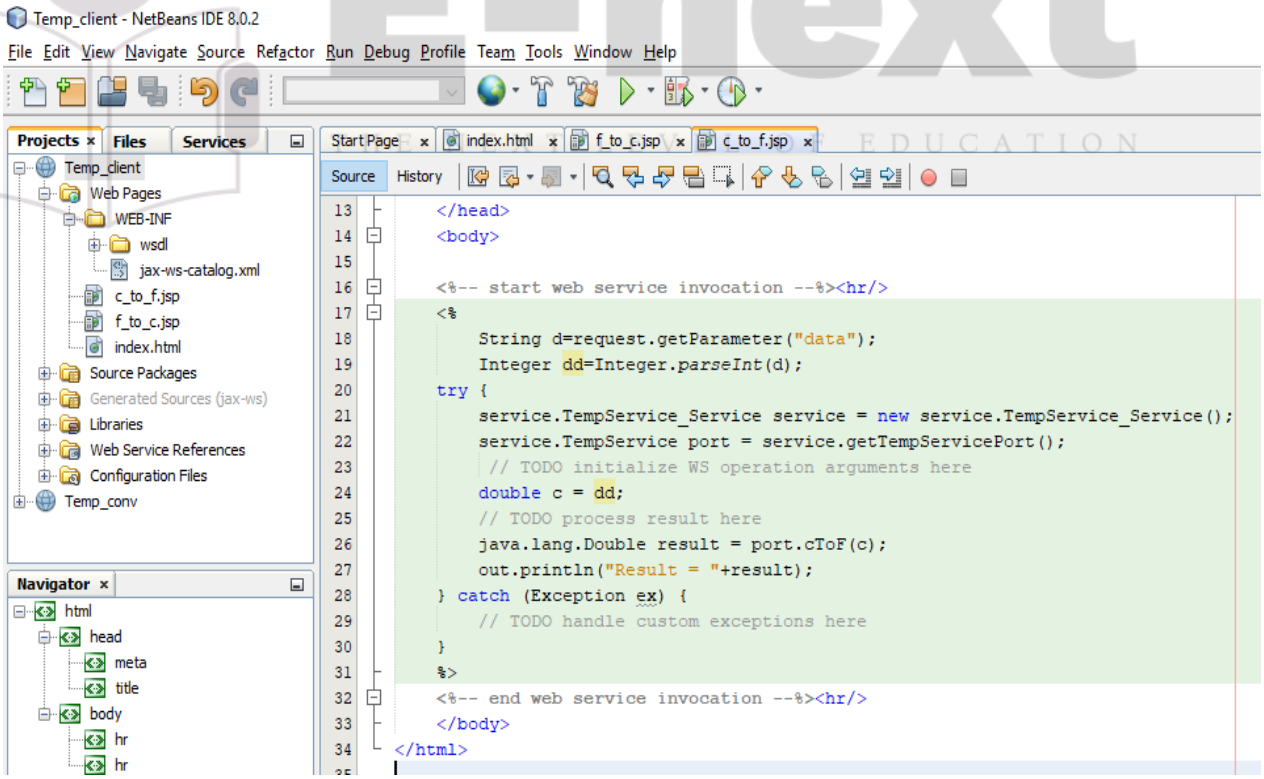


30. New window will appear **select the C_to_F by expanding it** and **click on OK**. So that selected code in second pic will be automatically generated.

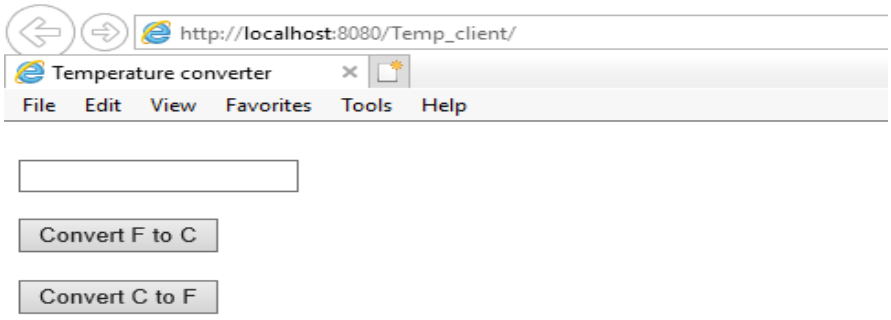




31. Now, **make the selected area in step 30 as like selected area in below pic** by adding some line of code.



32. Now **Open the f_to_c.jsp file** and **follow the steps from 28 to 31**. **Only the change** is in **30 number step** and i.e. **instead of C_to_F, you have to select F_to_C.**
33. Now **run the Temp_client project**. An window will be open like below.

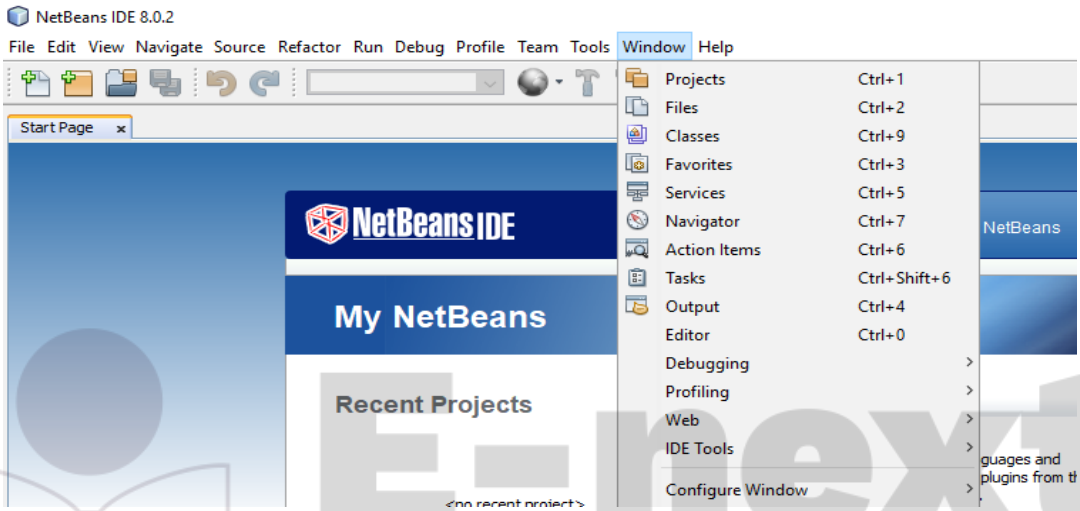


34. Now you can to enter any numeric data into textbox and if you will click the first button it will convert the numeric value into Celsius and vice-versa for the second button.
35. There are so many methods to consume the web service. But I found it easy.

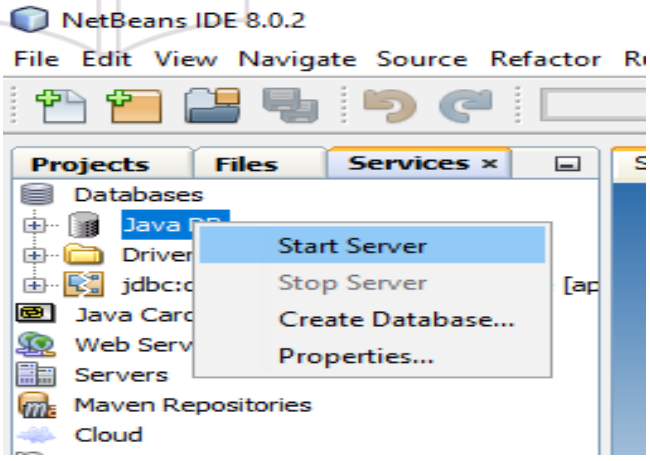
PRACTICAL-2

AIM: Write a program to implement the operation can receive request and will return a response in two ways. a) One - Way operation b) Request –Response.

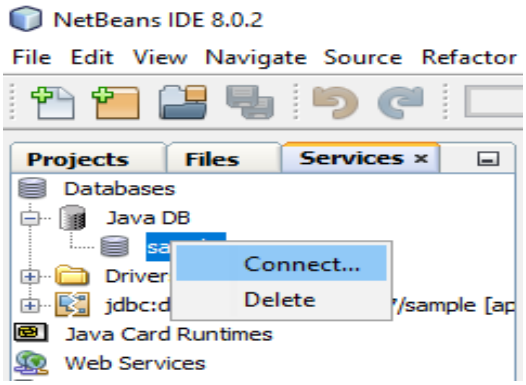
1. Click on Window menu and click on **Projects, Files** & **Services** to open it.



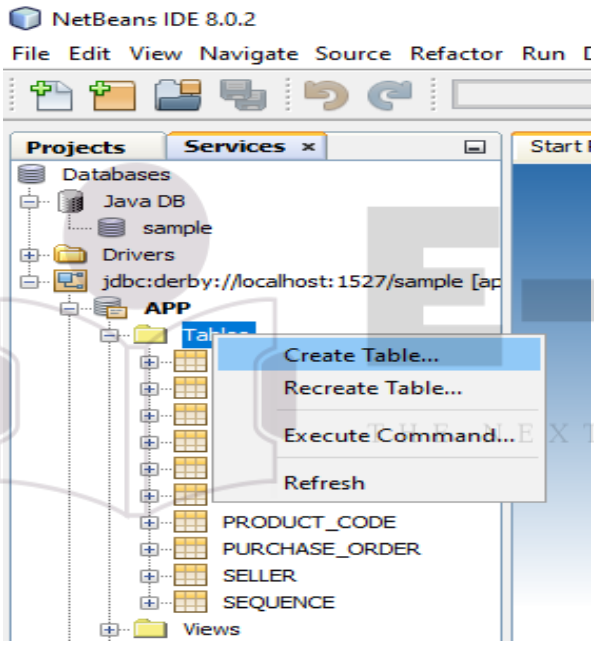
2. **Right click on Java DB** and then **click on Start Server** to start the server .



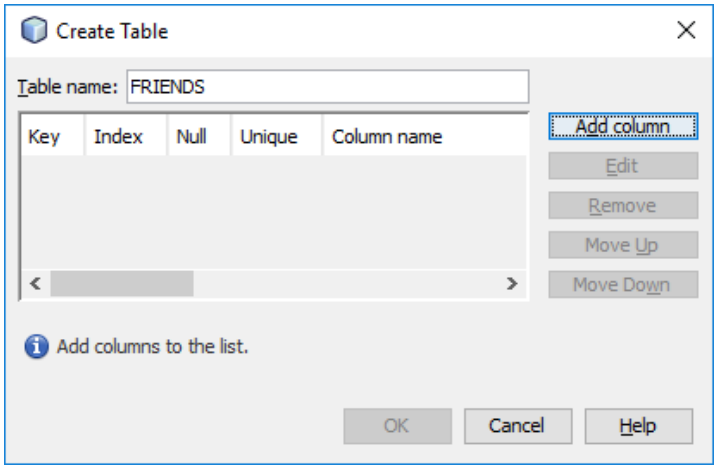
3. Now expand Java DB and **right click on sample** and then **click on connect** to connect the sample database with server.



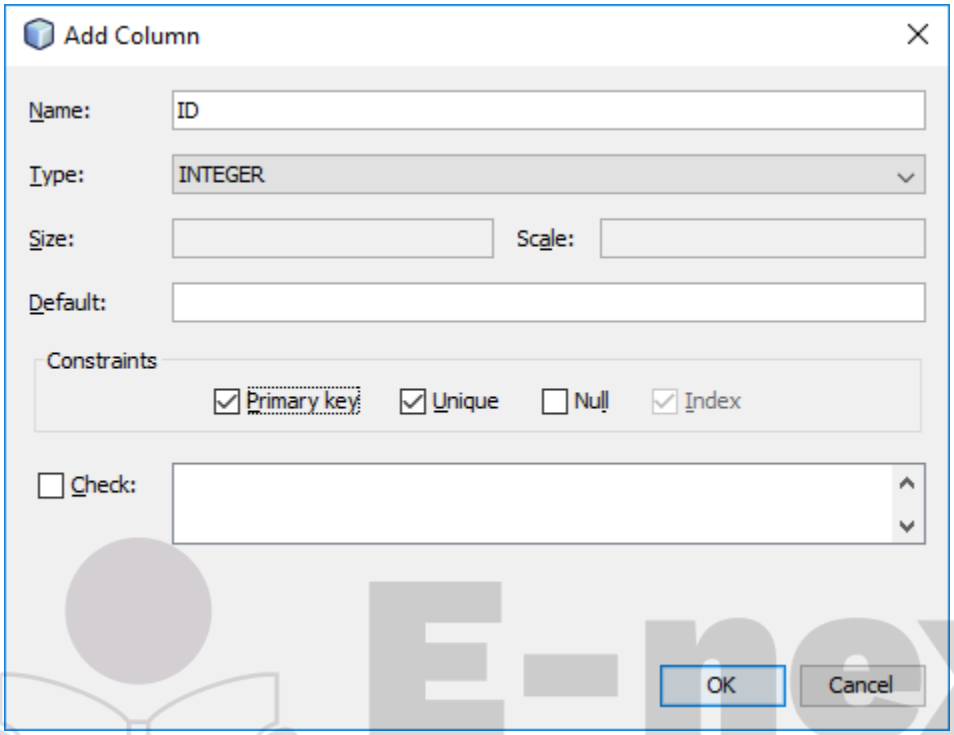
4. Now we are going to create a table in default database **sample**.
   Right click on Table -> Create Table



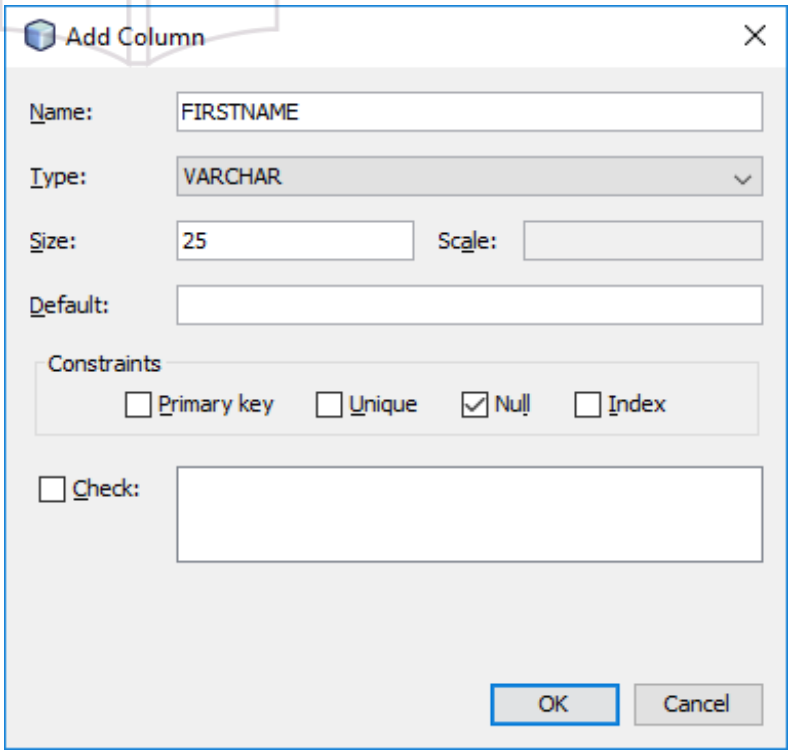
5. Give **table name** as **FRIENDS**.



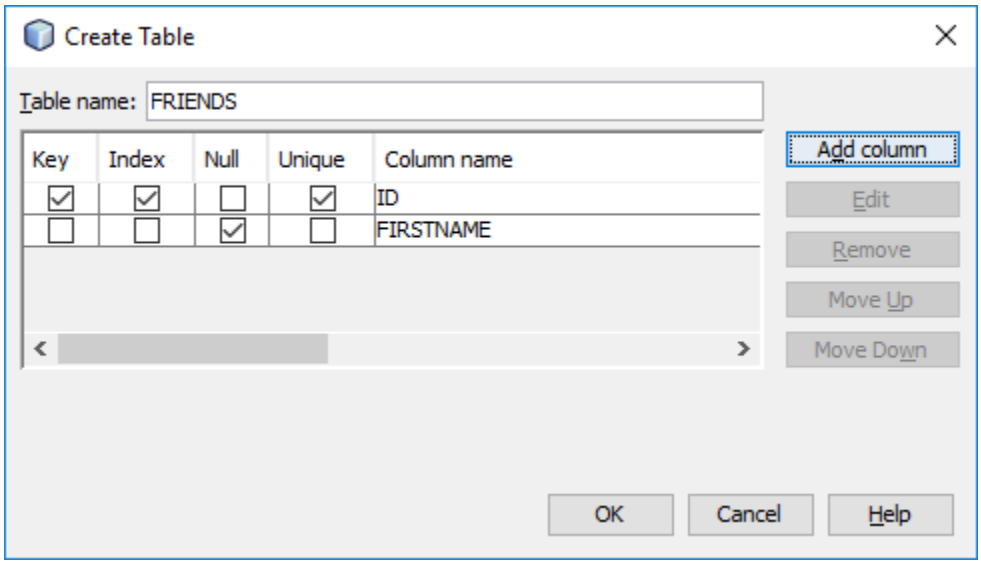
6. Now **click on Add column** button **to add columns** in table.
   Enter details as in below pic and **select Primary key**. After that **click on OK** button.



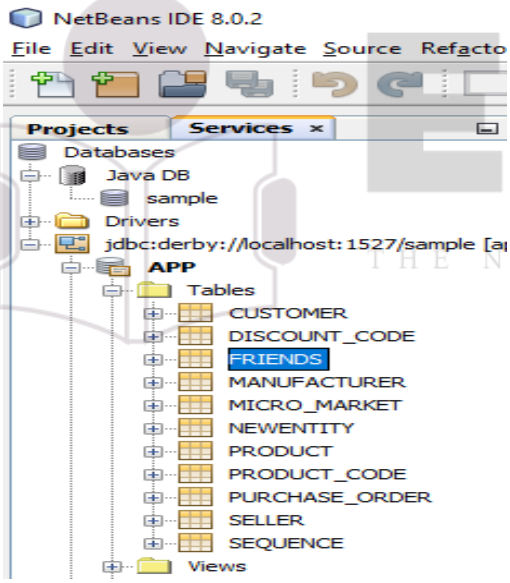
7. Now **add second column** with following detail. But **don't select primary** and click on OK button.



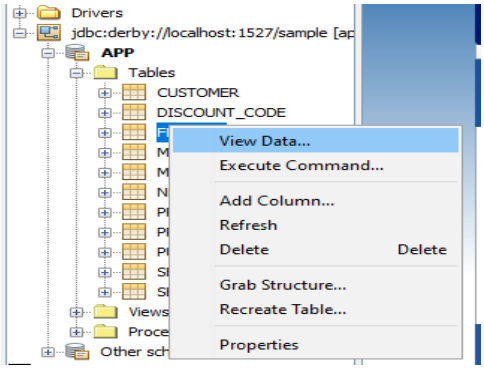
8. Now **click on OK** button.



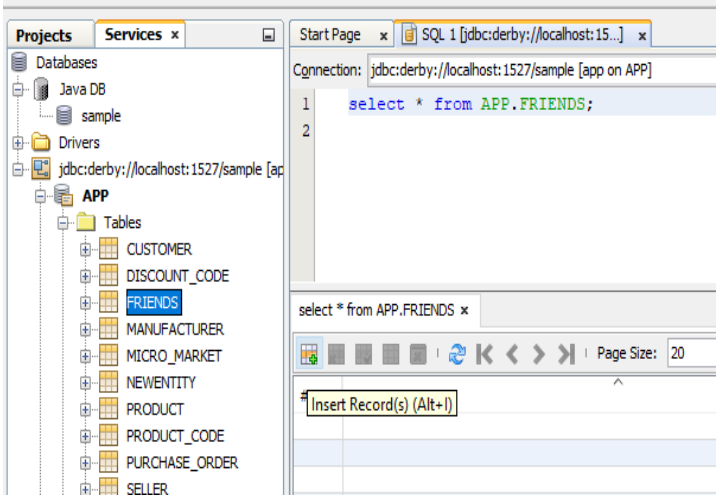
9. Now you can see a table with name **FRIENDS** in the table.



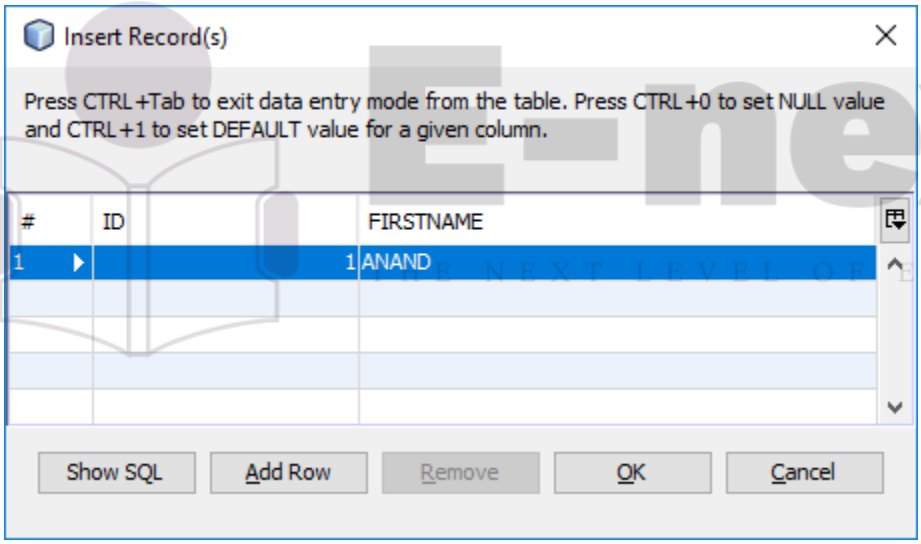
10. Right click on FRIENDS to view and add records into it.



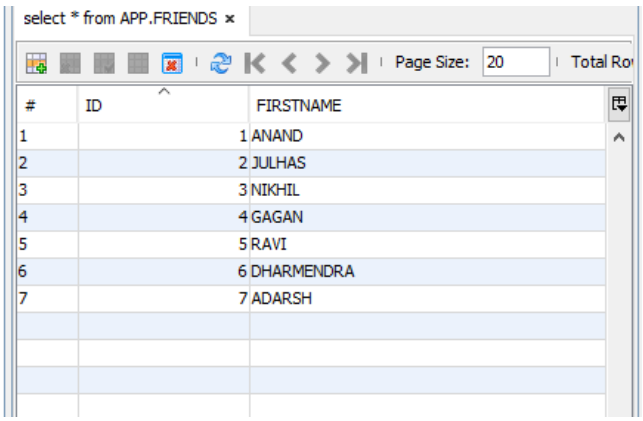
11. Now **click on the leftmost icon in second panel to insert some record**.



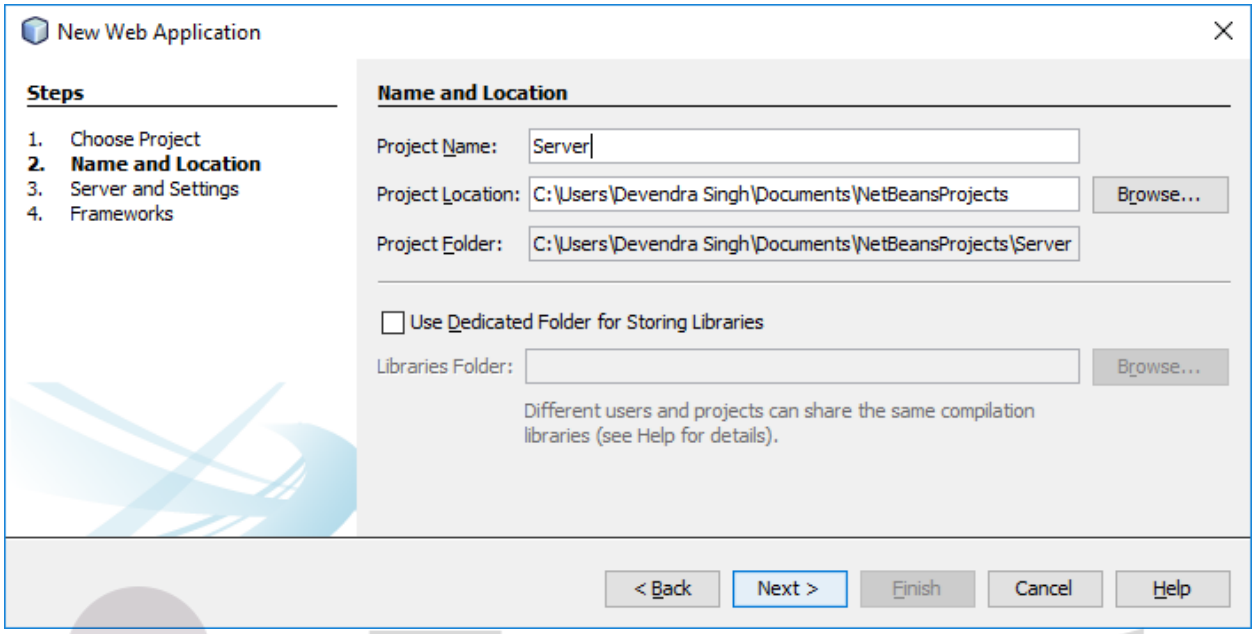
12. Insert a record and then **click on Add Row button to insert more record**. After that **click on OK button** to finish.



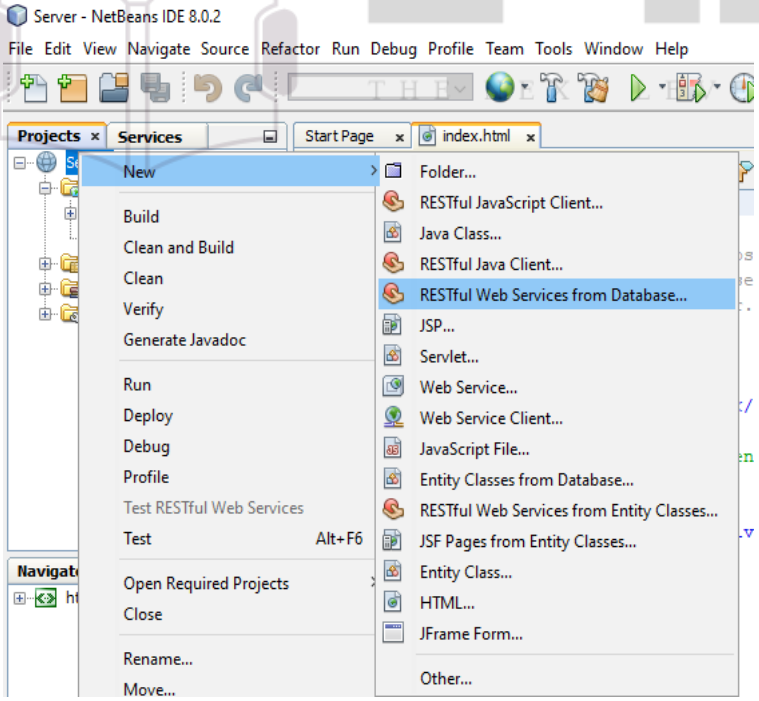
13. As you can see, I have entered 7 records.



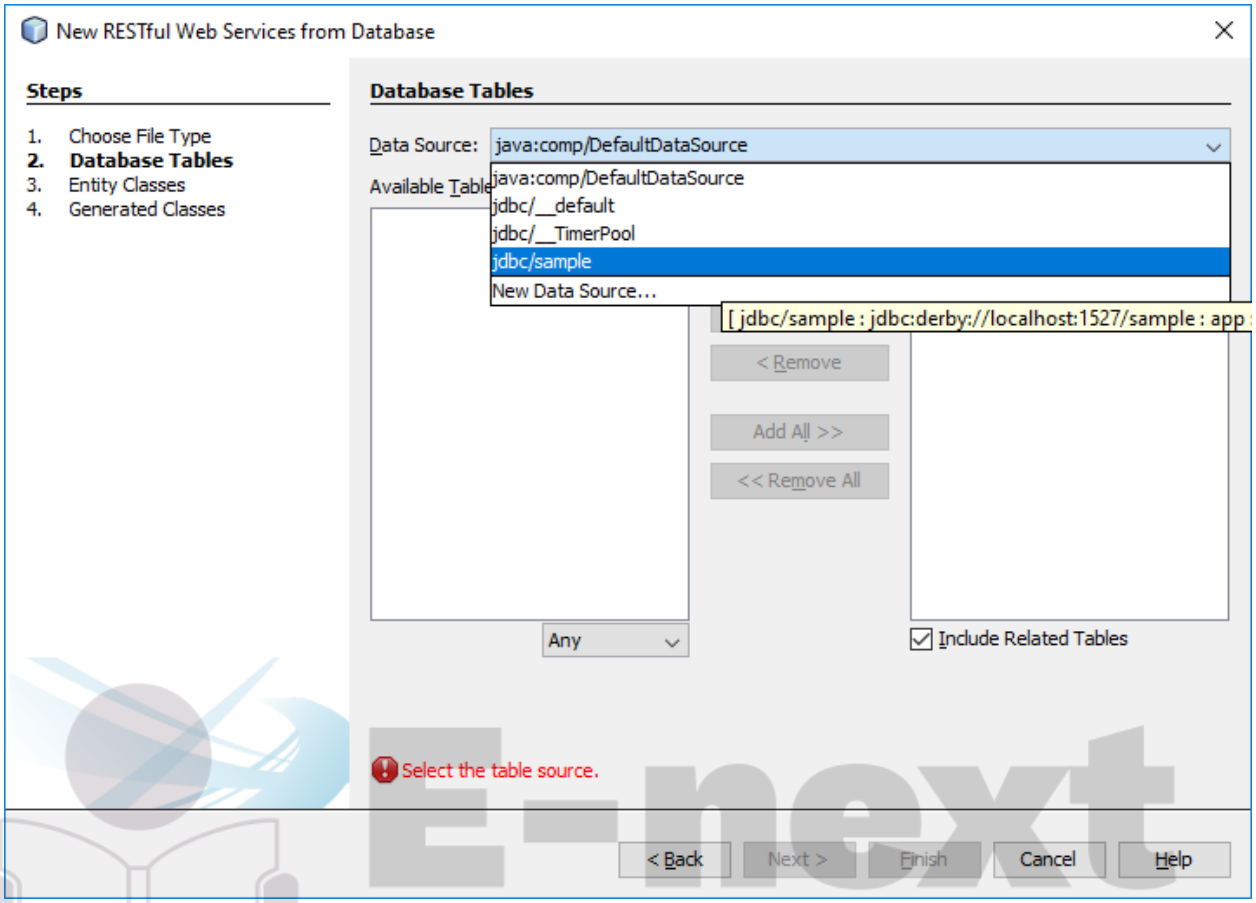
14. Now **create a web application** with name **Server**. After that **click on Next and then Finish** button.



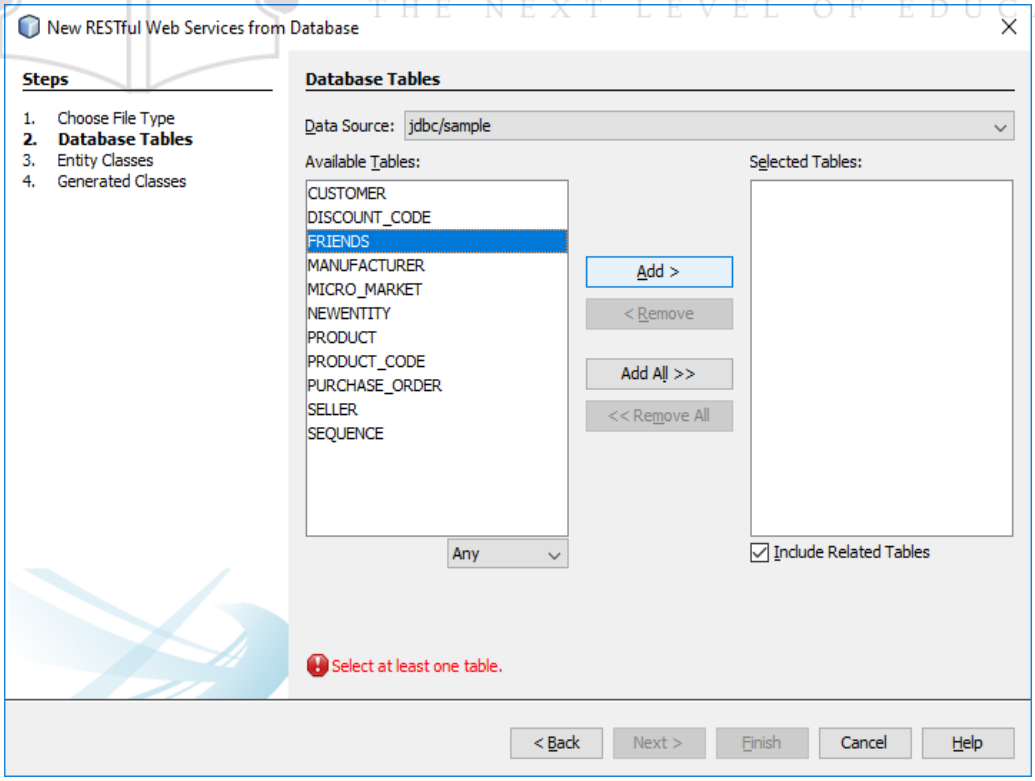
15. Now **create a RESTful Web Service from Database by right click on project** name.



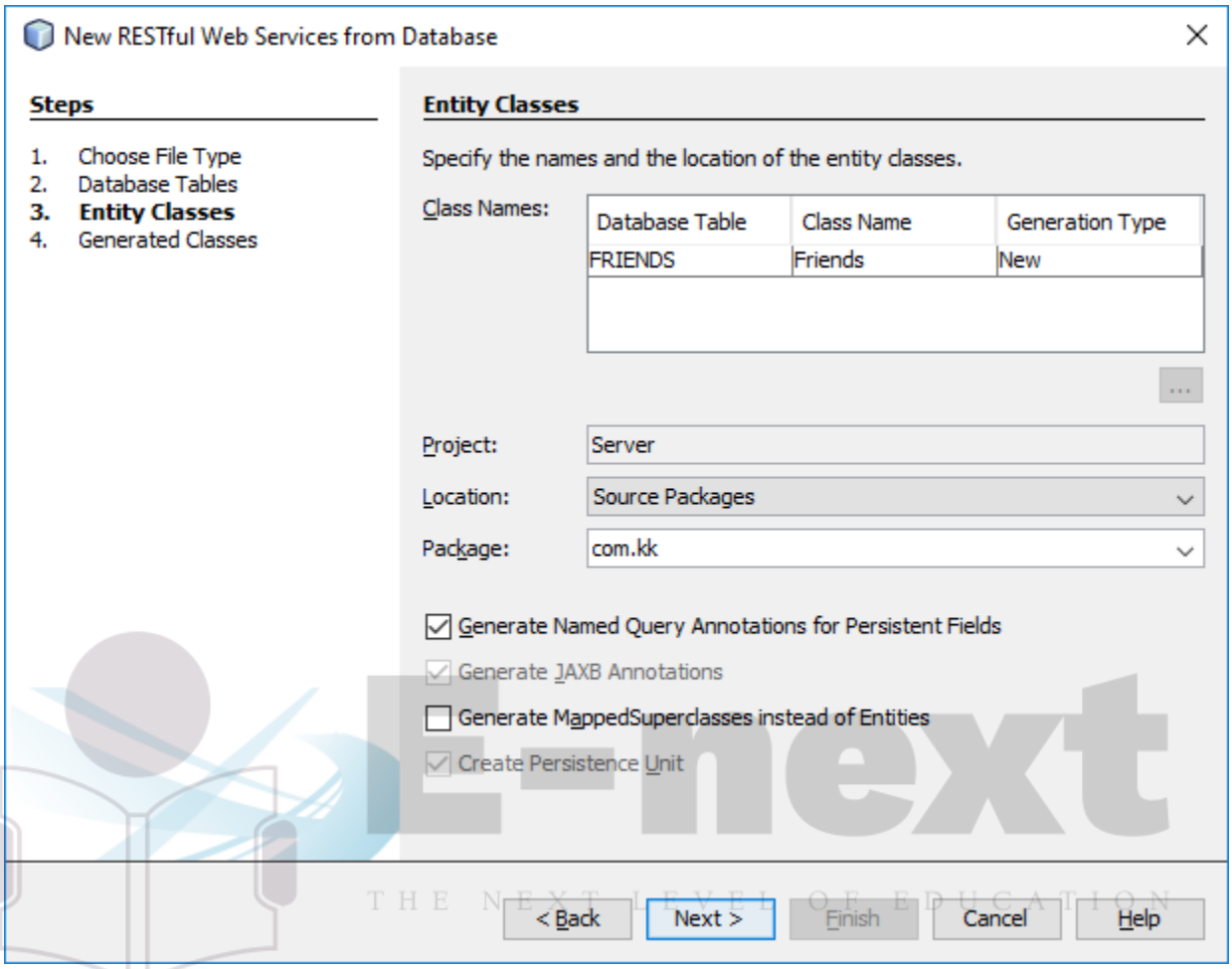
16. Choose **Data Source jdbc/sample**.



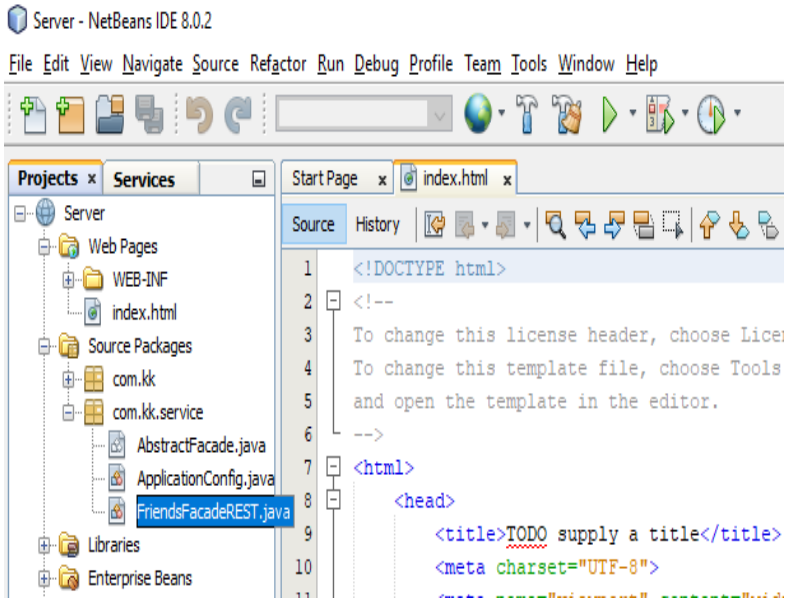
17. Now **select FRIENDS** and **click on Add button**. After that **click on Next** button.



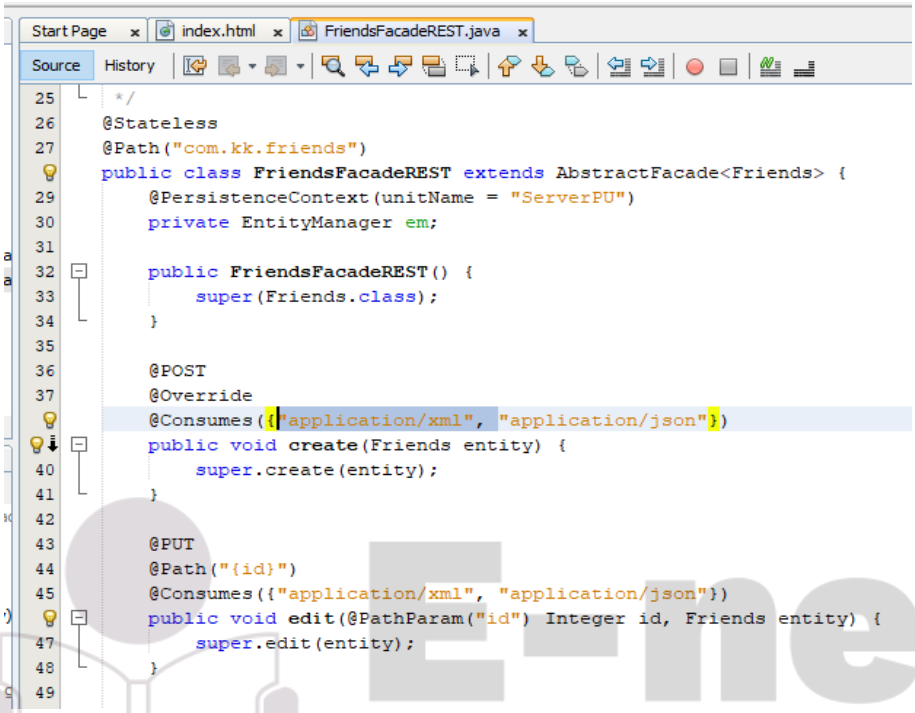
18. Enter Package name as **com.kk** and **click on Next** button and **then Finish**.



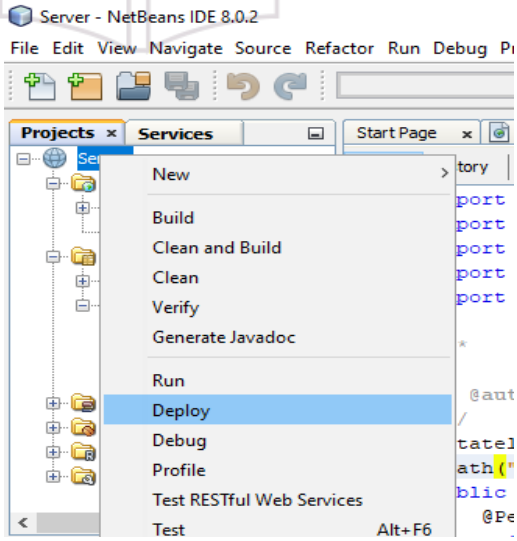
19. Now **open selected file by double click** on it.



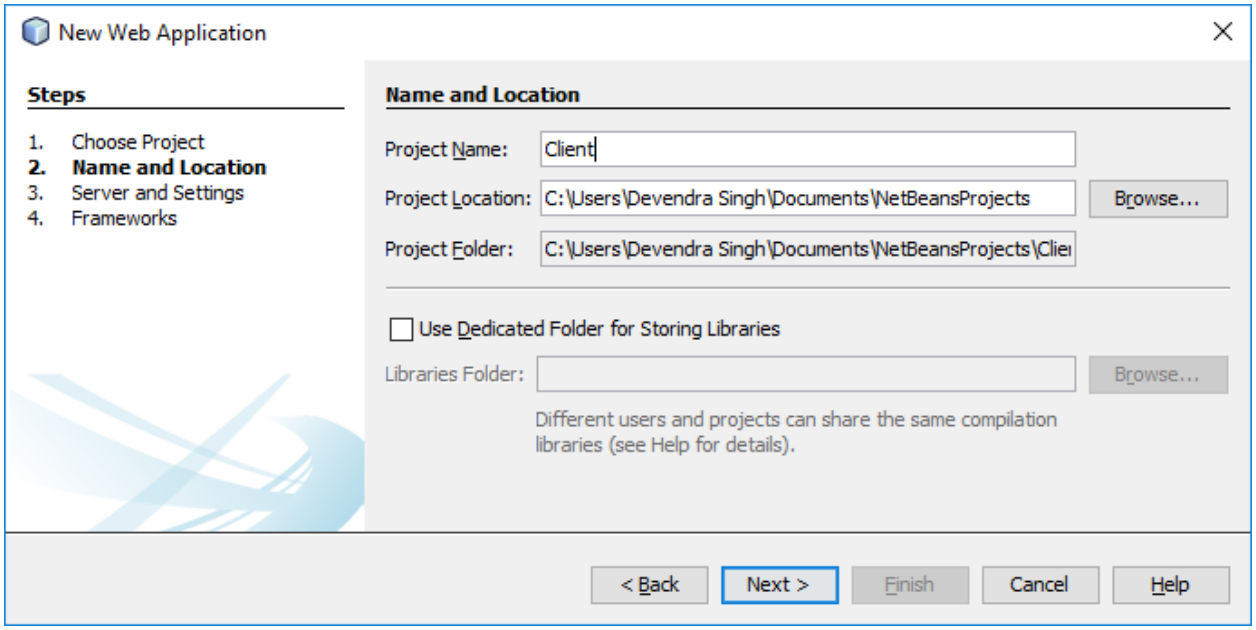
20. Now **remove the selected part from every method in this file**. So that it will communicate only in JSON format. You can also use methods to convert it. But this is easiest method.



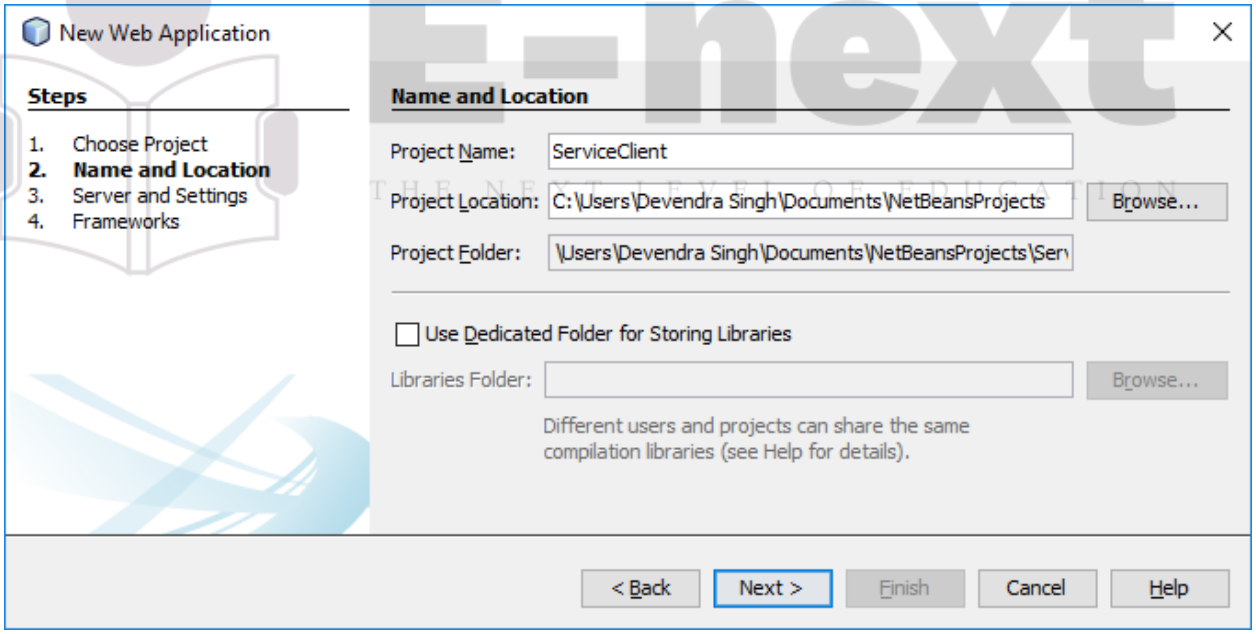
21. After that **right click on project name and Deploy** it.



22. Now **create one more Web Application** as **Client**. After that **click on Next** and then **Finish** button.

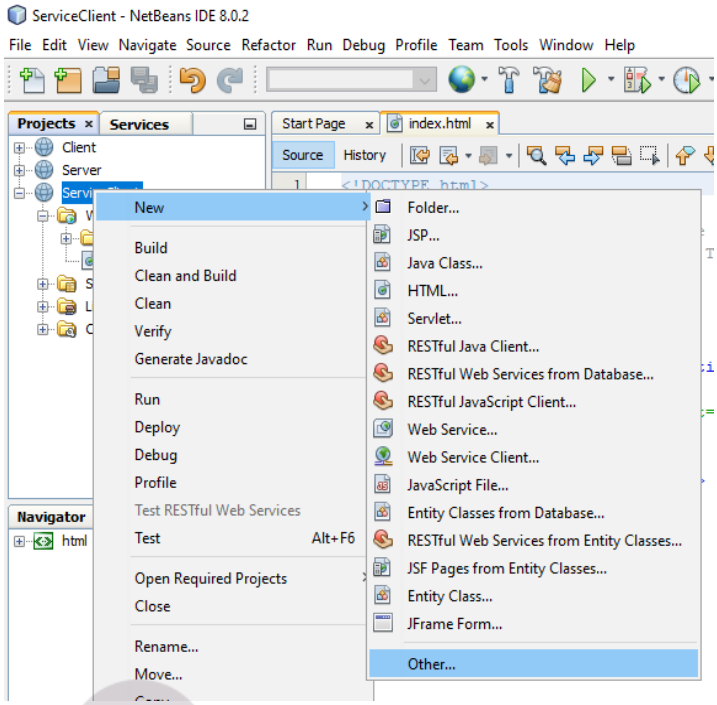


23. Create a **Web Application** with name **ServiceClient**.

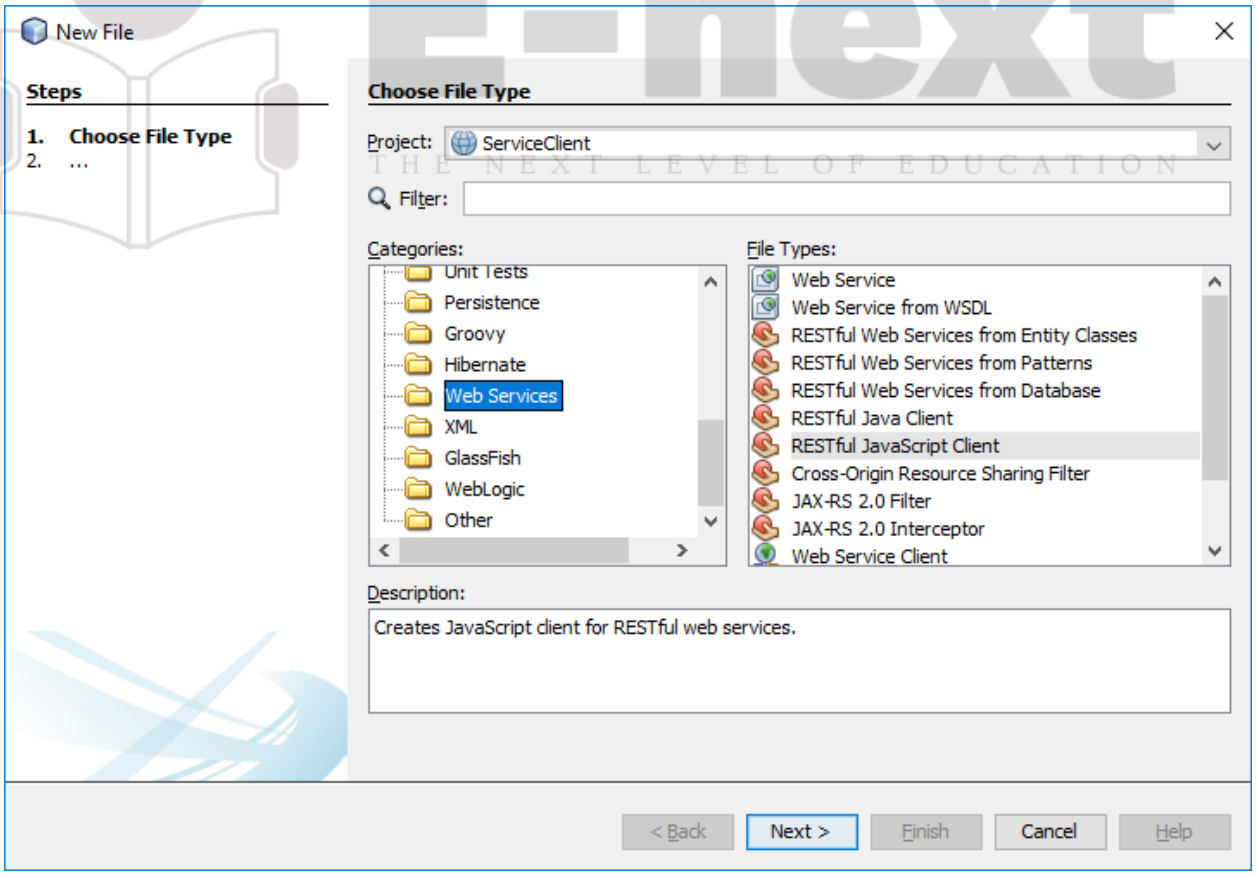


24. Now **create a RESTful Java Client**.

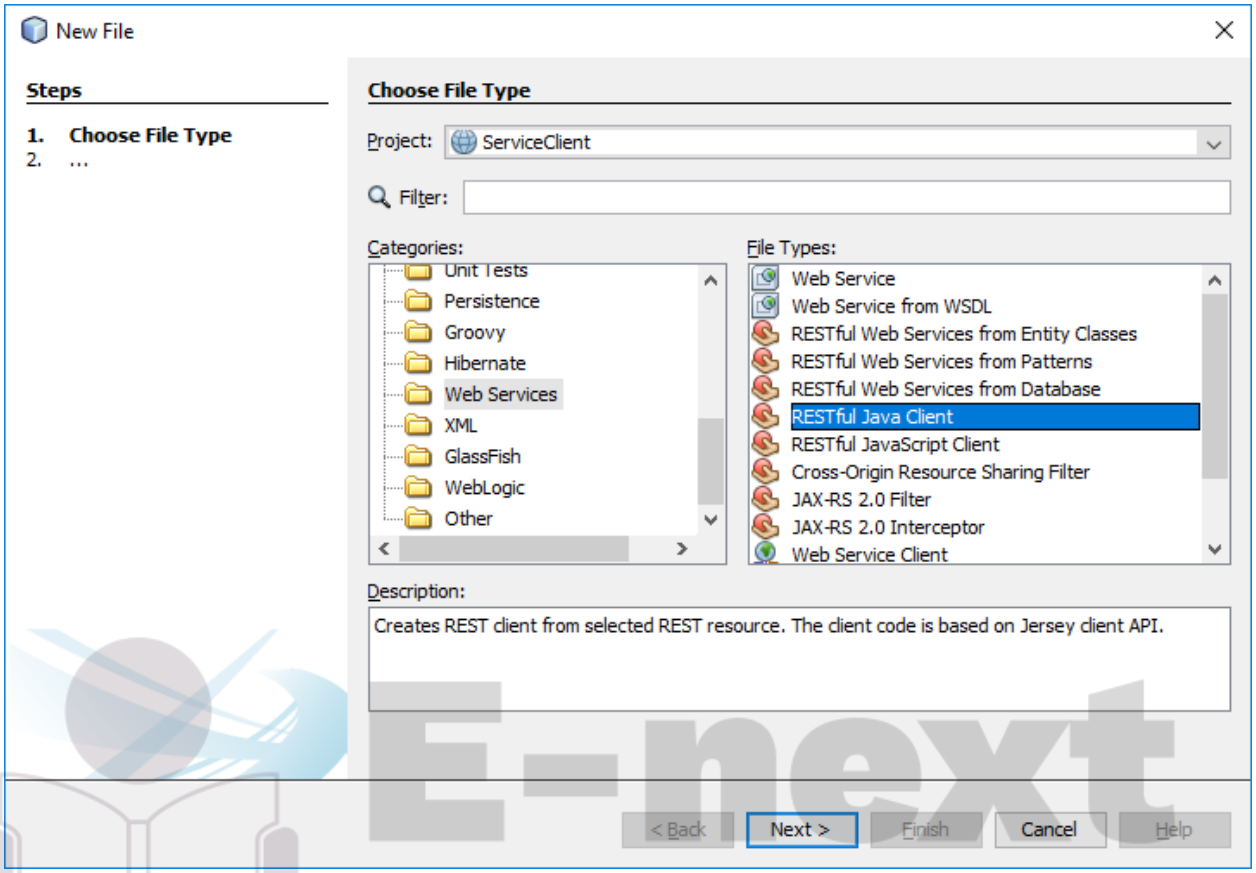
Right click on ServiceClient -> New -> Other.



25. Drag down and select Web Services and **in side panel select RESTful Java Client**.



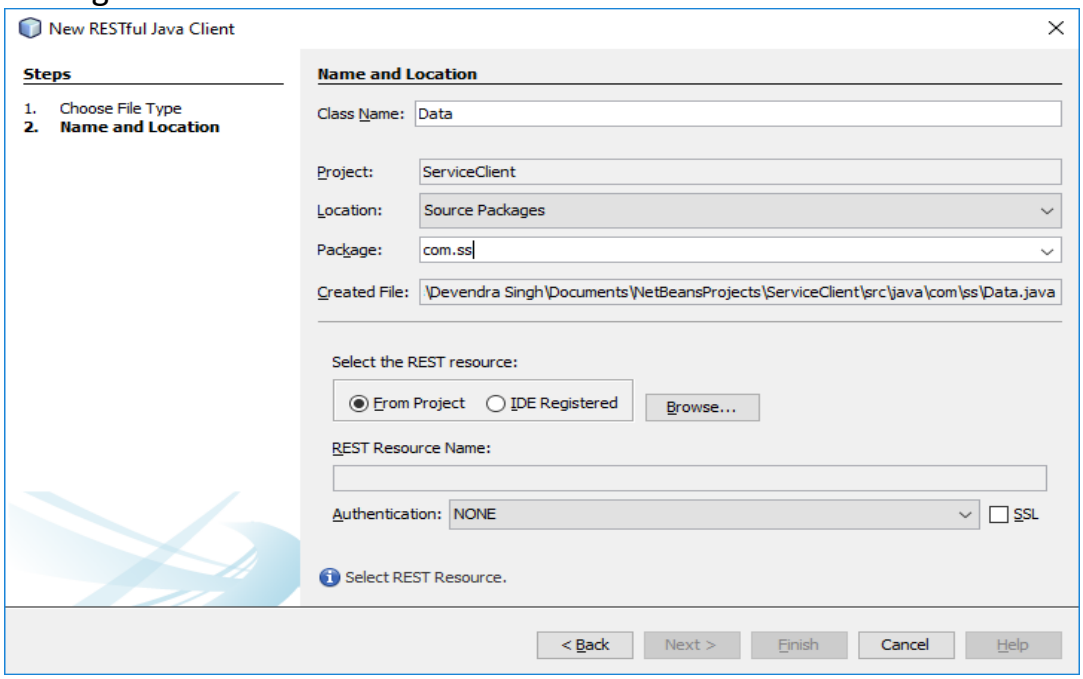
26. After **select RESTful Java Client click on Next**.



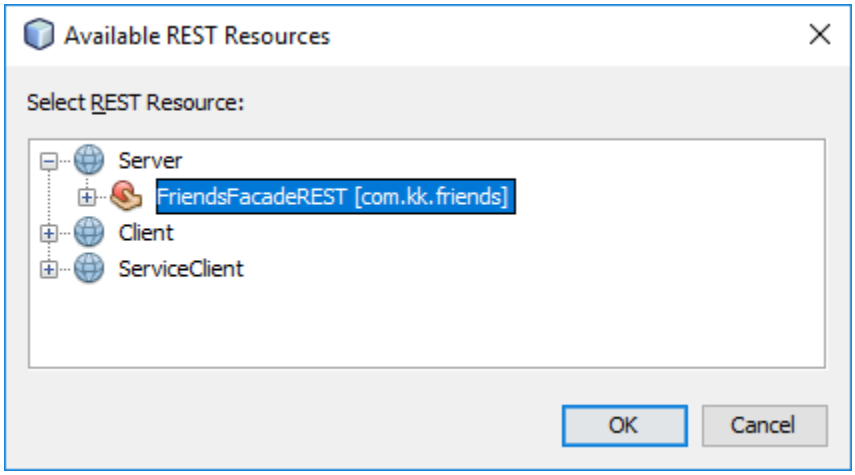
27. Enter following data.

Class Name -> **Data**

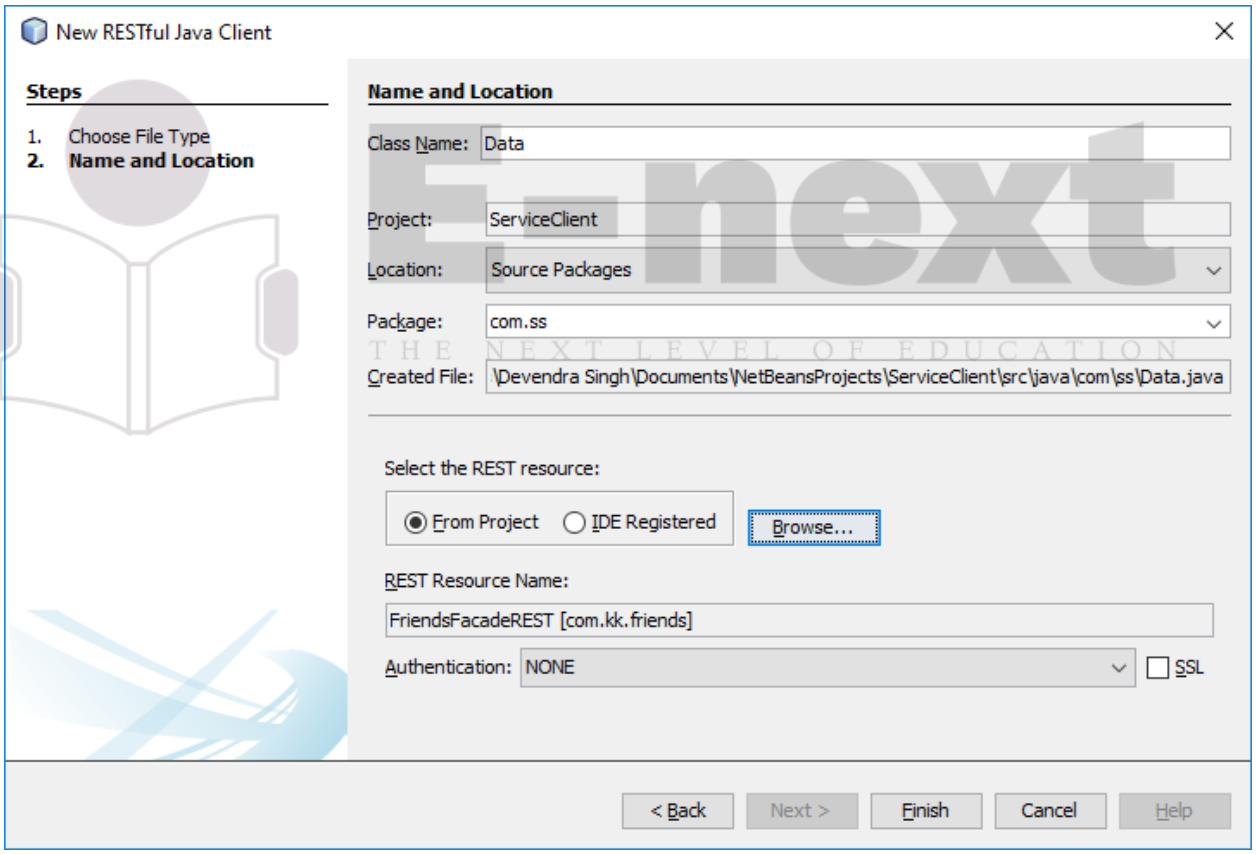
Package -> **com.ss**



28. Now **click on Browse button** and **select the option into the below pic**. After select **click on OK button**.

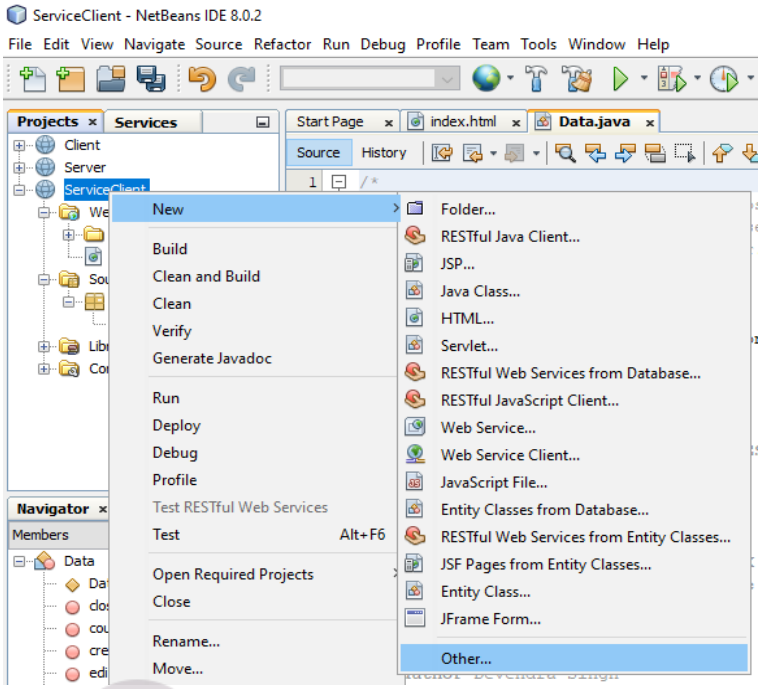


29. Click on **Finish**.

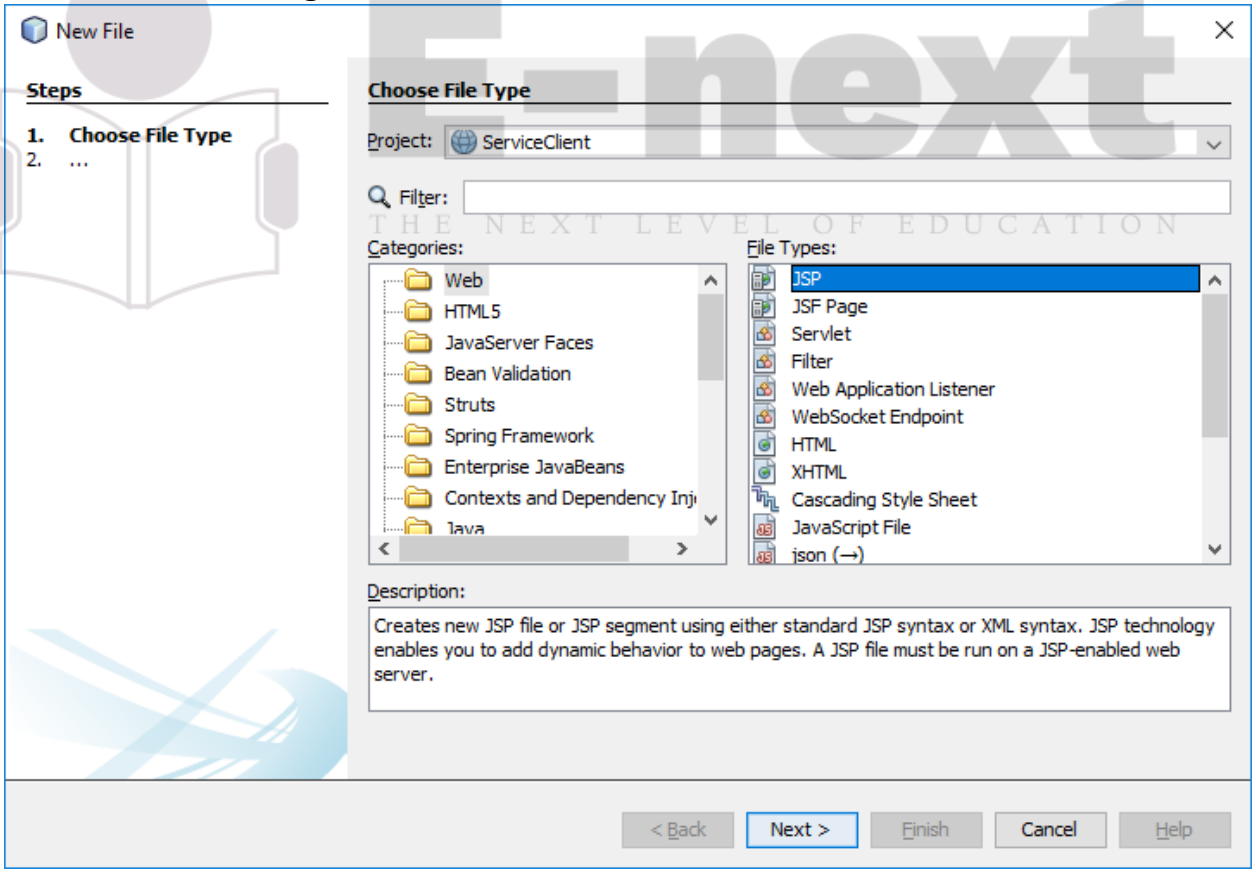


30. Now create a JSP page.

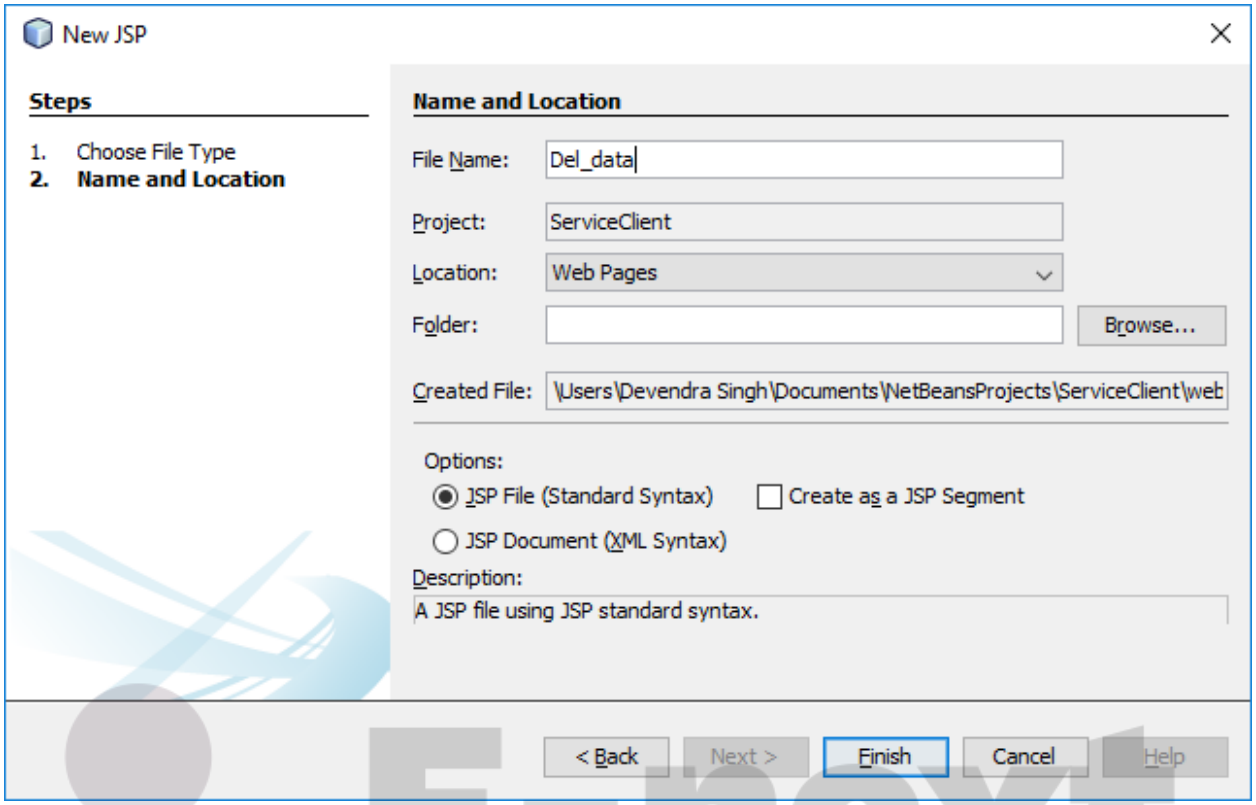
Right click on ServiceClient -> New -> Other



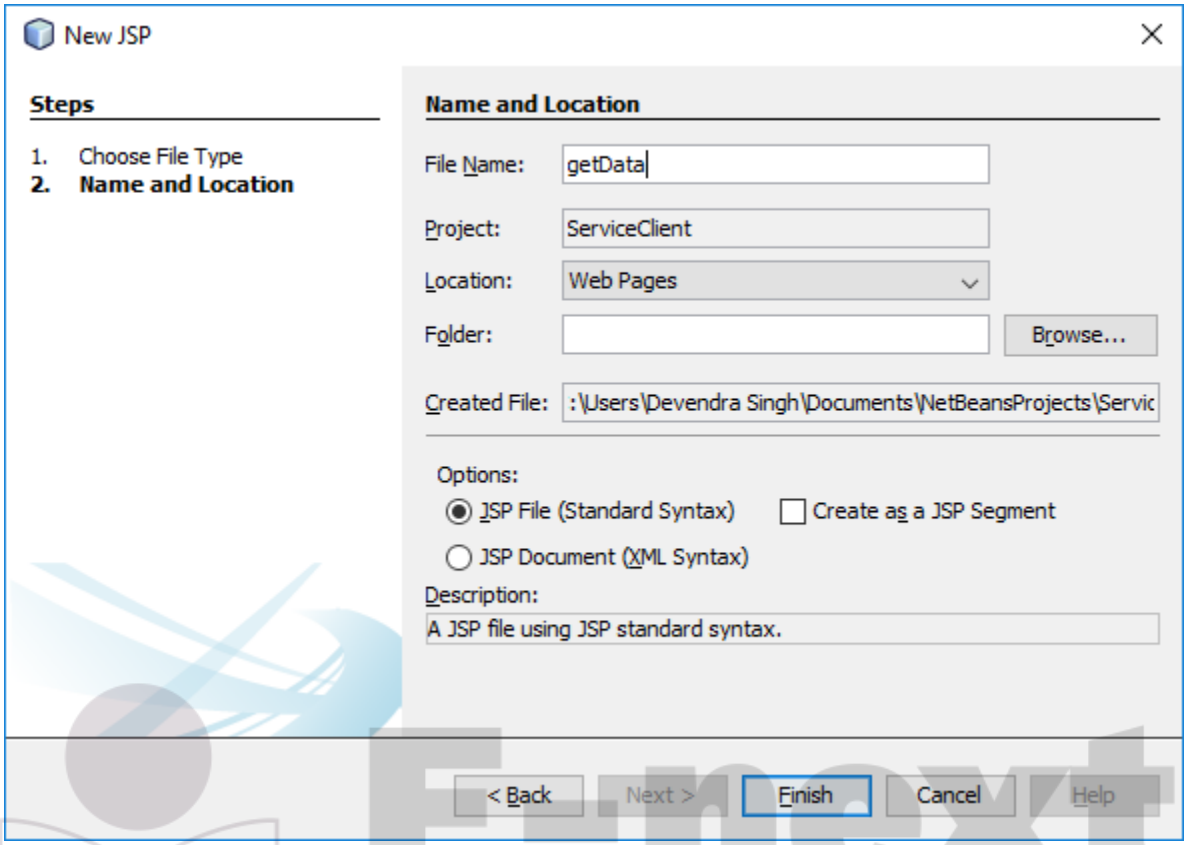
31. Select Web in Categories section -> Select JSP and click on Next button.



32. Enter File Name Del_data and **click on Finish button**.

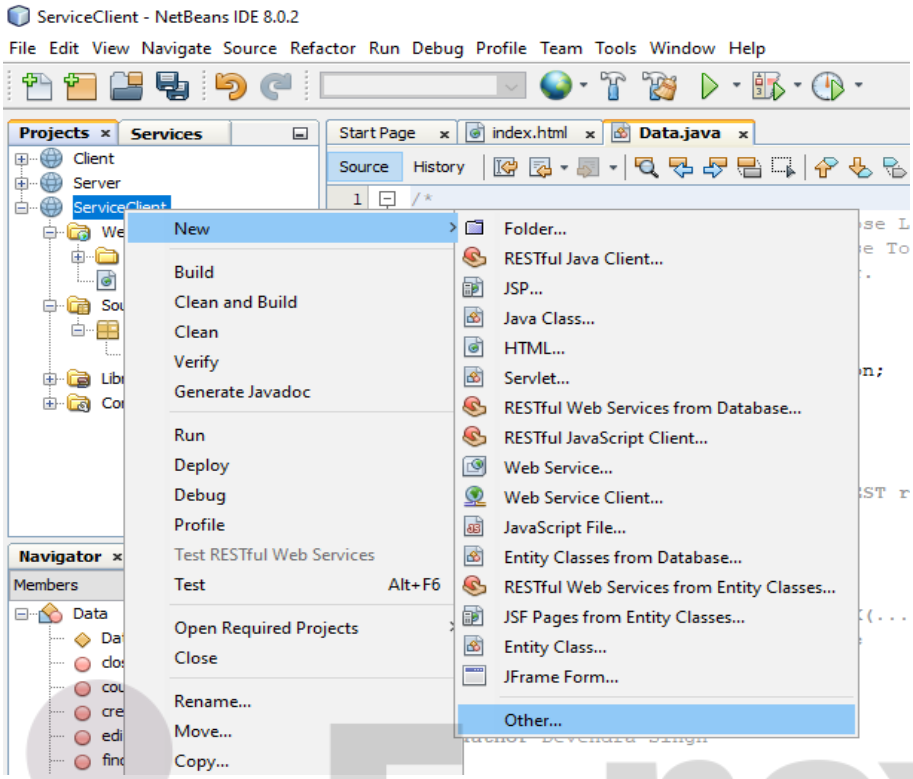


33. Now **create one more JSP file** by **follow the step number 30, 31 & 32**. **But File Name will be getData**.

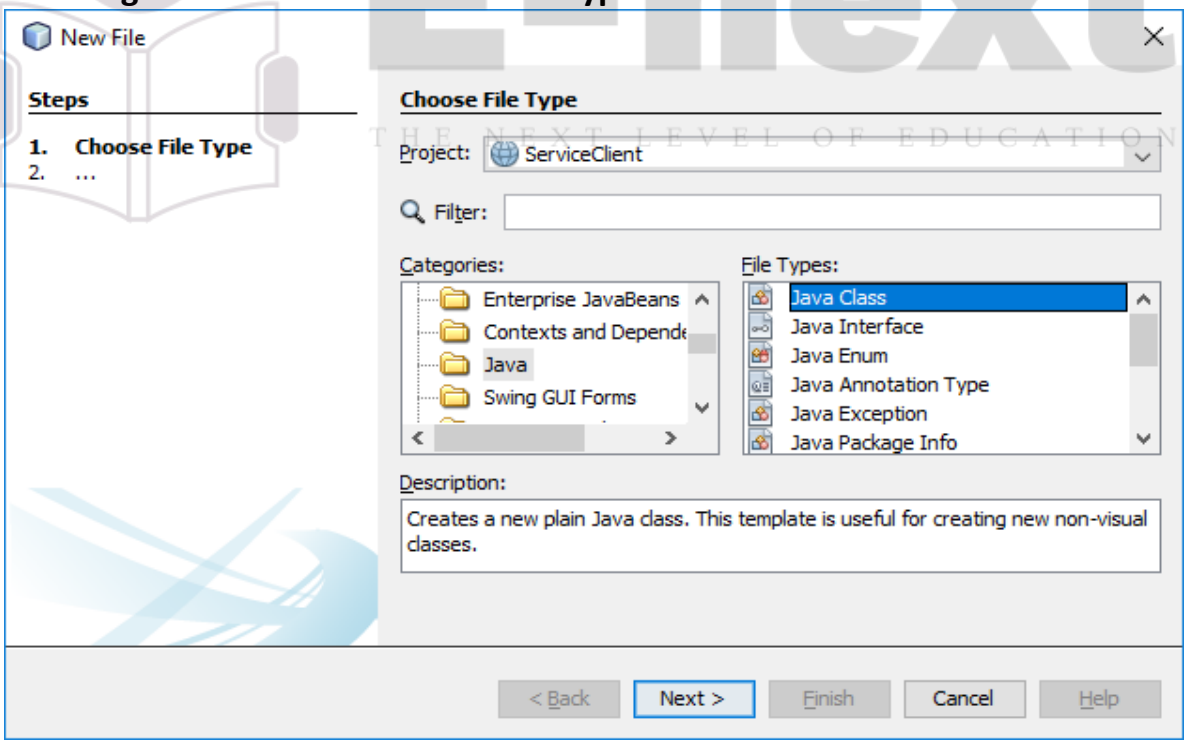


34. Now create a Java class.

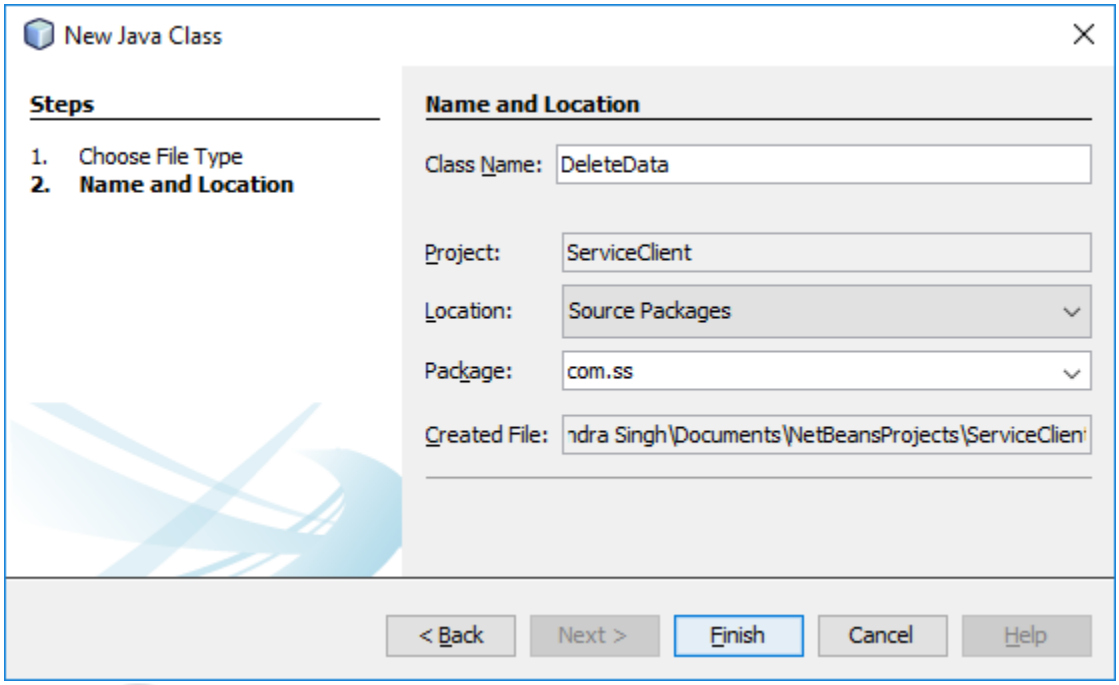
Right click on ServiceClient -> New -> Other



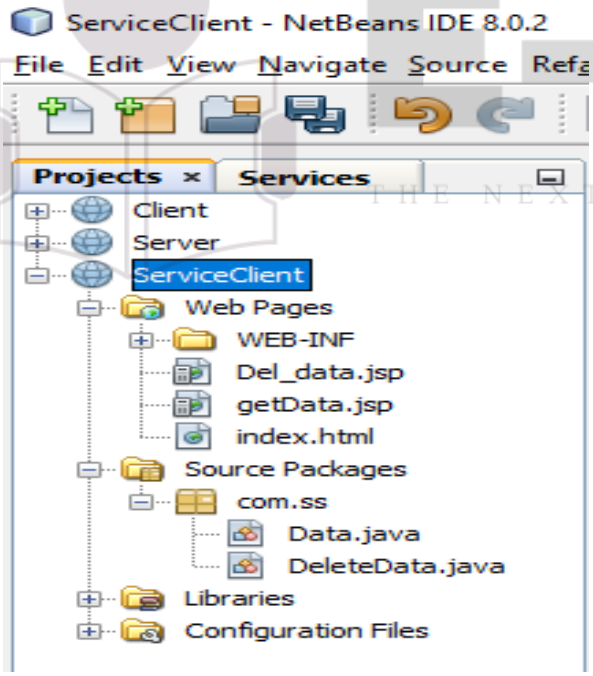
35. In Categories select Java and in **File Types select Java Class**. Click **Next** button.



36. Enter **Class Name DeleteData** and **Package com.ss**. After that click on **Finish**.



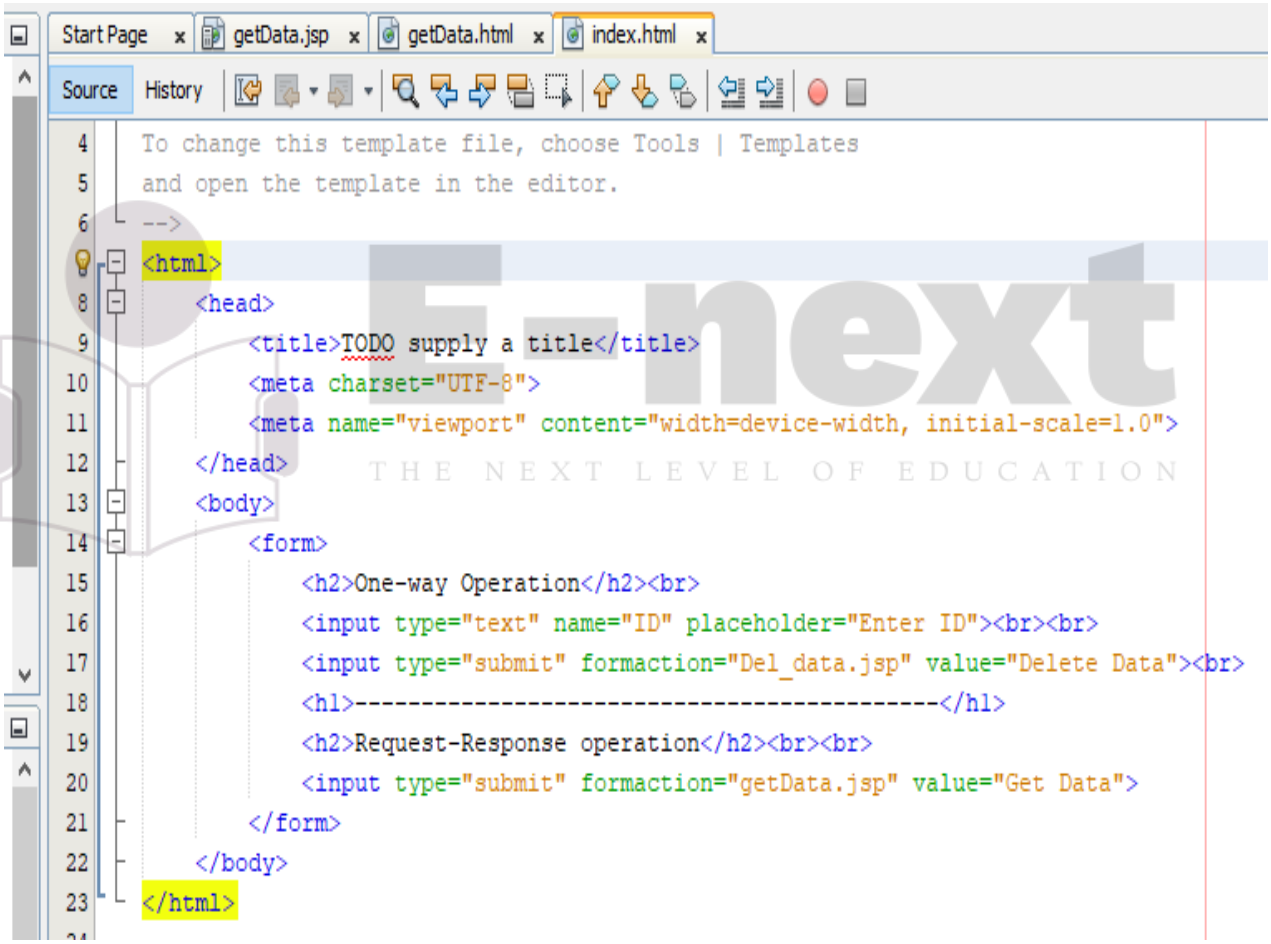
37. Your project file structure will look like below.



38. Now **open the index.html** of ServiceClient project **by double click on it** and **add the following code in between body tag**.

```
<form>
   <h2>One-way Operation</h2><br>
```

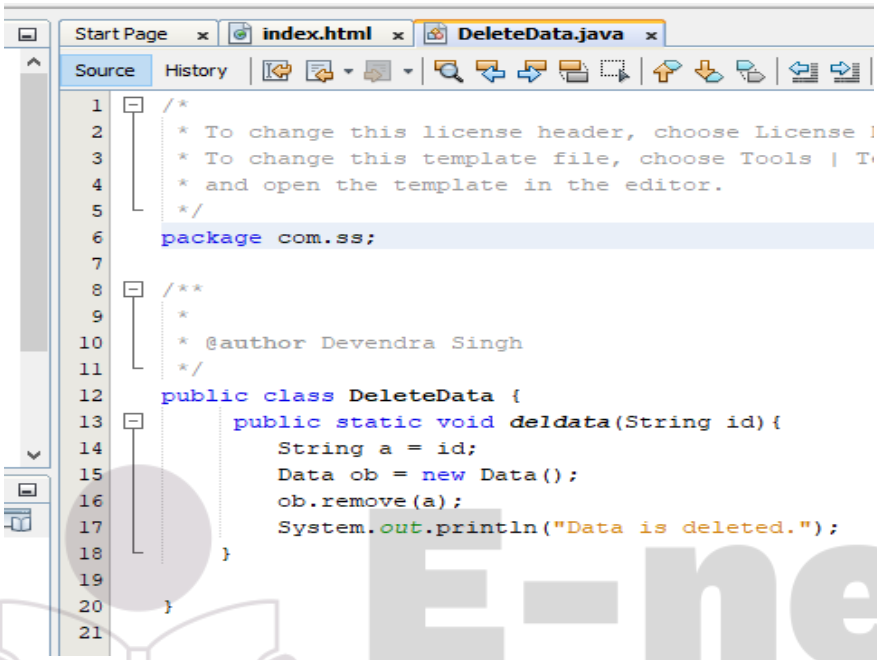
```
    <input type="text" name="ID" placeholder="Enter ID"><br><br>
    <input type="submit" formaction="Del_data.jsp" value="Delete
Data"><br>

    <h1>----------------------------------------</h1>

    <h2>Request-Response operation</h2><br><br>
    <input type="submit" formaction="getData.jsp" value="Get Data">
</form>
```



39. Now **open DeleteData.java file** by **double click on it** and **add the following code in the class** and **save it by pressing Ctrl+S**.

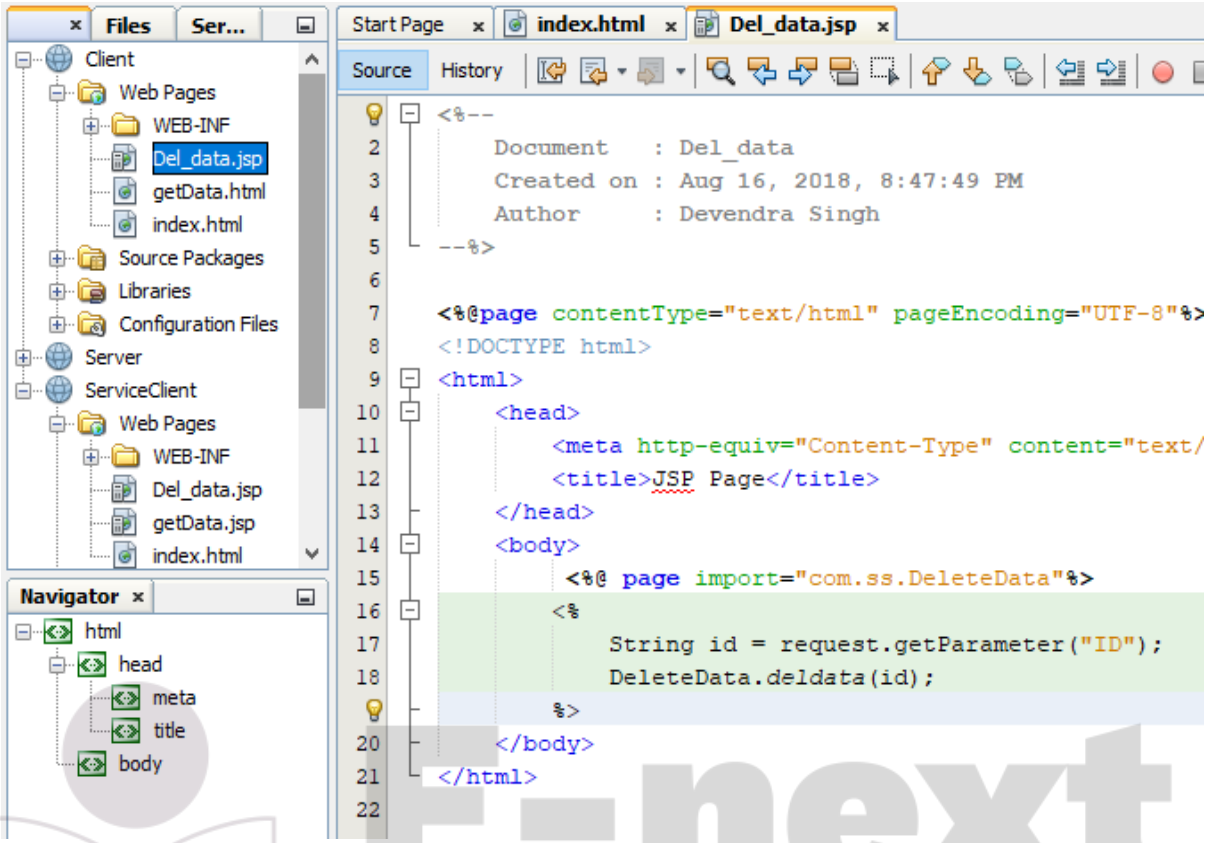
```
public static void deldata(String id){
        String a = id;
```

```
    Data ob = new Data();
    ob.remove(a);
    System.out.println("Data is deleted.");
  }
```



40. Now **open the Del_data.jsp file** and **replace the contents of body** with the following code.

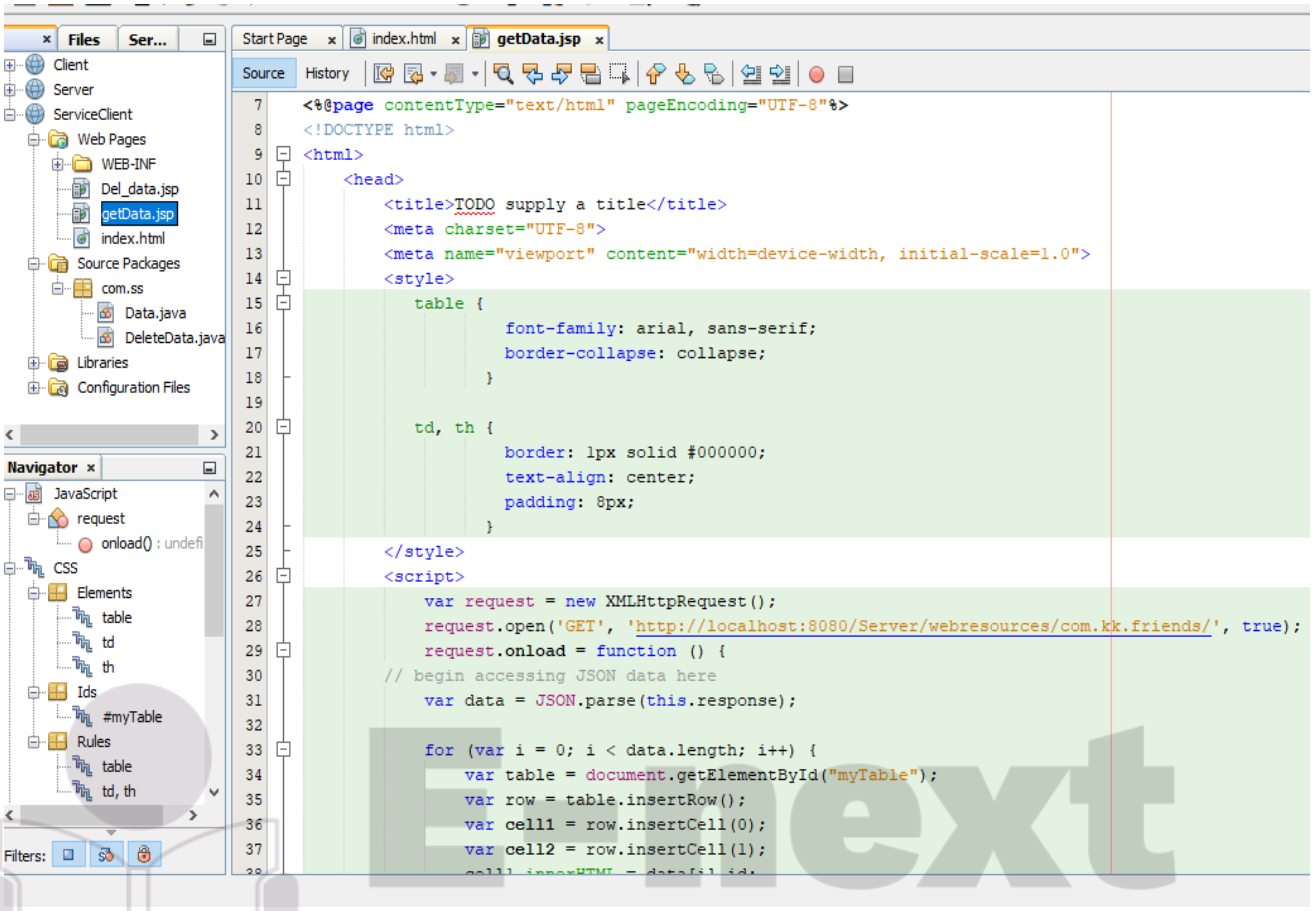

```
<%@ page import="com.ss.DeleteData"%>
    <%
      String id = request.getParameter("ID");
      DeleteData.deldata(id);
    %>
```



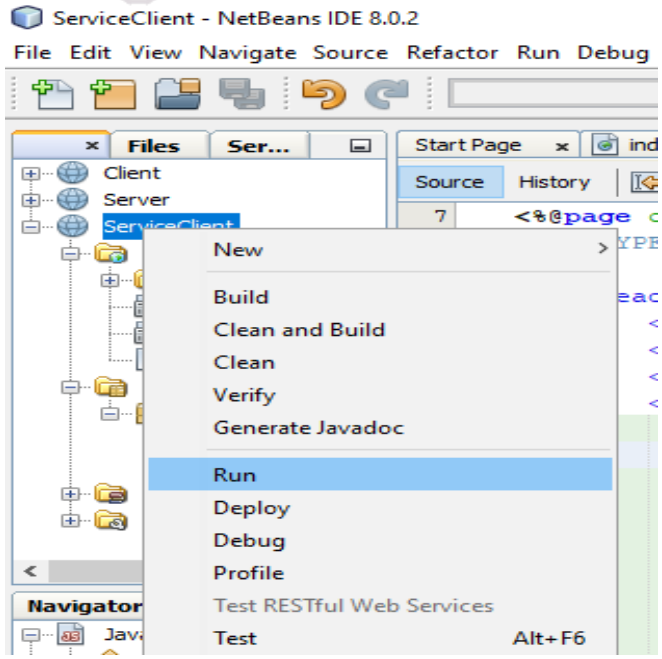
41. Now **open the getData.jsp file** and **replace the contents of html tag** with the following code and save it.

```
<head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <style>
      table {
          font-family: arial, sans-serif;
          border-collapse: collapse;
        }

      td, th {
          border: 1px solid #000000;
          text-align: center;
          padding: 8px;
        }
```

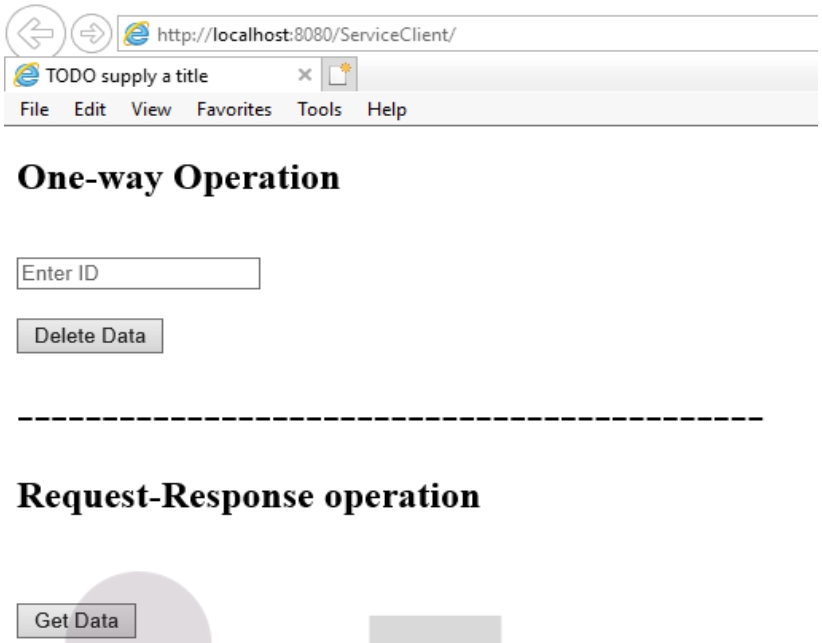
```
    </style>
    <script>
      var request = new XMLHttpRequest();
      request.open('GET',
'http://localhost:8080/Server/webresources/com.kk.friends/', true);
      request.onload = function () {
   // begin accessing JSON data here
      var data = JSON.parse(this.response);

      for (var i = 0; i < data.length; i++) {
         var table = document.getElementById("myTable");
        var row = table.insertRow();
        var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);
        cell1.innerHTML = data[i].id;
        cell2.innerHTML = data[i].firstname;
      }
    };

    request.send();
    </script>

  </head>
  <body>
    <table id="myTable">
      <tr>
        <th> ID</th>
        <th>NAME</th>
      </tr>
    </table>

  </body>
```



42. Now **Run the ServiceClient** project.



43. On run the project following window will open in browser.



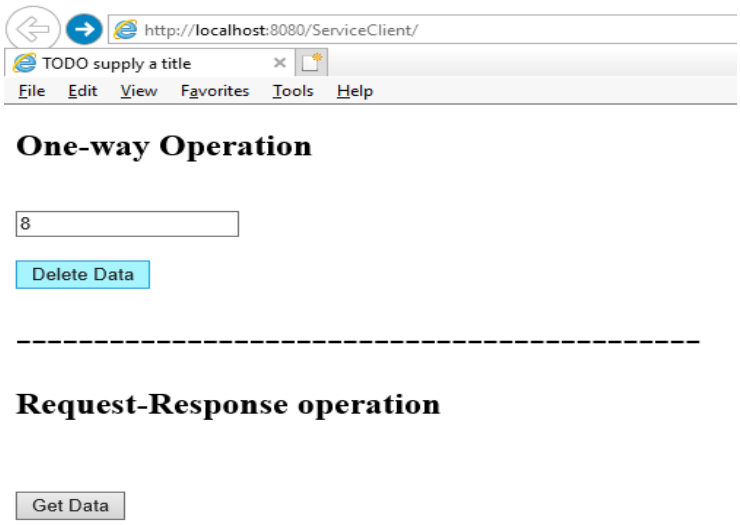
If you will click on Delete Data button, record of entered ID will be deleted.

If you will click on Get Data button Request for the data will be send and it will return the data as response. Hence it is two way operation.

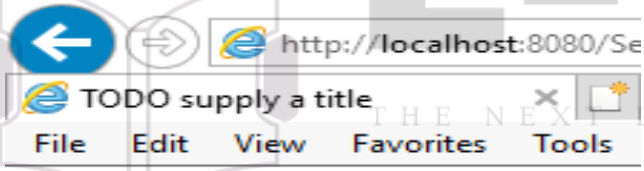
44. By **click on Get Data button**.

ID	NAME
1	ANAND
2	JULHAS
3	NIKHIL
4	RAVI
5	GAGAN
6	DHARMENDRA
7	ADARSH
8	DEVENDRA

45. Entering ID 8 and **clicking on Delete Data button**.



46. Now **go back and again click on Get Data button**. **Refresh the page**, You will see data with id number 8 is deleted.



ID	NAME
1	ANAND
2	JULHAS
3	NIKHIL
4	RAVI
5	GAGAN
6	DHARMENDRA
7	ADARSH

Create new Web Project give it name pract_2

Press next

Press Finish

Right Click on Project pract_2 and select new then select web service

Give

Name to web service as myws

Give package name as ws

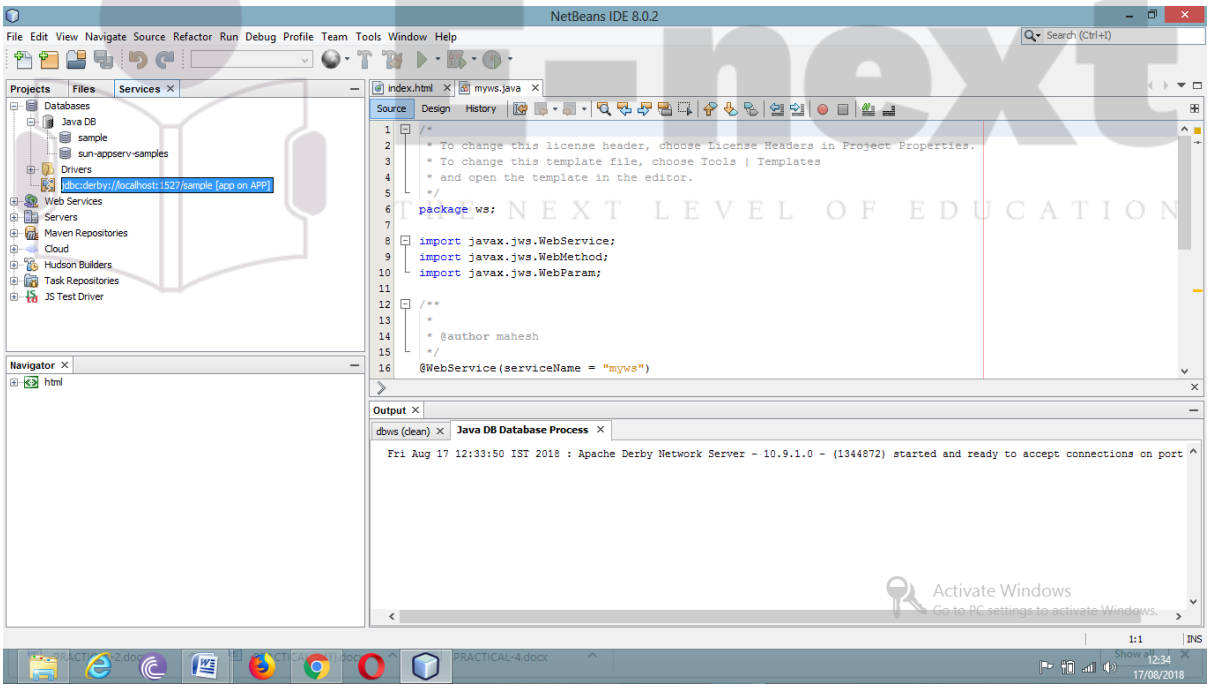
Press Finish

Go to service

Select java DB

Right click on java DB select Start server

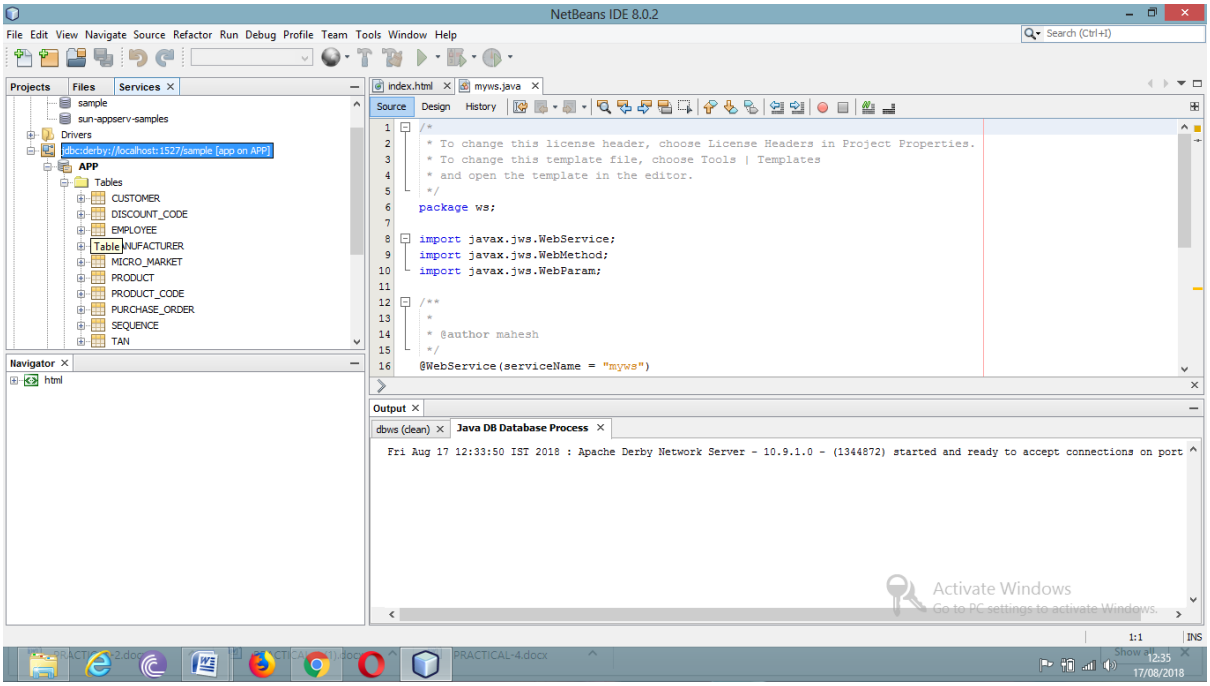
Right click on connection string and select connect



Expand connection string

It will show APP

Expand APP it will show table. Right click on table and select create table.



Give name to table as tan

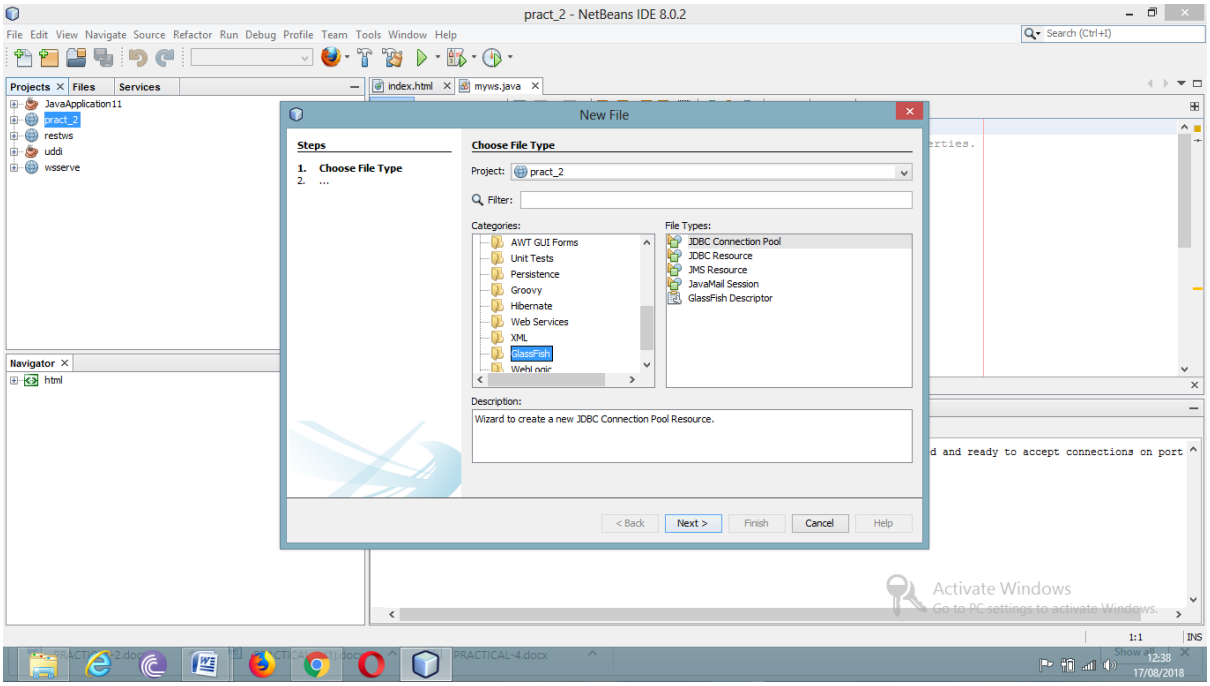
Then select add column

I have given name to table as Tan with column as rn and name

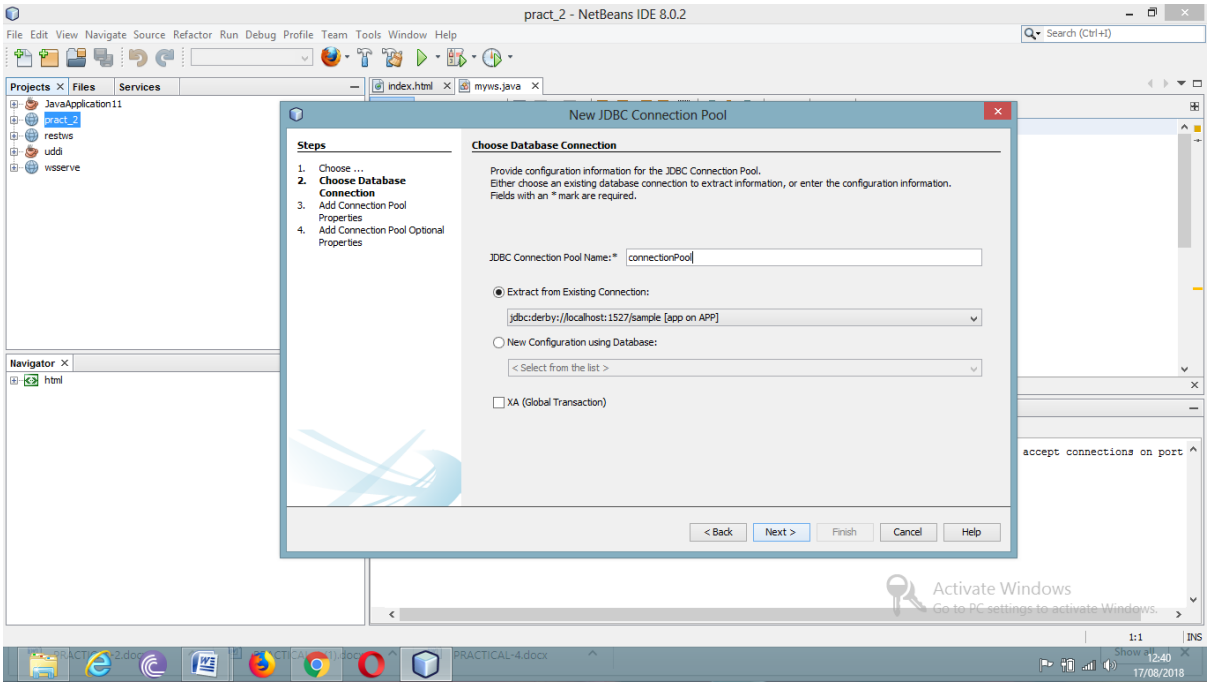
Come back to project

Right click Project name pract_2 select new select others in it select glassfish server

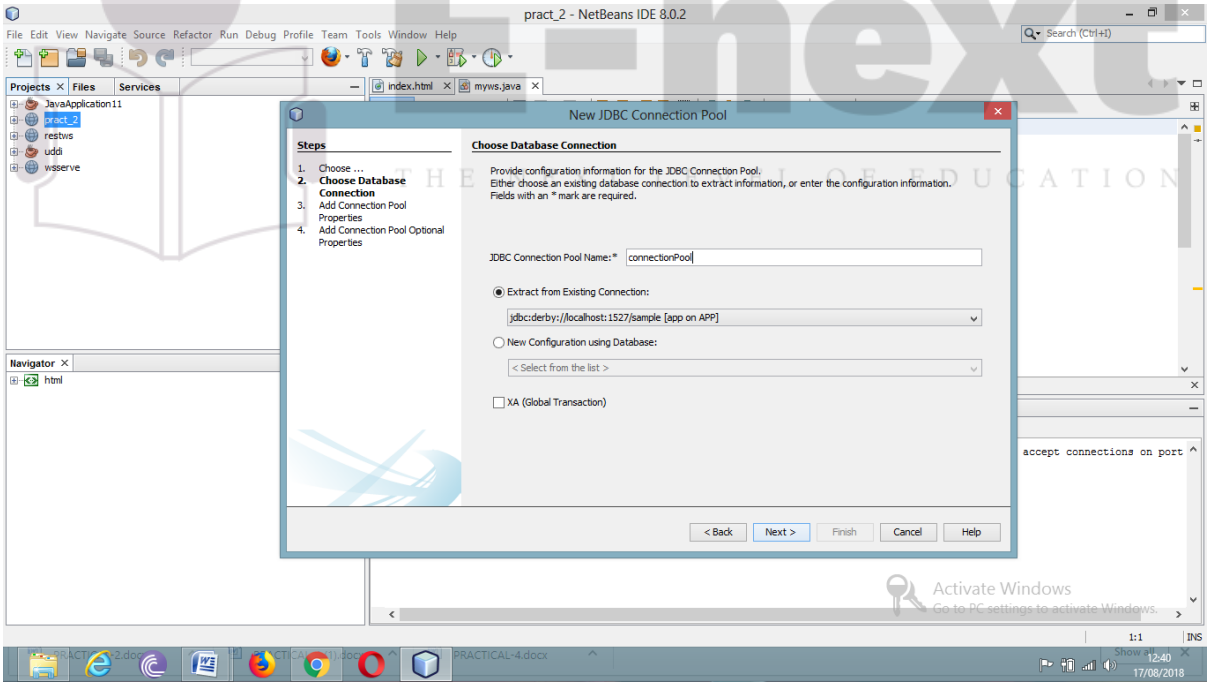
In that select jdbc connection pool



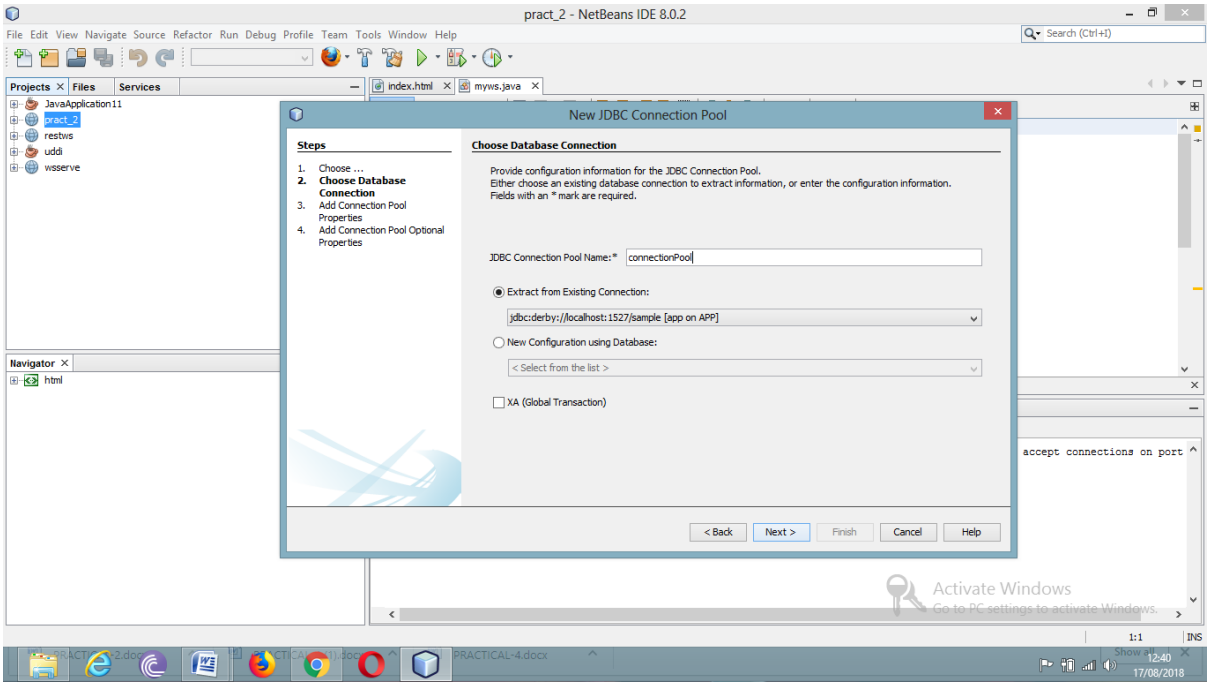
Press next



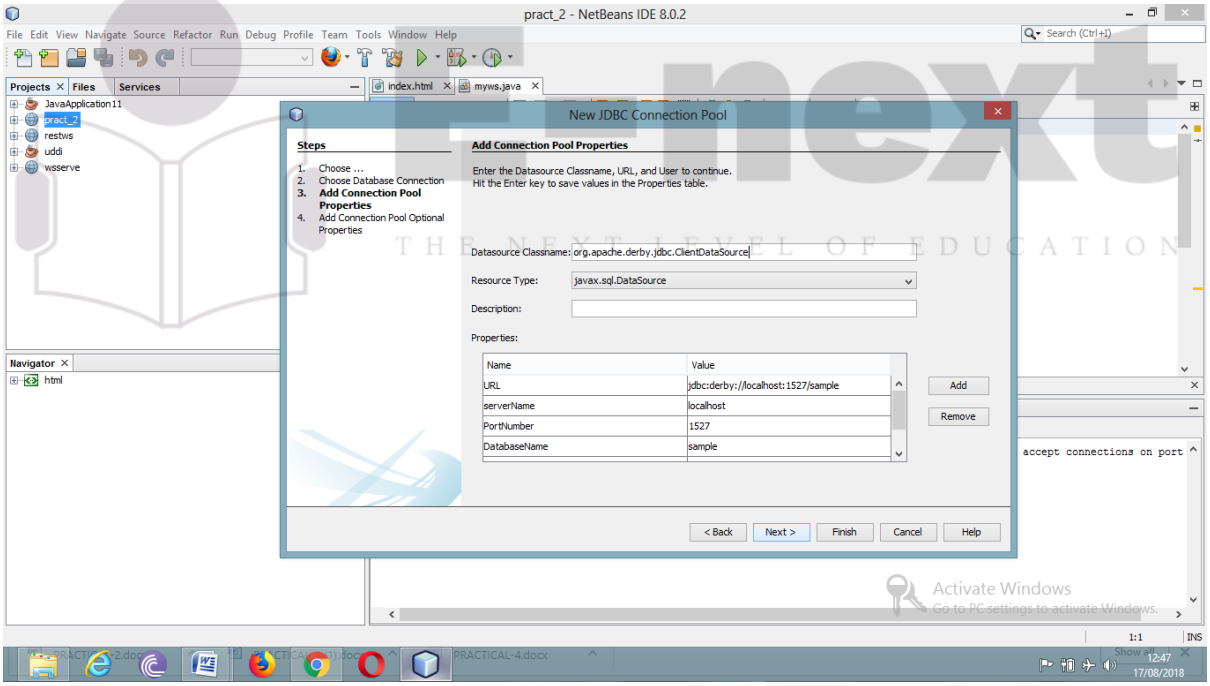
Press next



Press next



Press next

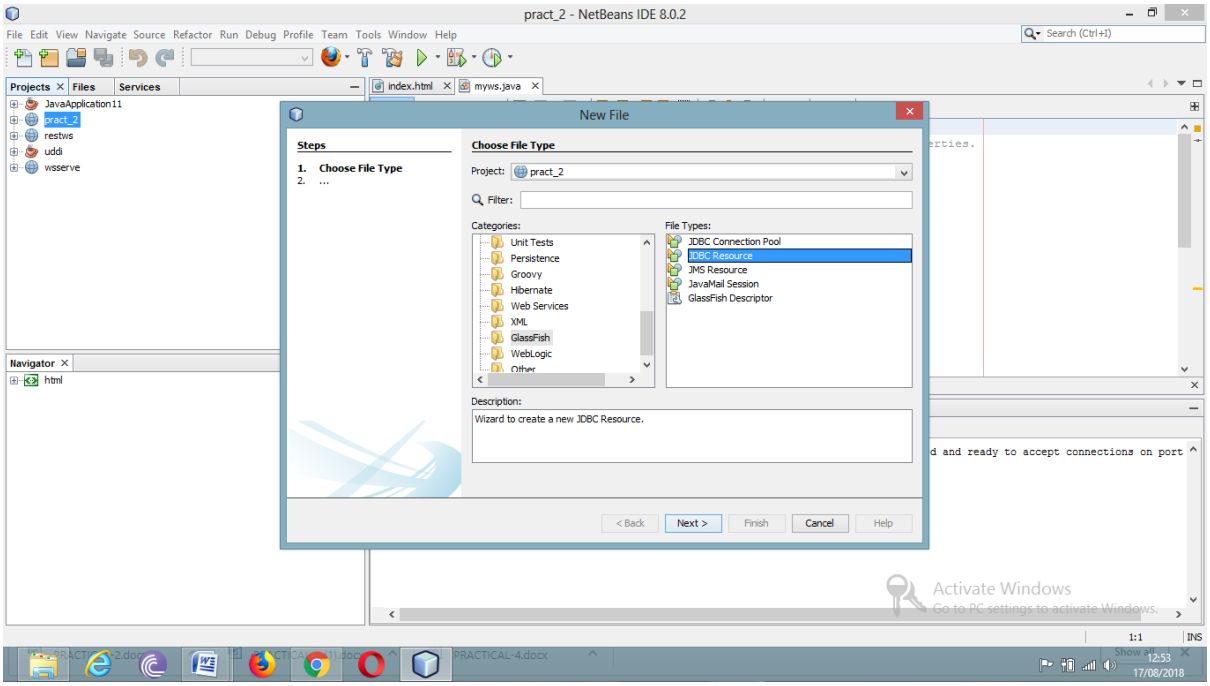


Press next

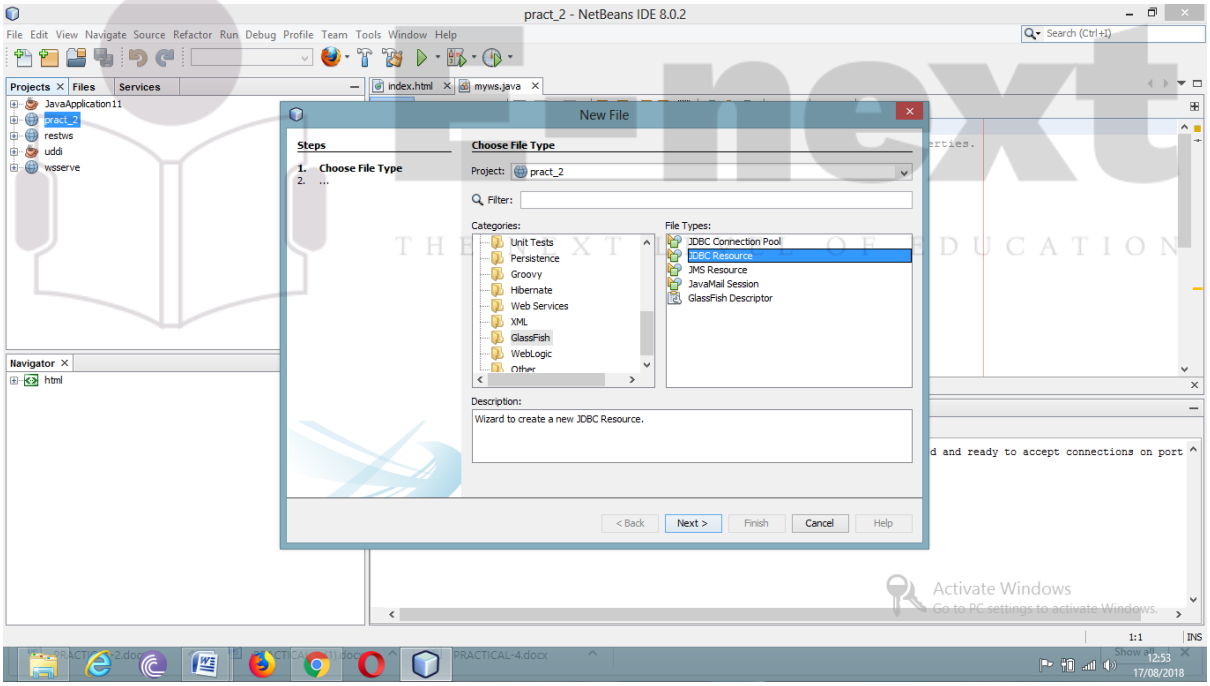
Press finish

Right click project name select new select other

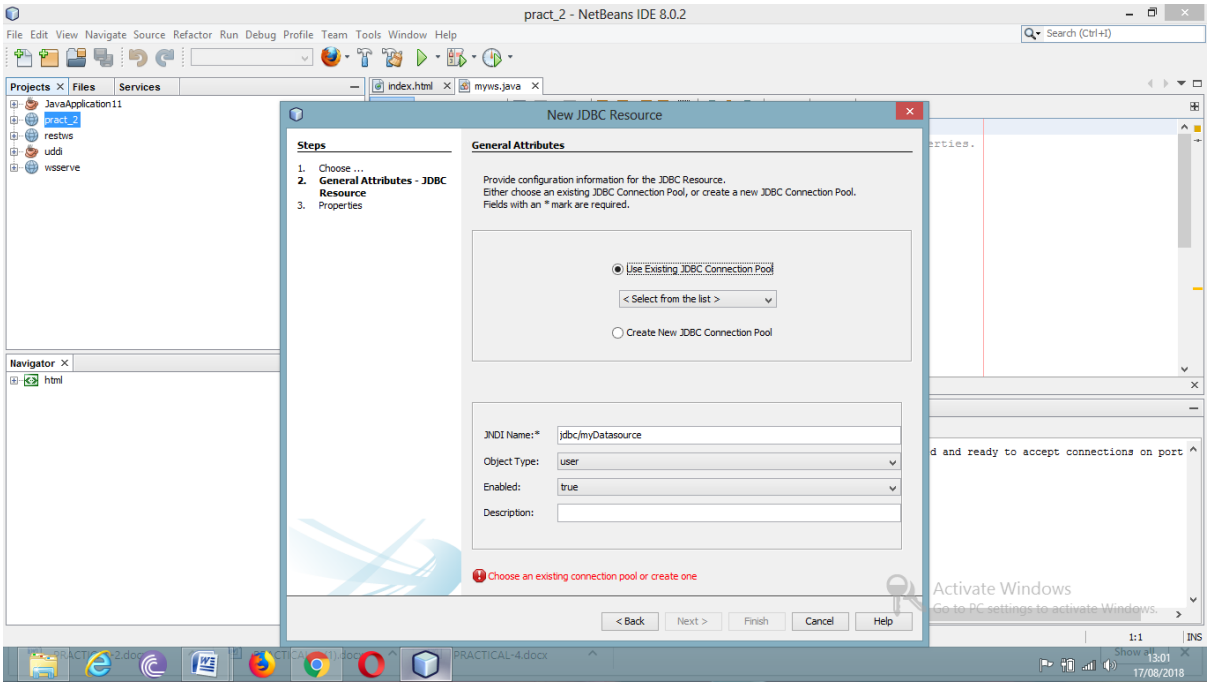
Select glassfish Select jdbc resource



Press next

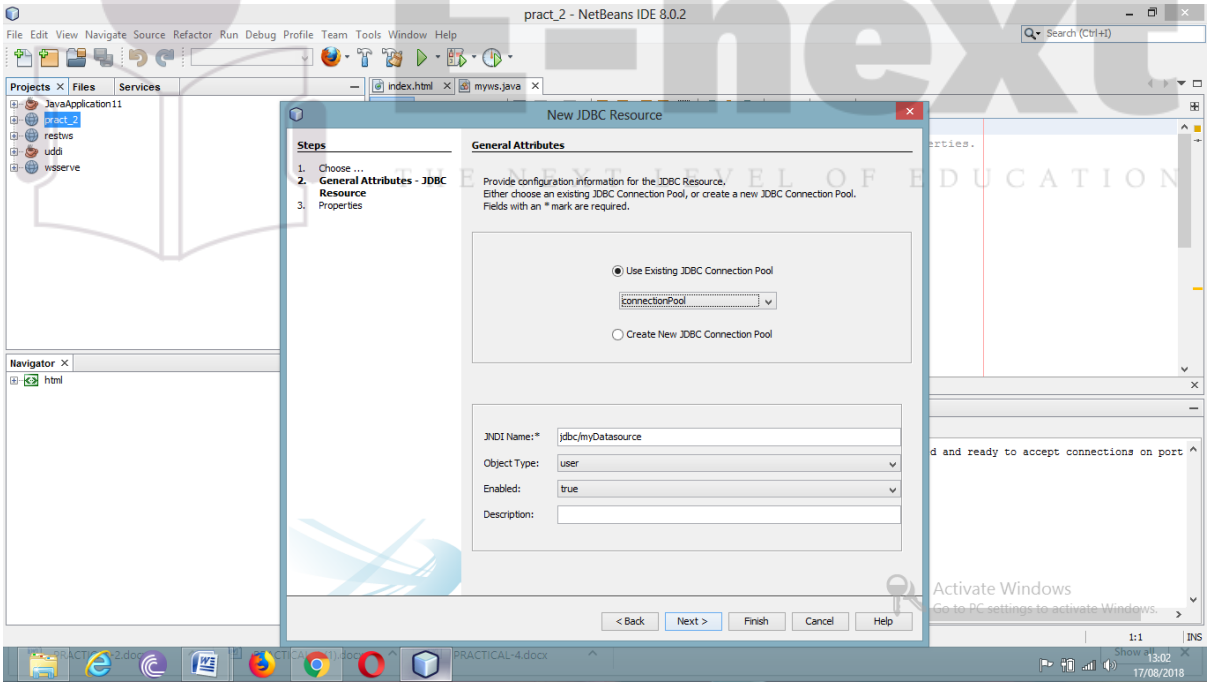


Press next



Click on existing connection pool

Select from combobox connection pool



Change JNDI name to jdbc/mydb

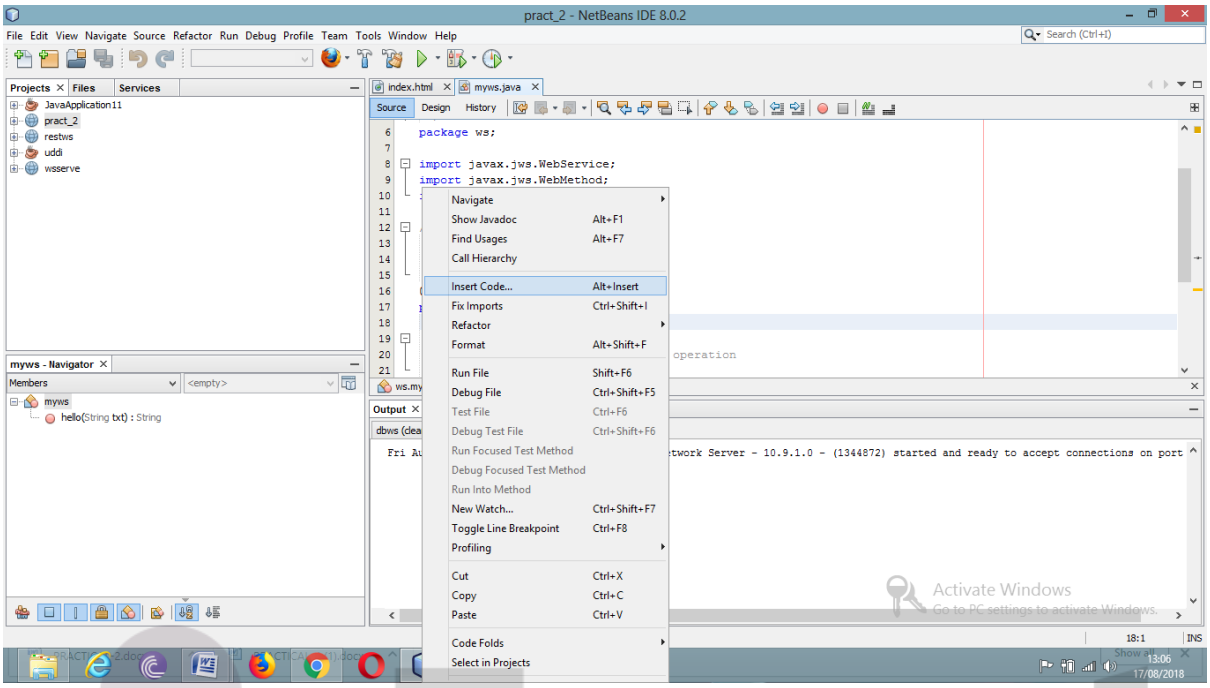
Press next

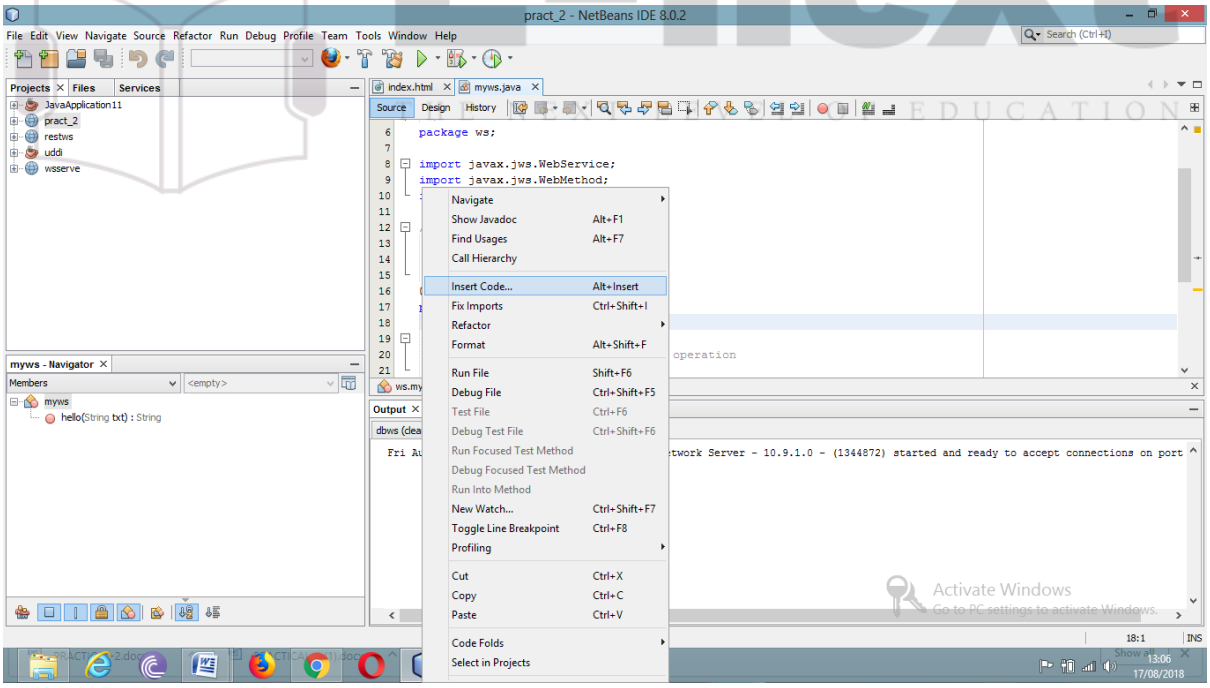
Press finish

It will open file myws.java

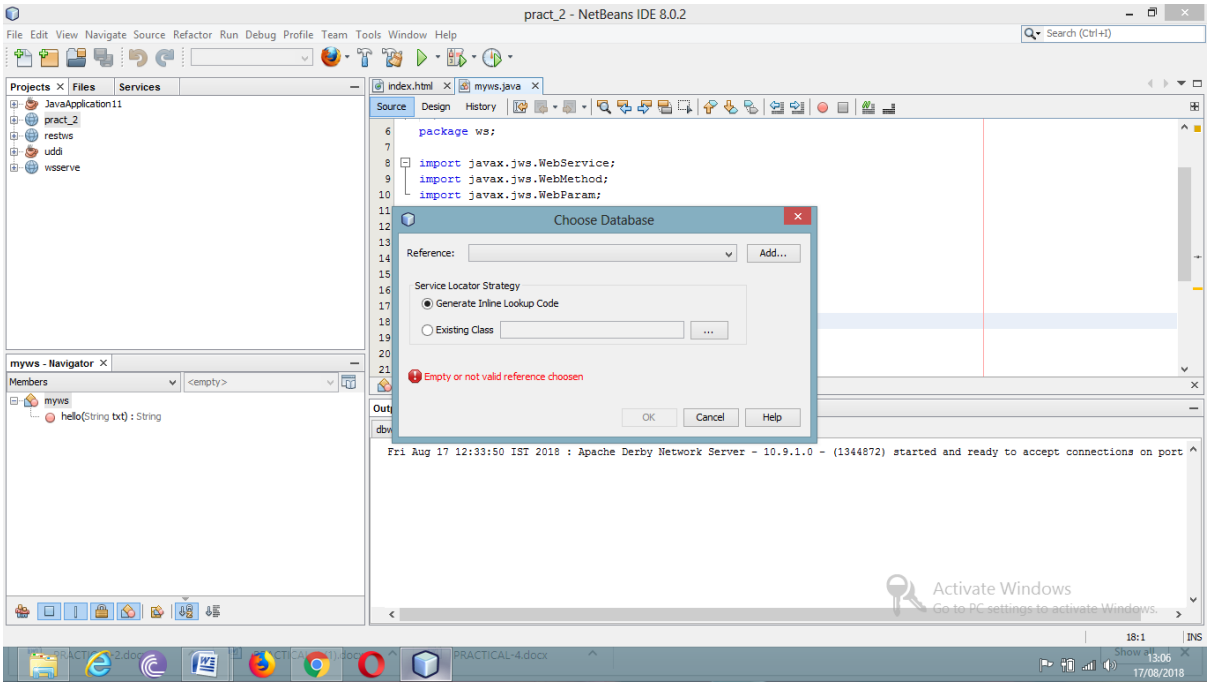
Right click on line number 18 (inside class myws)

Select insert code



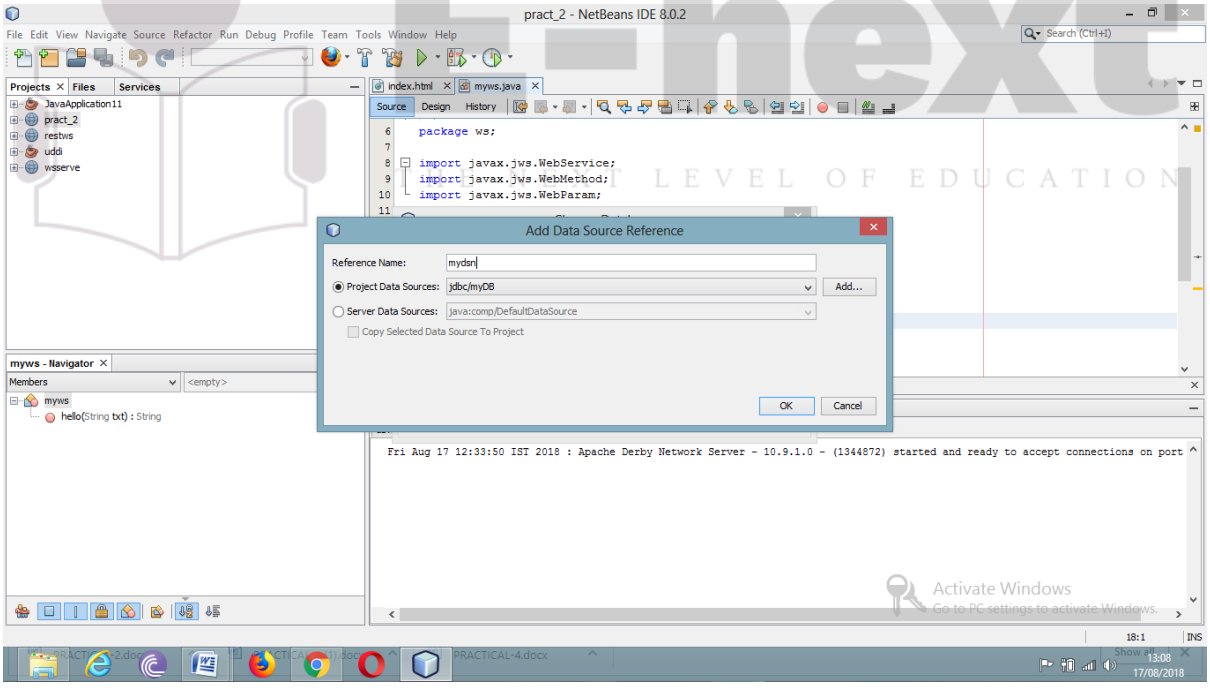


Select use database



Click on add

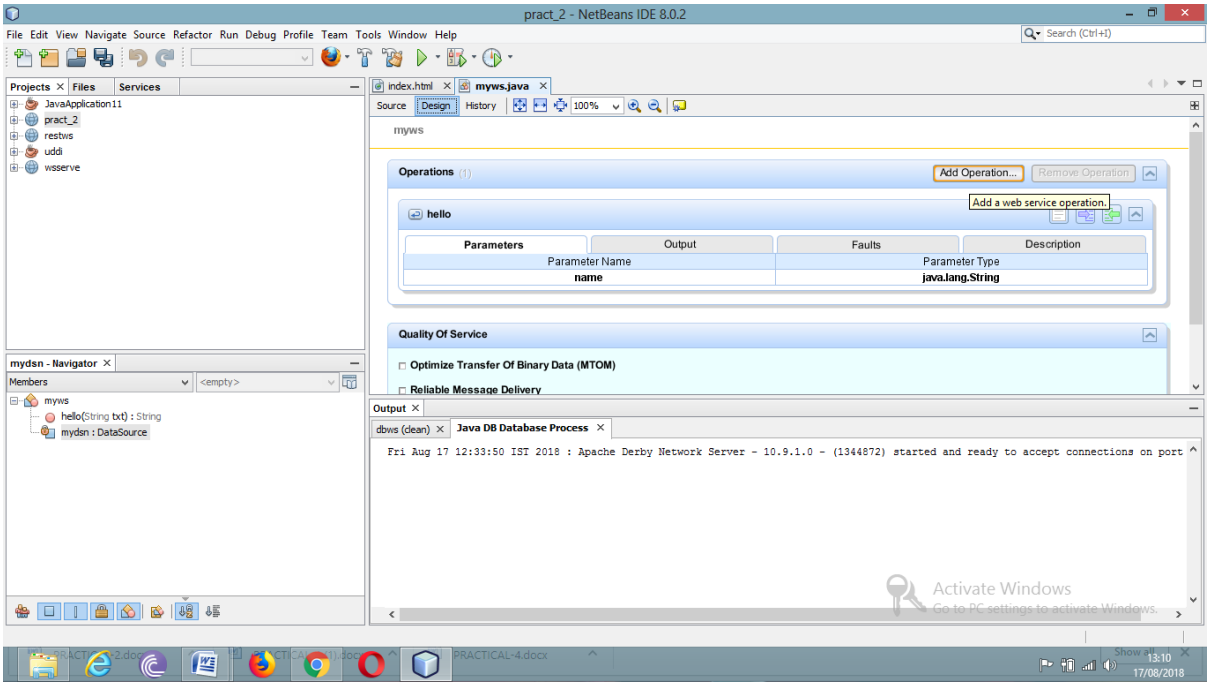
Give any name I have given my dsn



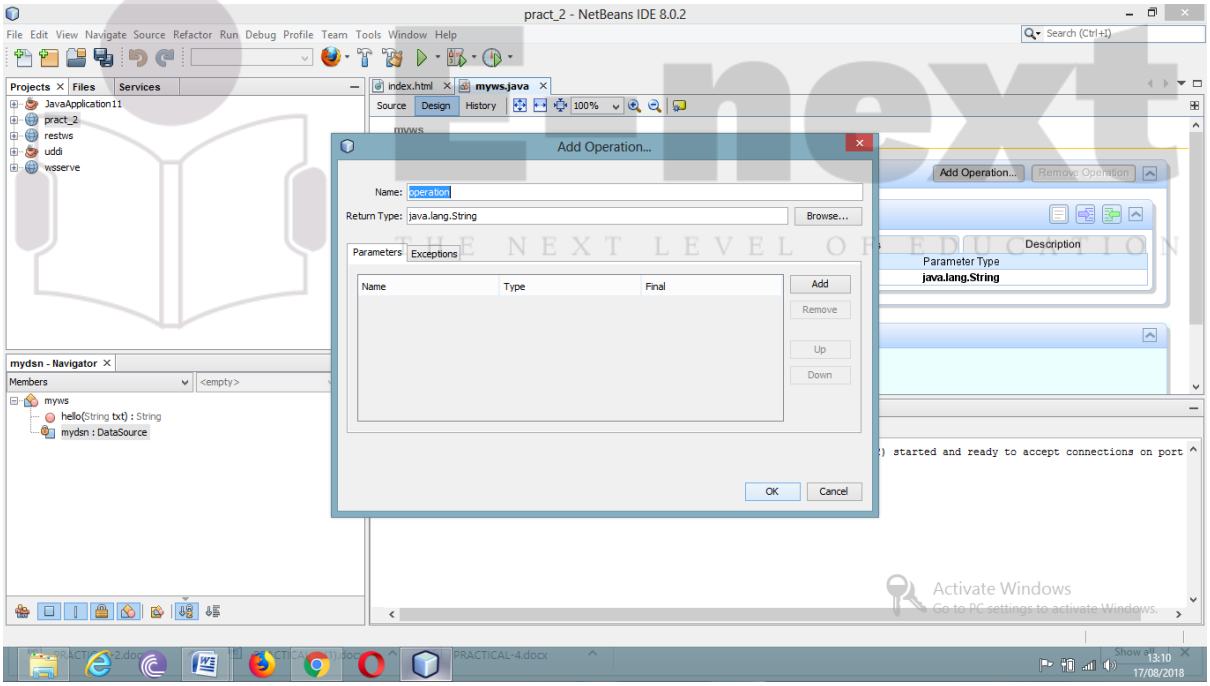
Press ok press ok

Now click on design

Select add operation



It will open



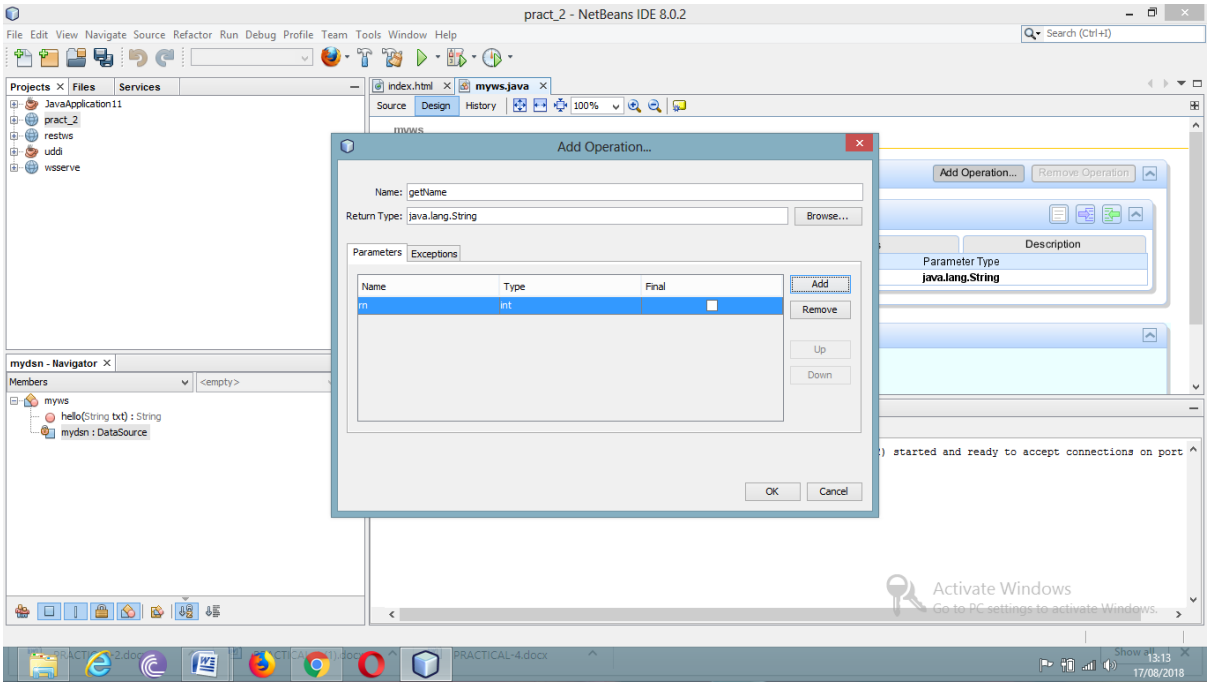
We want a two way web service which takes rn from user and return name

So give name as getName and return type is String

And click on add

And define parameter

My parameter name is rn and its type is int



Press ok

Very similar way define one way function which insert roll no and name in database Tan

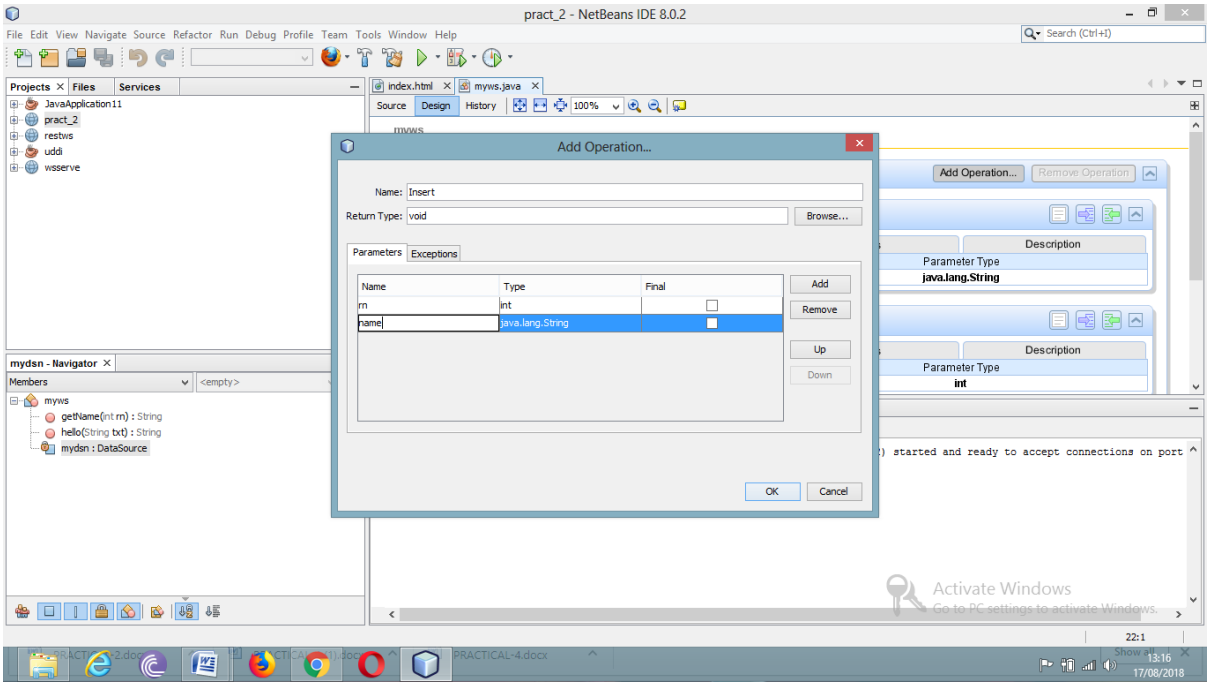
So again click on add operation give name as insert and return type as void

Click on add

Then give parameter as rn with type as int

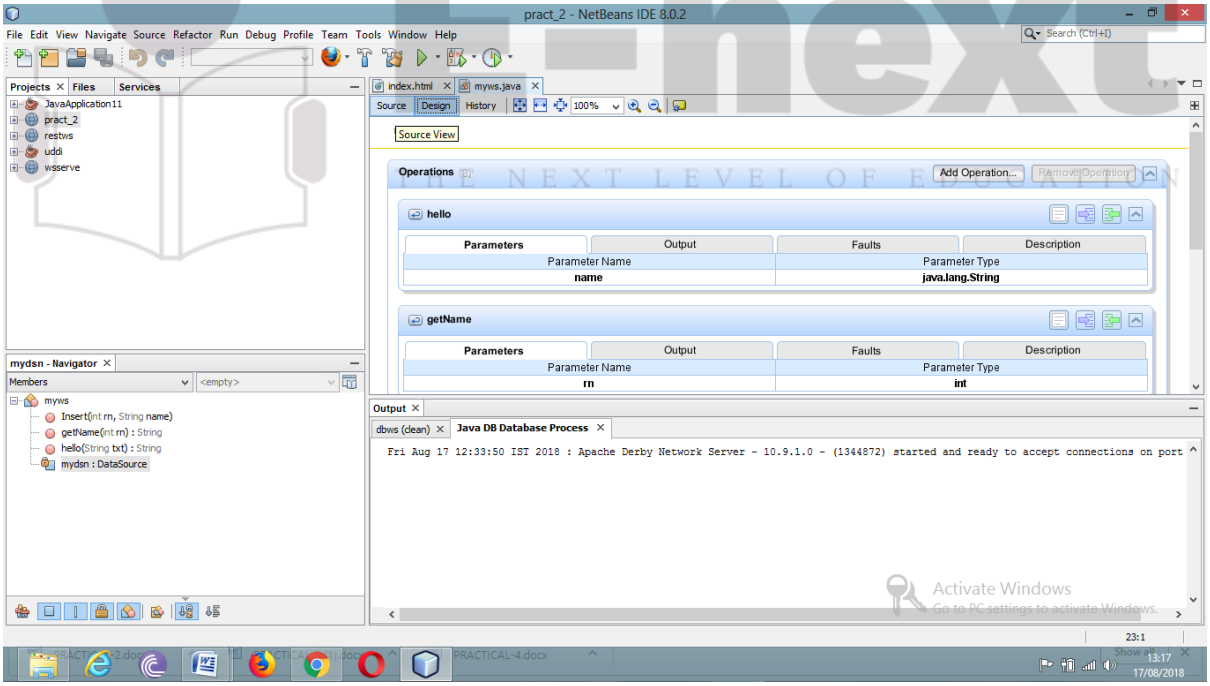
Again press add

Give parameter as name and type as string

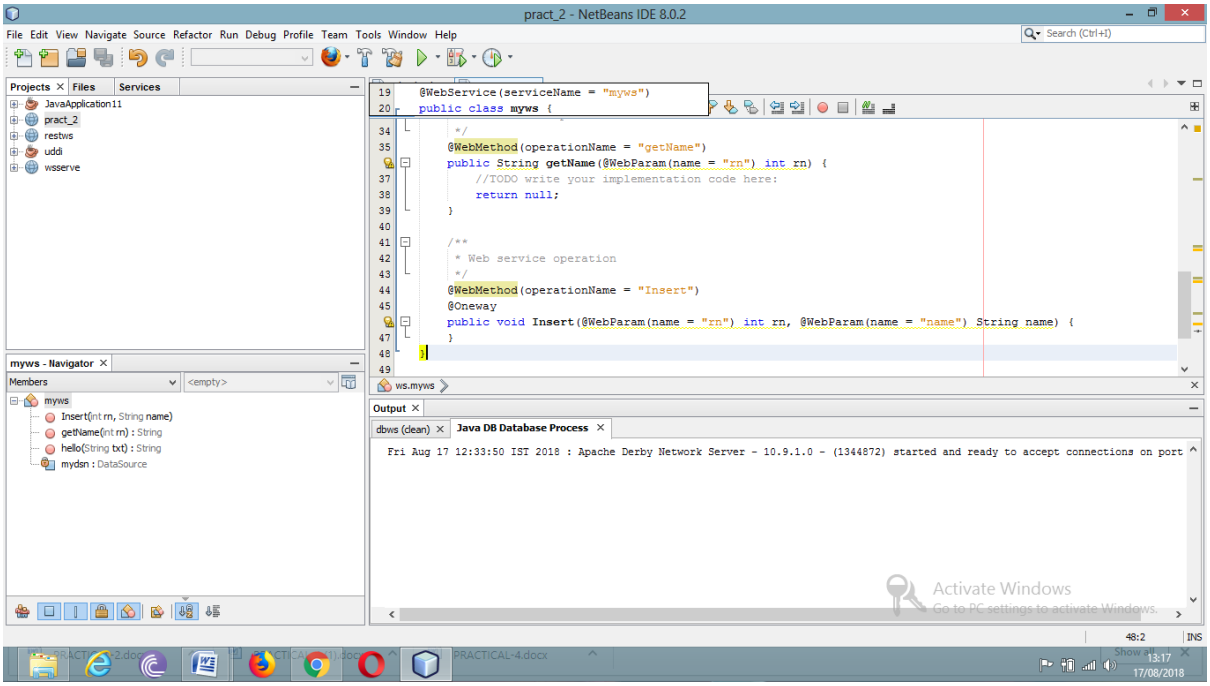


Press ok

Now click on source

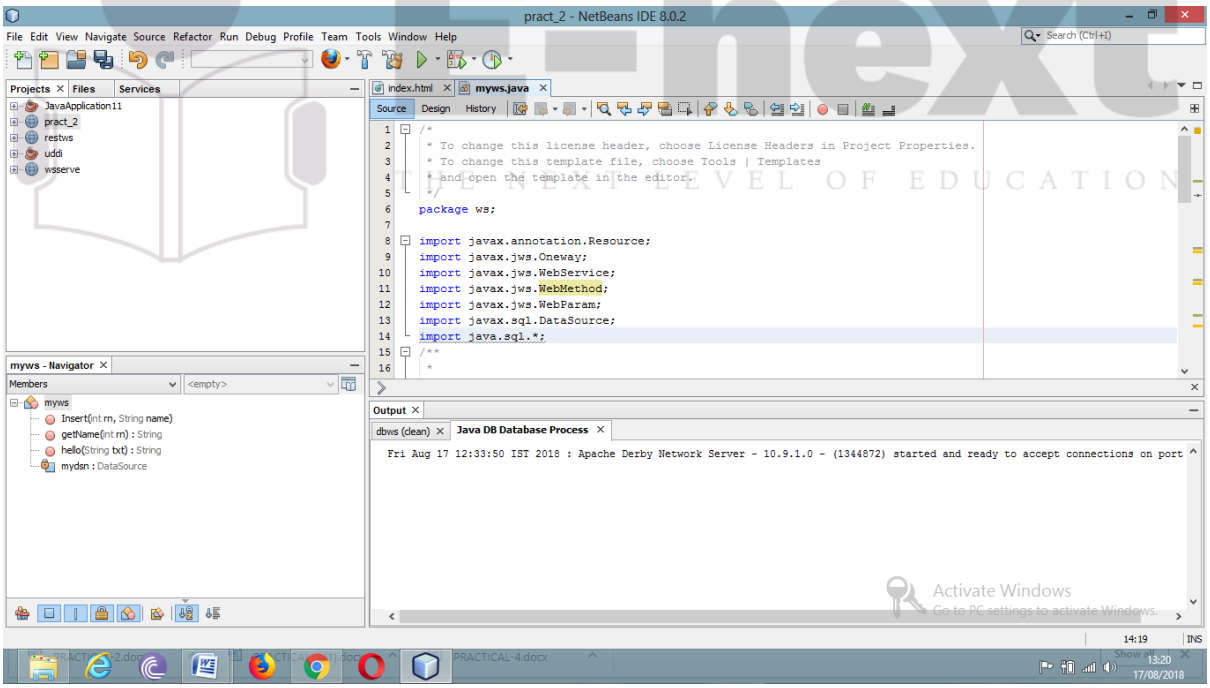


It will show you code (Functions you have created)



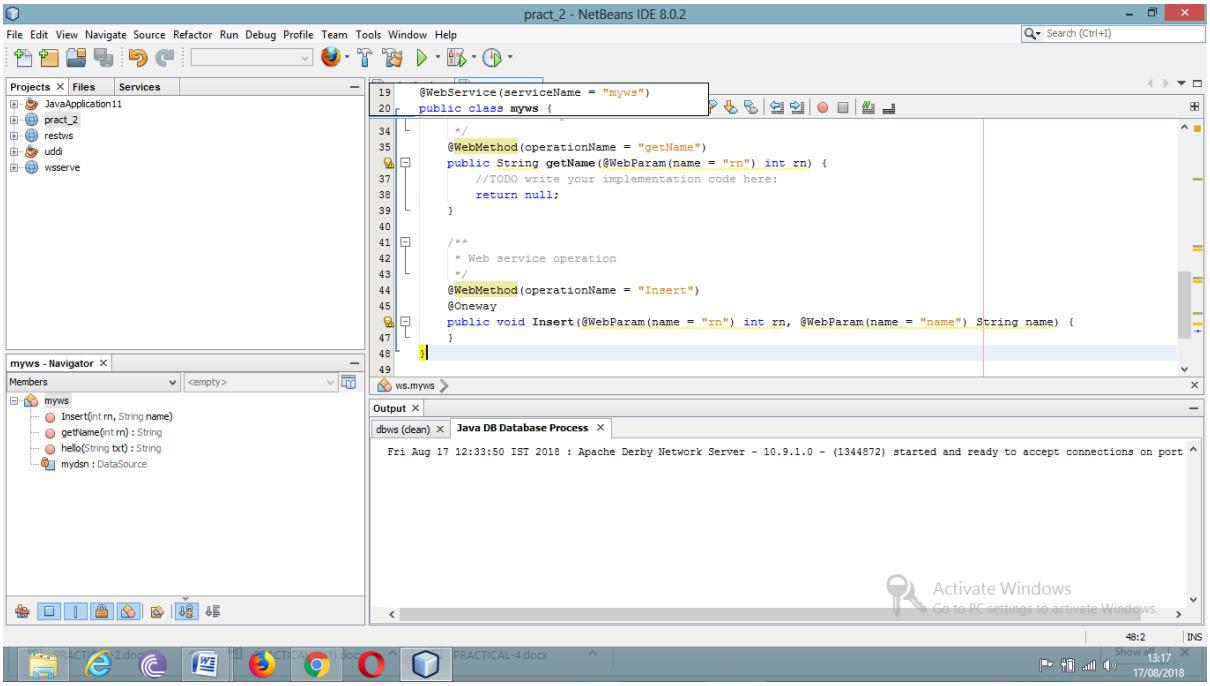
As we are working with database so we need to import java.sql.*;

So go on the top of same file and add import java.sql.*; statement



Now come to function getName()

Go inside it and remove the line “todo code here”



And type here

```
try{
        Connection c=mydsn.getConnection();
        PreparedStatement ps=c.prepareStatement("select * from tan where rn=?");
        ps.setInt(1, rn);
        ResultSet r=ps.executeQuery();
        if(r.next())
           return r.getString(2);
        else
           return "No name found";


        }catch(Exception e)
           {return "error";}
```

Now save it

Very similar way write the code for Insert Function (One way)

Right Click on Project name pract_2

Select deploy

Once Deployed our web service is ready

Now We will Create Client (Servlet)

So create new Project (web java)

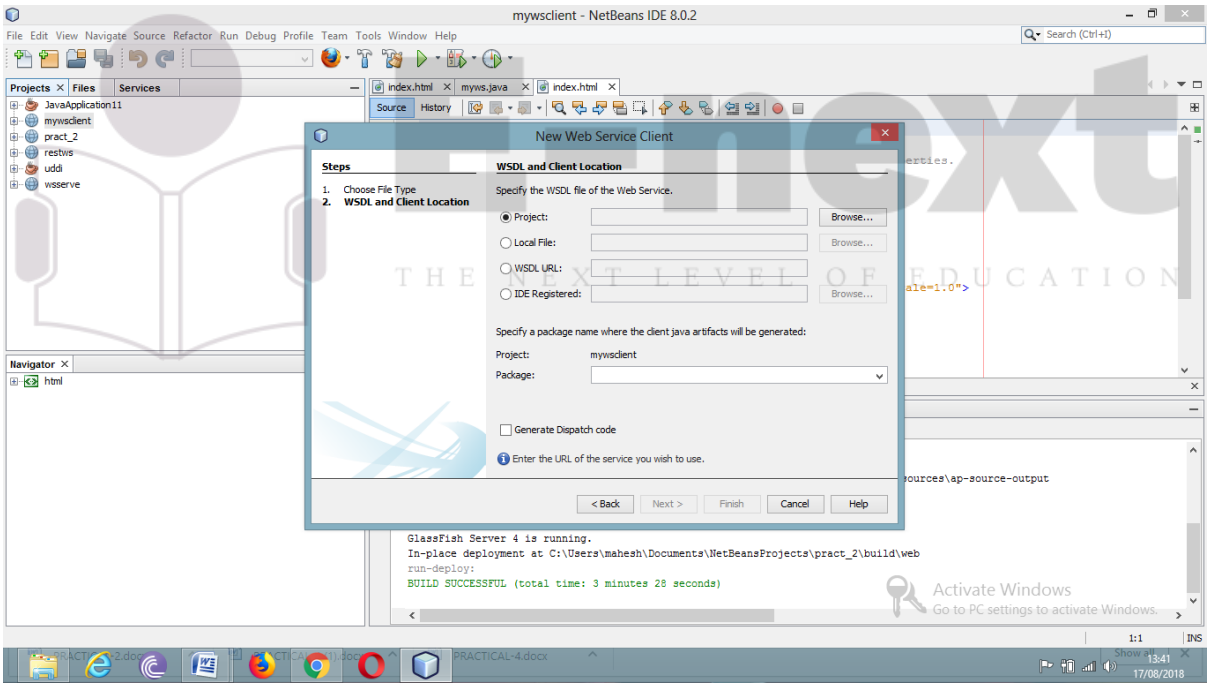
Select new project select new select web java

Give it name mywsclient Press next Press finish

Now right click on project name mywsclient

select new

select web service client

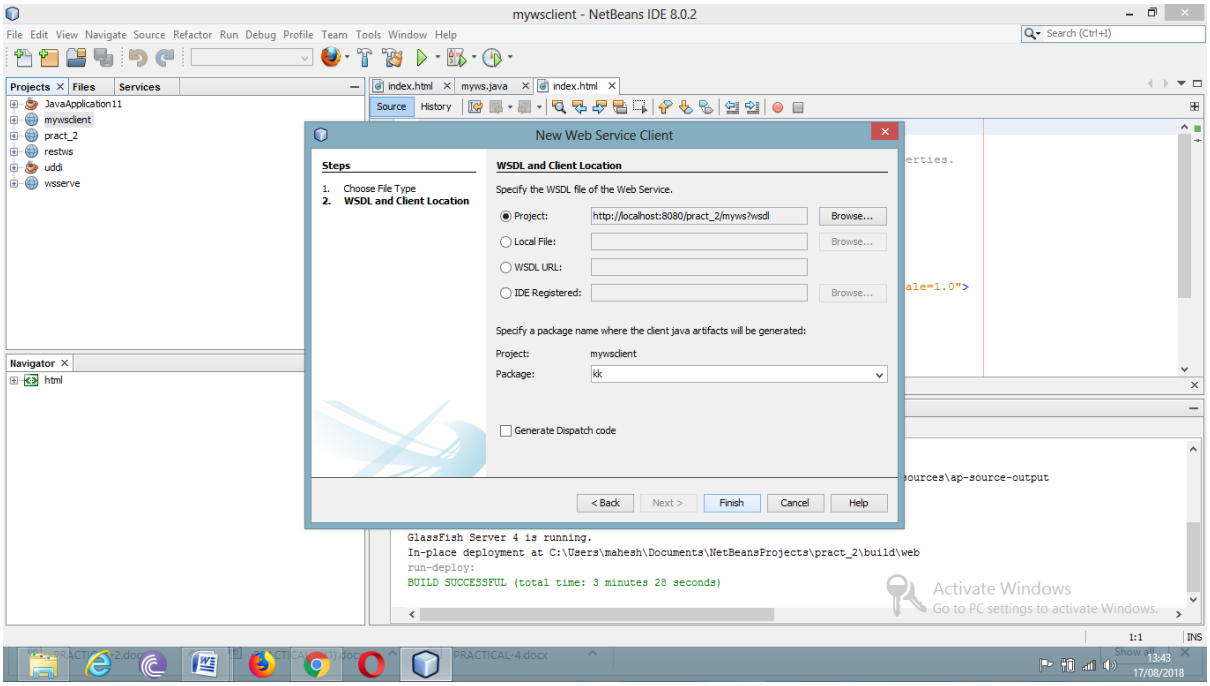


click on Project then select brows

expand pract_2

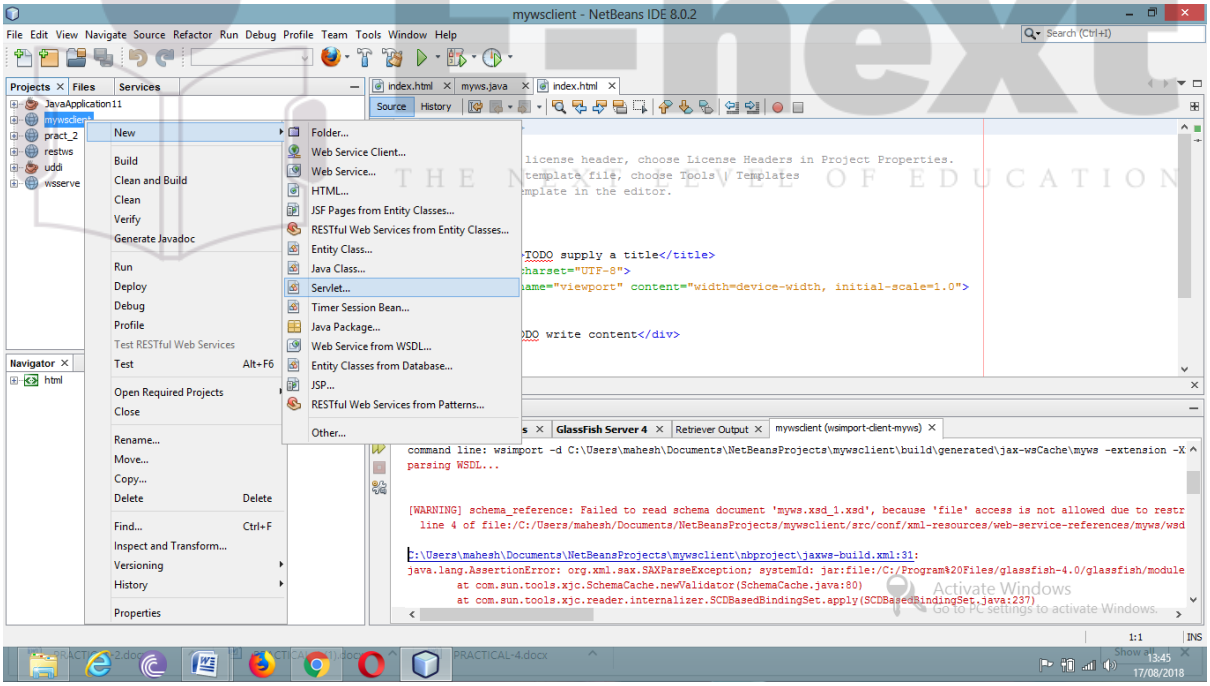
select myws and press ok

Give package name kk



Press finish

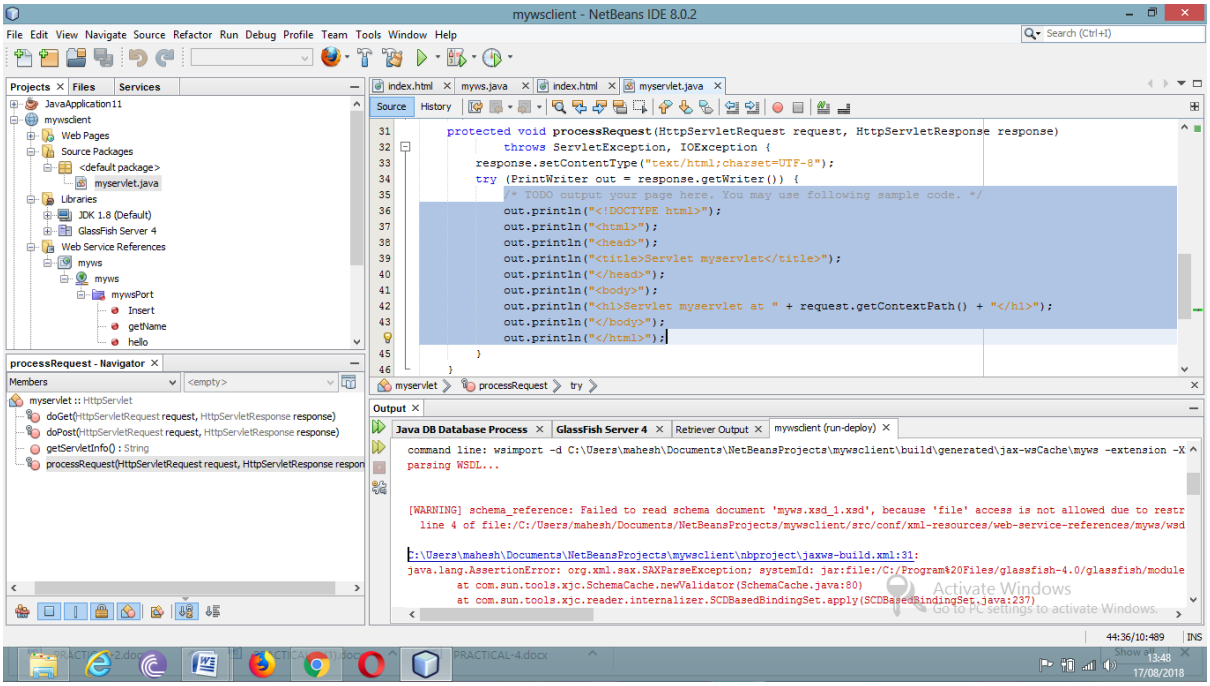
Now right click on mywsclient project select new select servlet



Give servlet name as myservlet

Press finish

Now remove lines from number 34 to 42 of Process request function of myservlet



Now the lines which we have removed i.e. from 34 to 42 ,type there

out.println(getName(1));

Now on the project window

Expand web service reference folder

You will find there getName function

Simply drag it in side myservlet.

It will create code automatically.

Now deploy mywsclient and run mywsclient by right clicking on the code of mywsclient

PRACTICAL-4

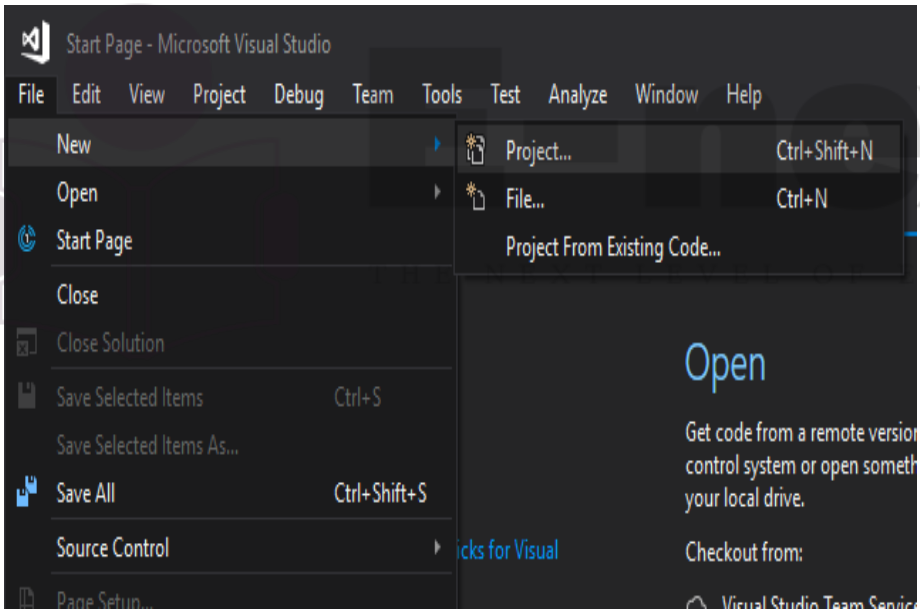
Aim: Develop client which consumes web services developed in different platform.

Requirement:

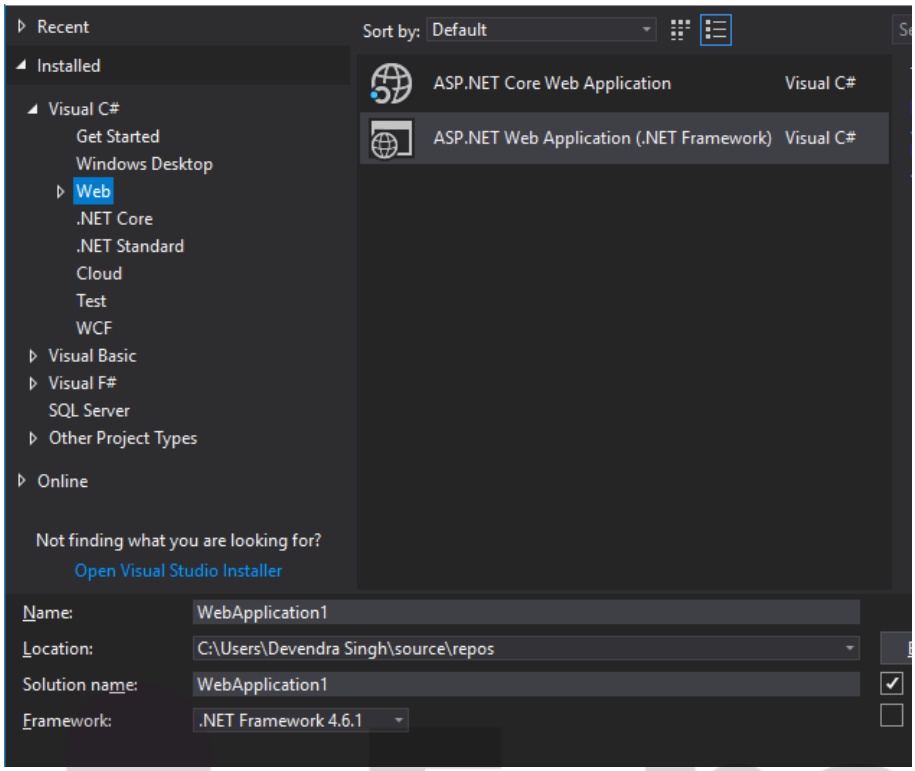
1. Visual Studio Community 2017
2. Version : 15.8 or latest

In this practical we are creating Web Service in Visual Studio ant then we will consume it in NetBeans.

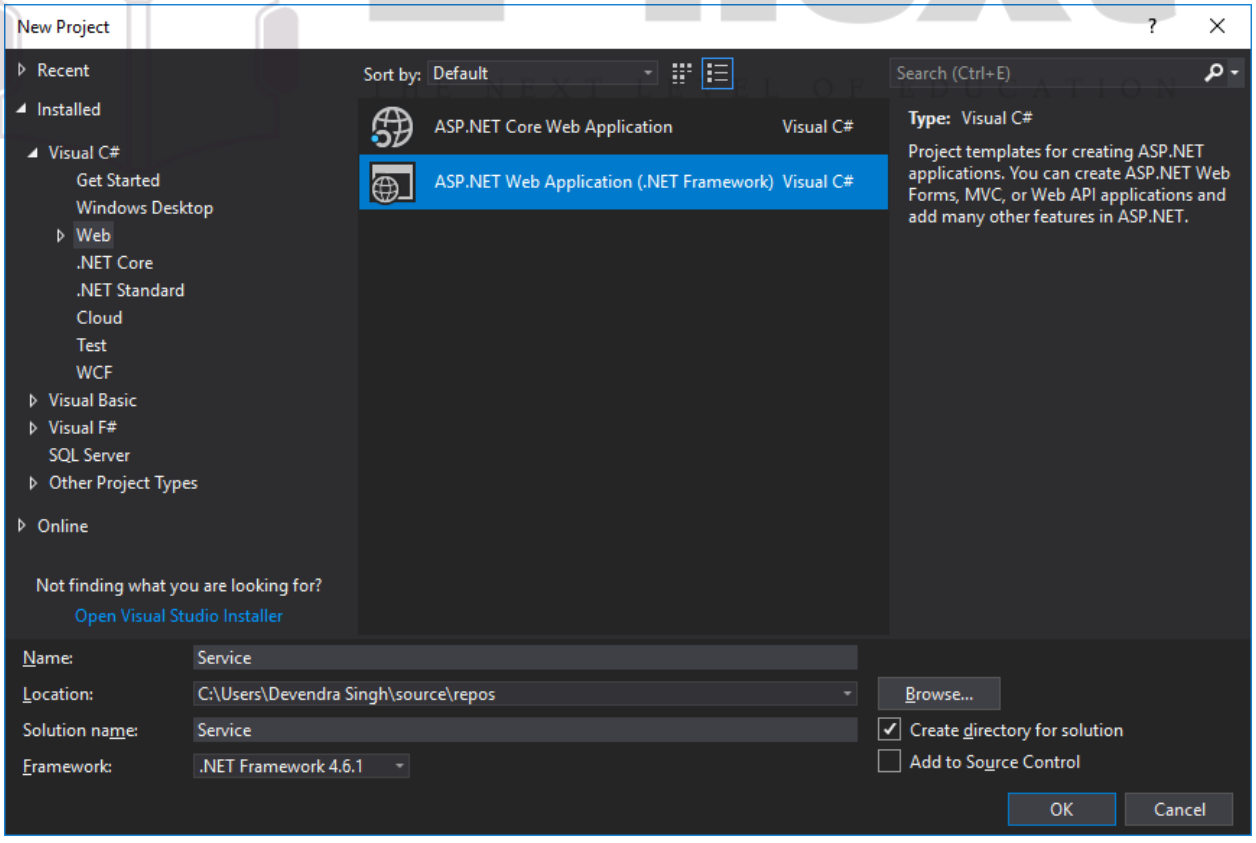
1. **Open Visual Studio IDE** and **click on File**.
   File -> New -> Project



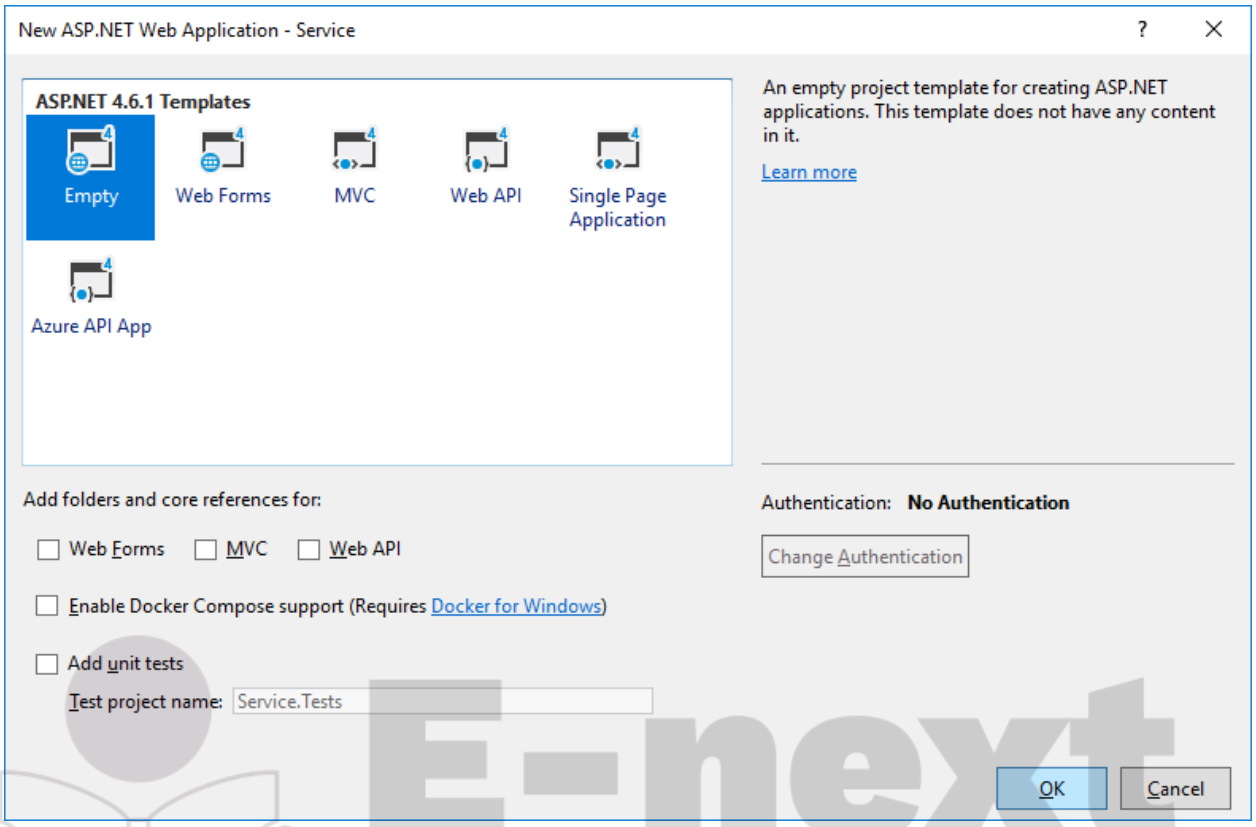
2. **Click on Web**.



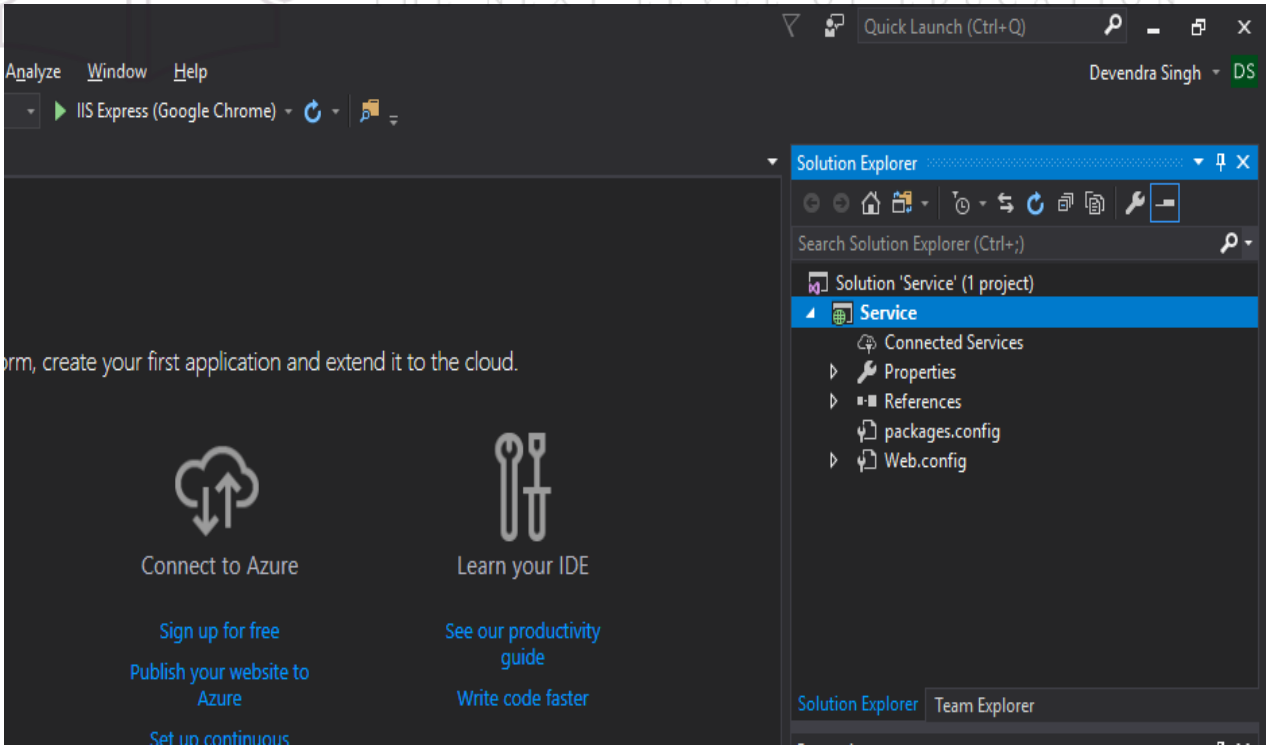
3. **Select ASP.NET Web Application** and give **Name** as **Service**. After that **click on OK** button.



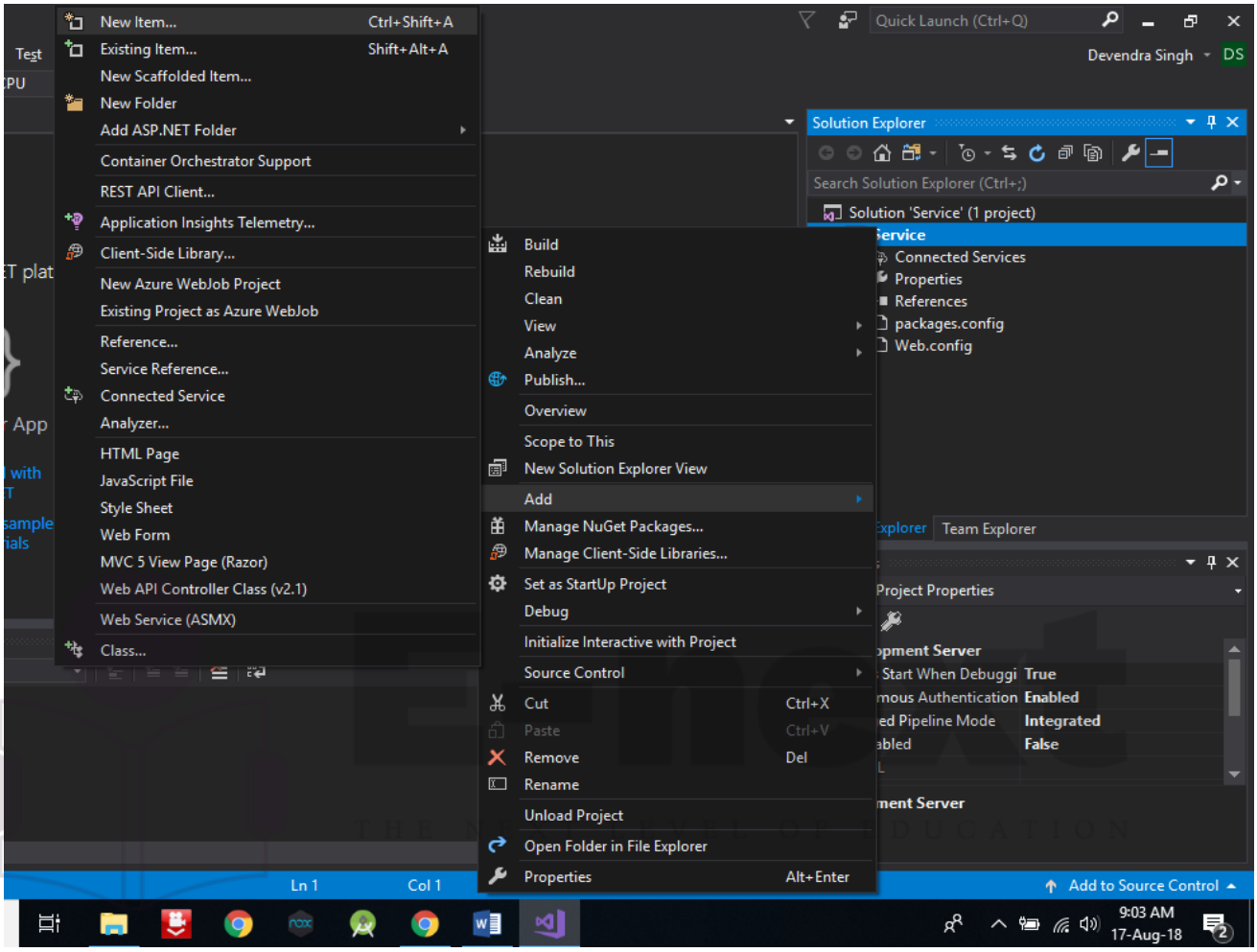
4. **Select Empty** and **click on OK** button.



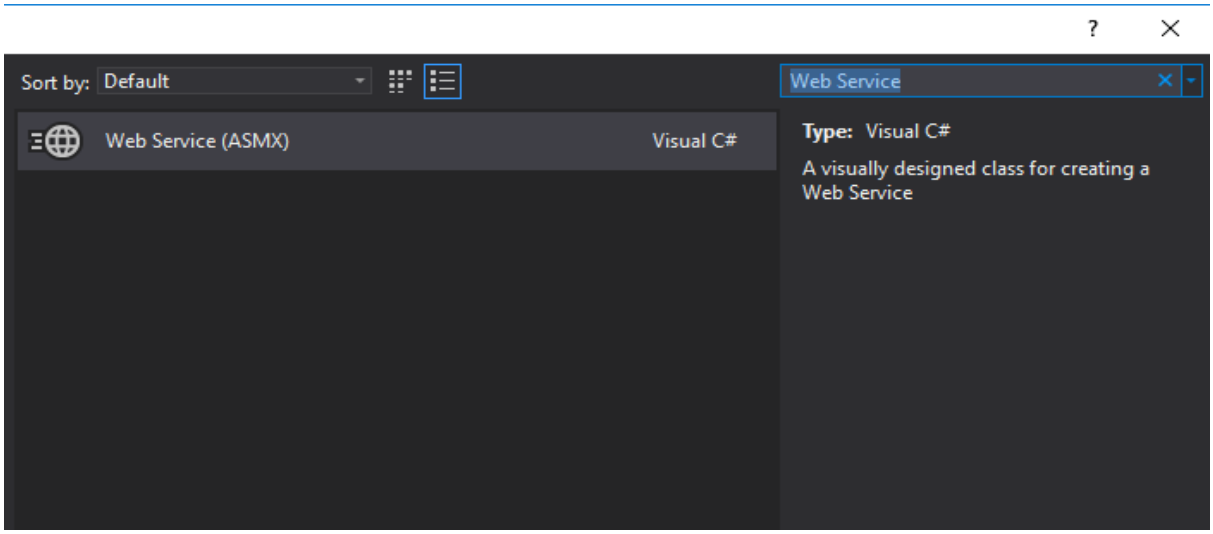
5. Now you can see, on right side in Solution Explorer Service project is created.



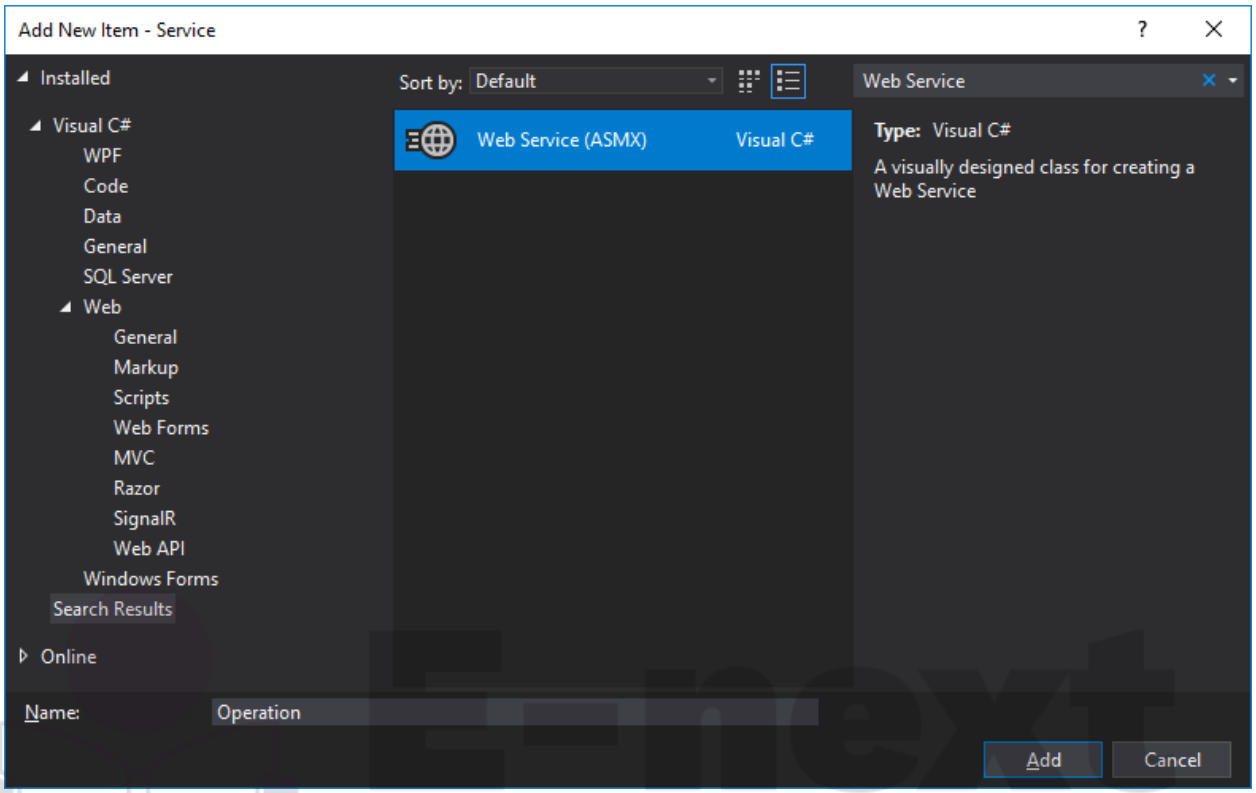
6. Right click on Service -> Add -> New Item.



7. Search for Web Service.

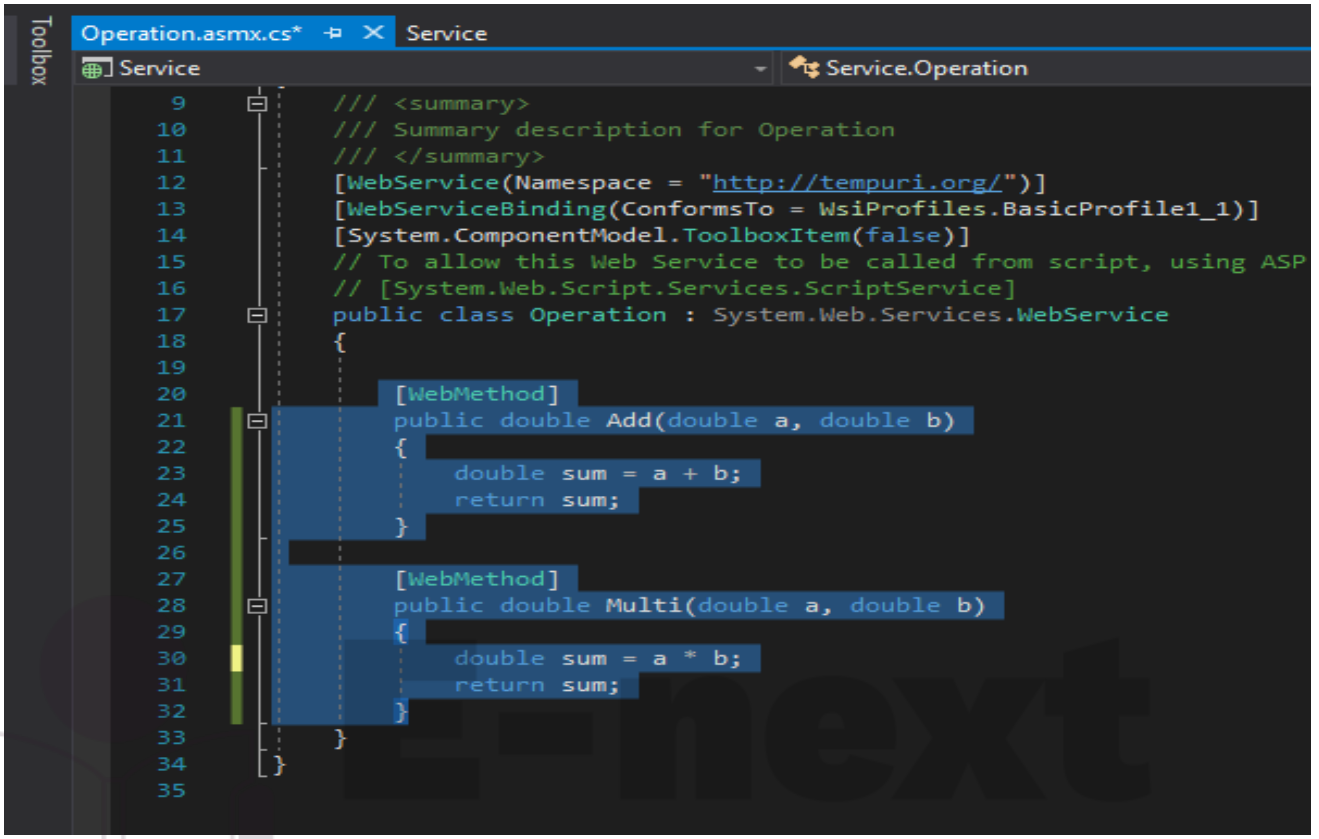


8. **Select Web Service** and **give Name** as **Operation**. After then **click on Add** button.



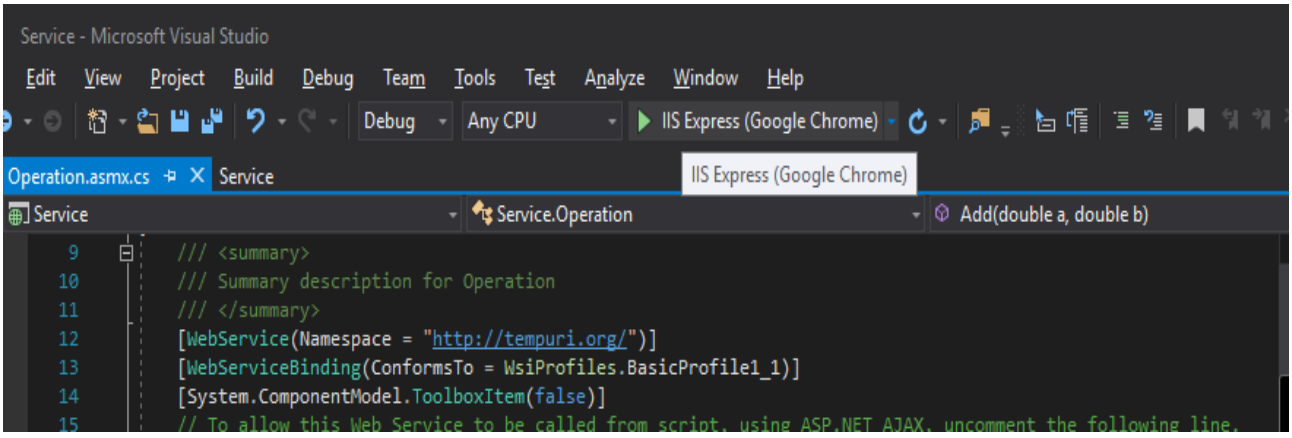
9. After click on Add button, Operaion.asmx.cs file will be automatically open otherwise open it from Solution Explorer and **Add the following code into Class Operation**.

```
[WebMethod]
 public double Add(double a, double b)
 {
    double sum = a + b;
    return sum;
 }

 [WebMethod]
 public double Multi(double a, double b)
 {
    double sum = a * b;
    return s
}
```

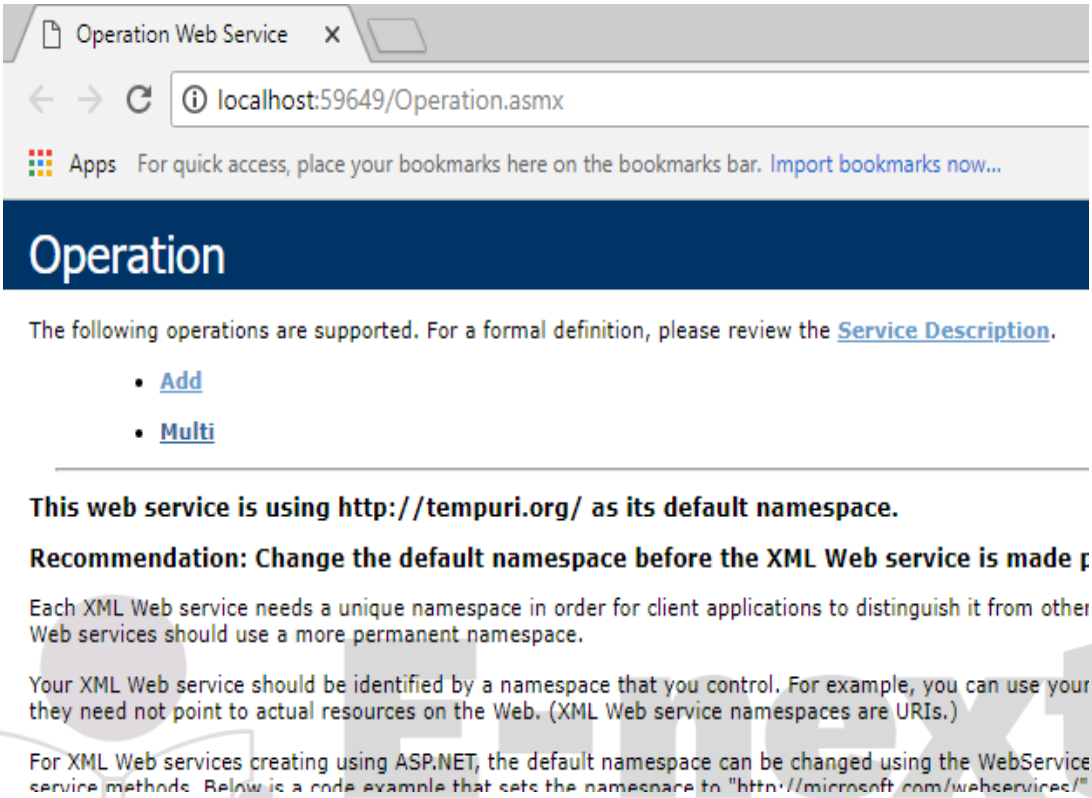


After that press Ctrl+S to save the methods. Actually we are creating two methods for Web Service. One is for addition of two numbers and second one is for multiplication of two numbers.

10. Now **run the project** by **click on Green arrow button below the Window menu**.

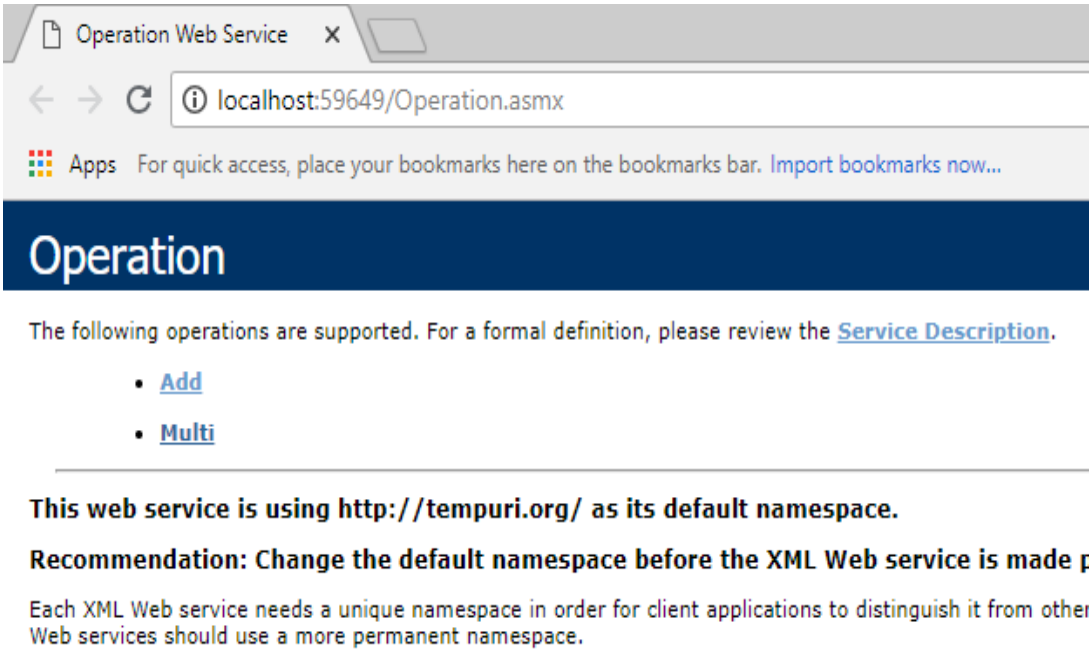


11. A window will open in browser and that is our web service.

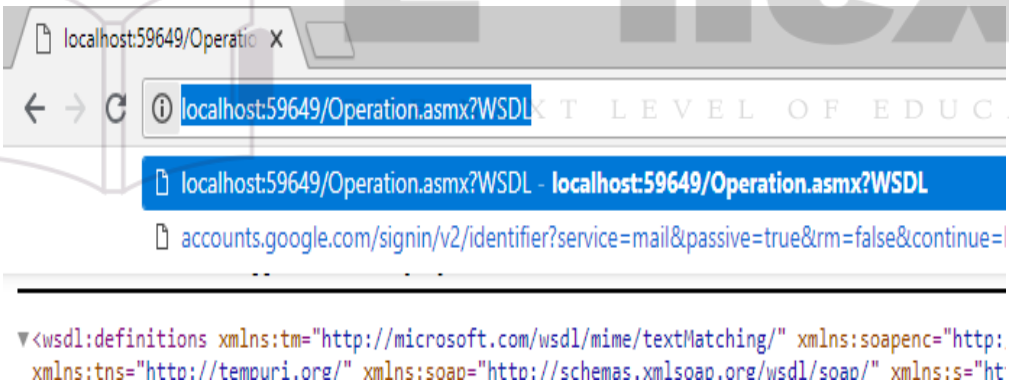


12. You can check services by click on Add or Multi option. But we don't need this.

13. Now **click on Service Description** option.



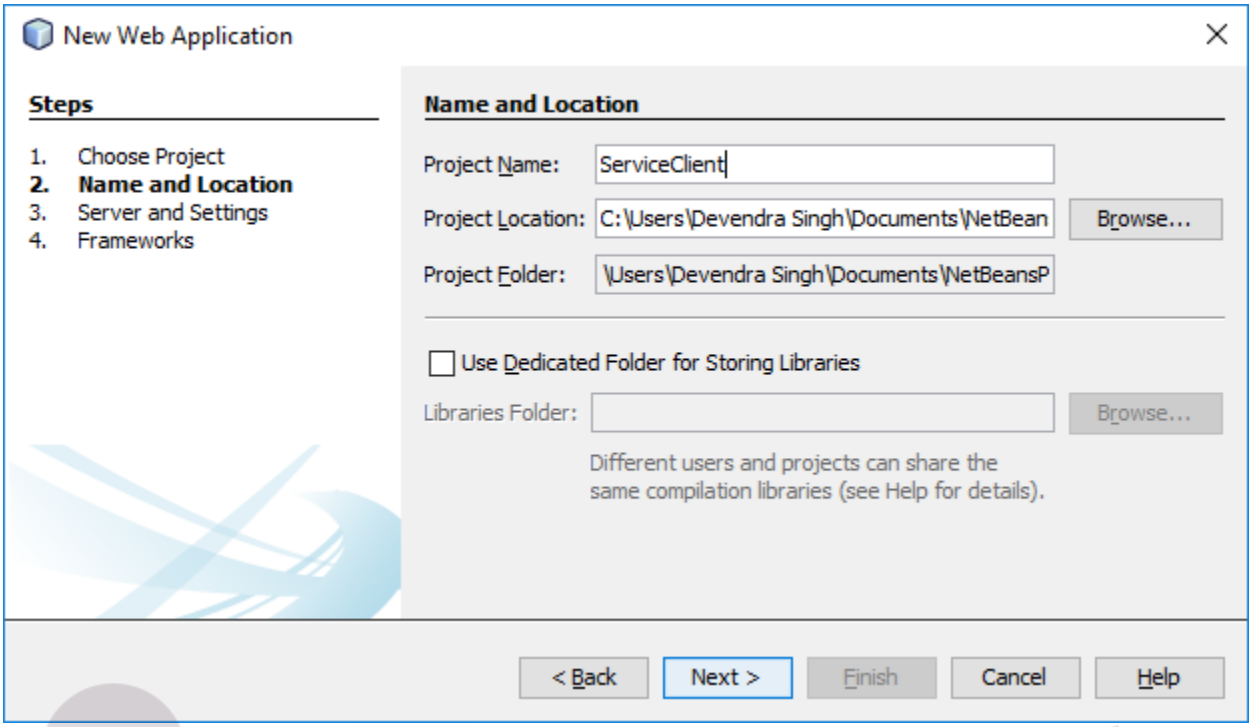
14. Now a new window will open. **Select the link and copy it**. **We will use this link in NetBeans to consume these services**.



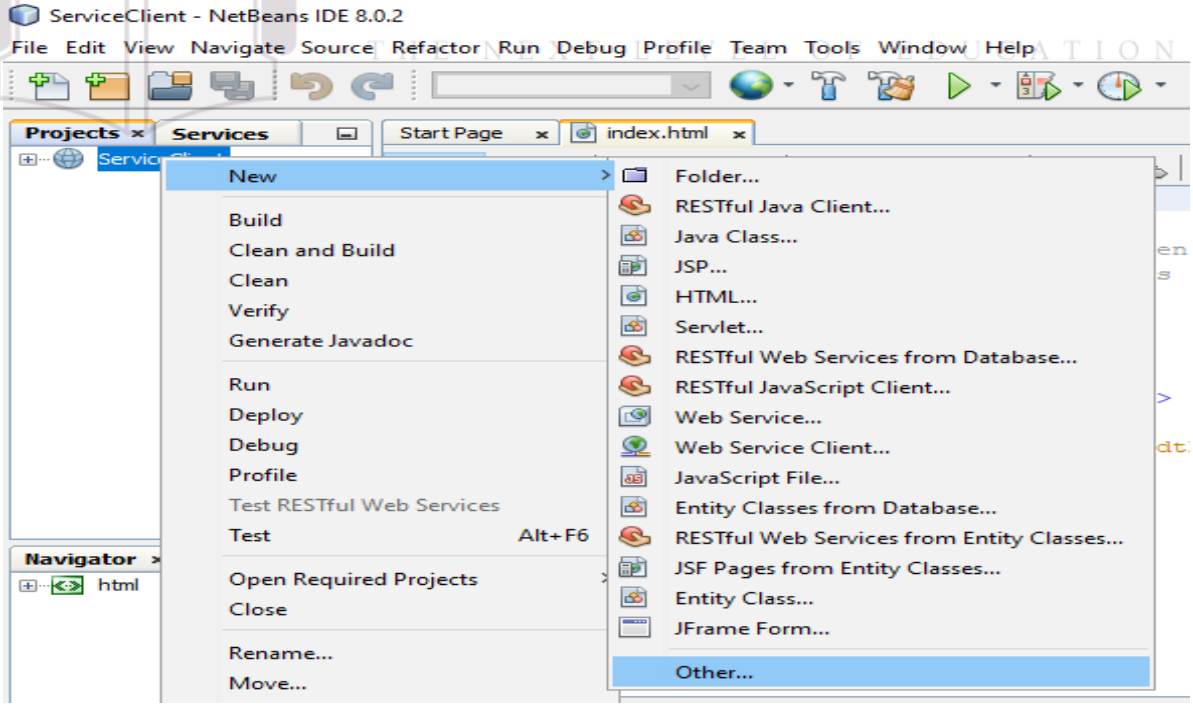
15. **Don't close Visual Studio and browser, just minimize it otherwise server will stop.** **But save the link anywhere**, so that we can use it later.

16. Now **open NetBeans**.

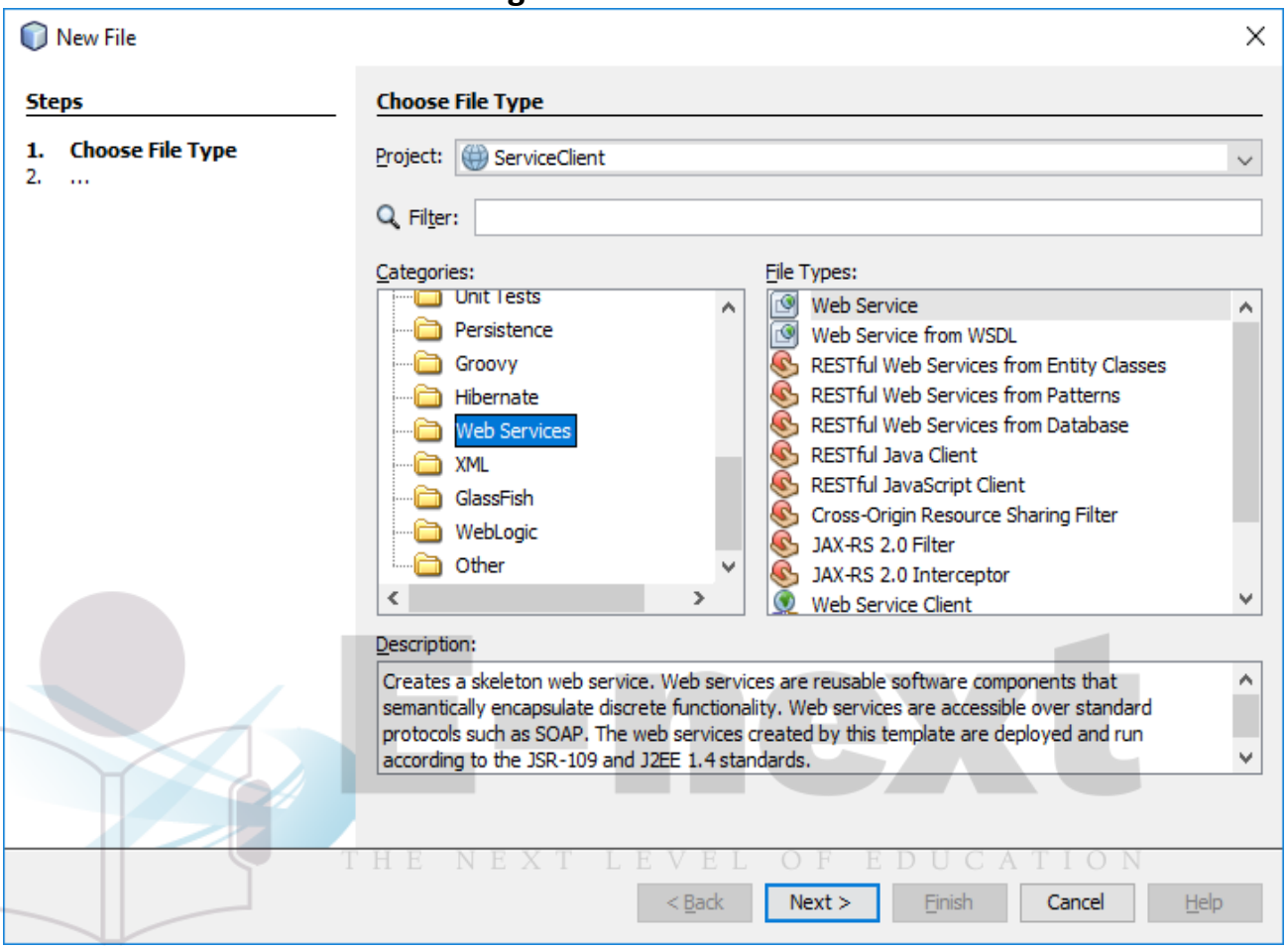
17. **Create a Web Application with name ServiceClient. Next -> Finish**.



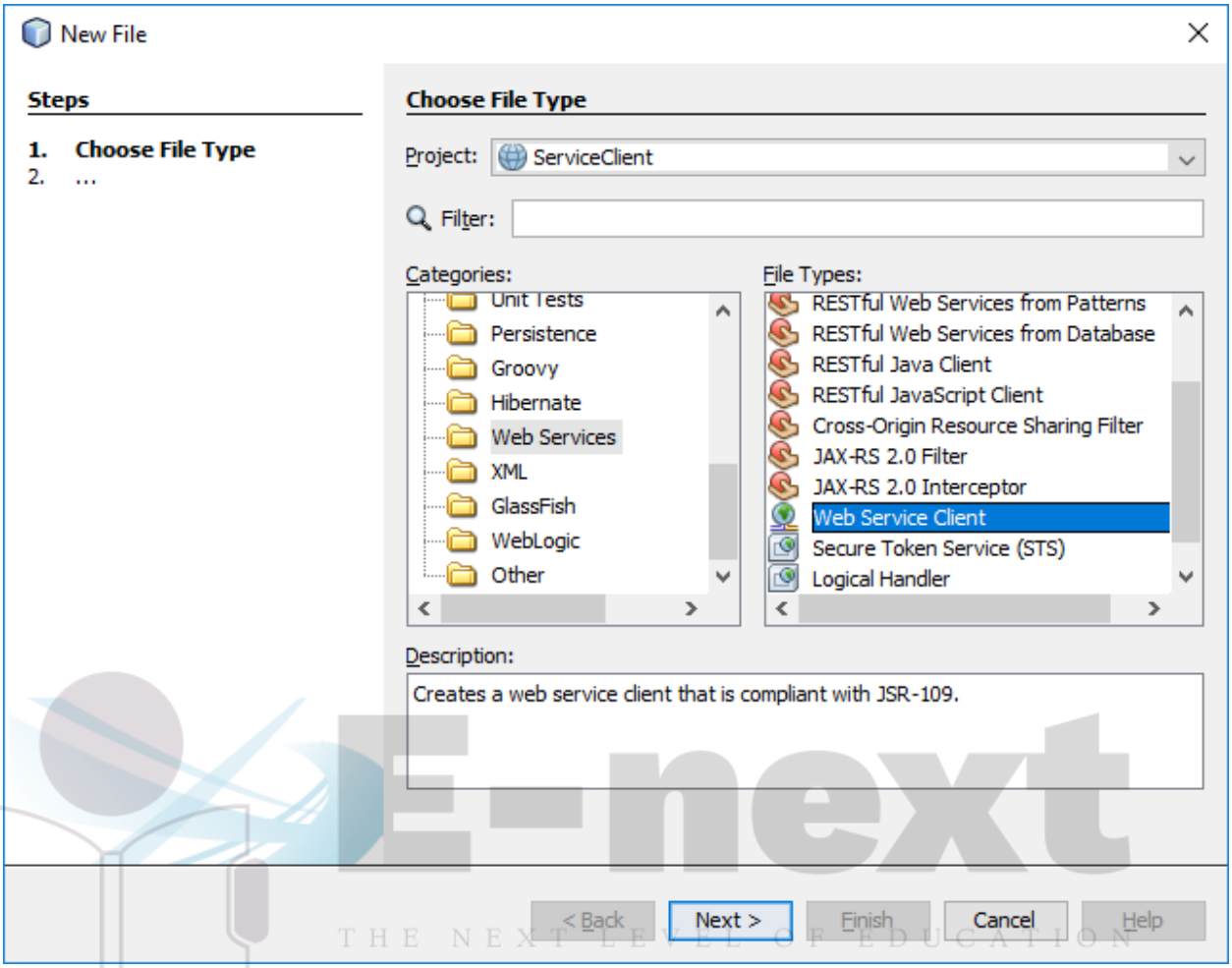
18. Now **create a Web Service Client**.
Right click on ServiceClient -> New -> Other.



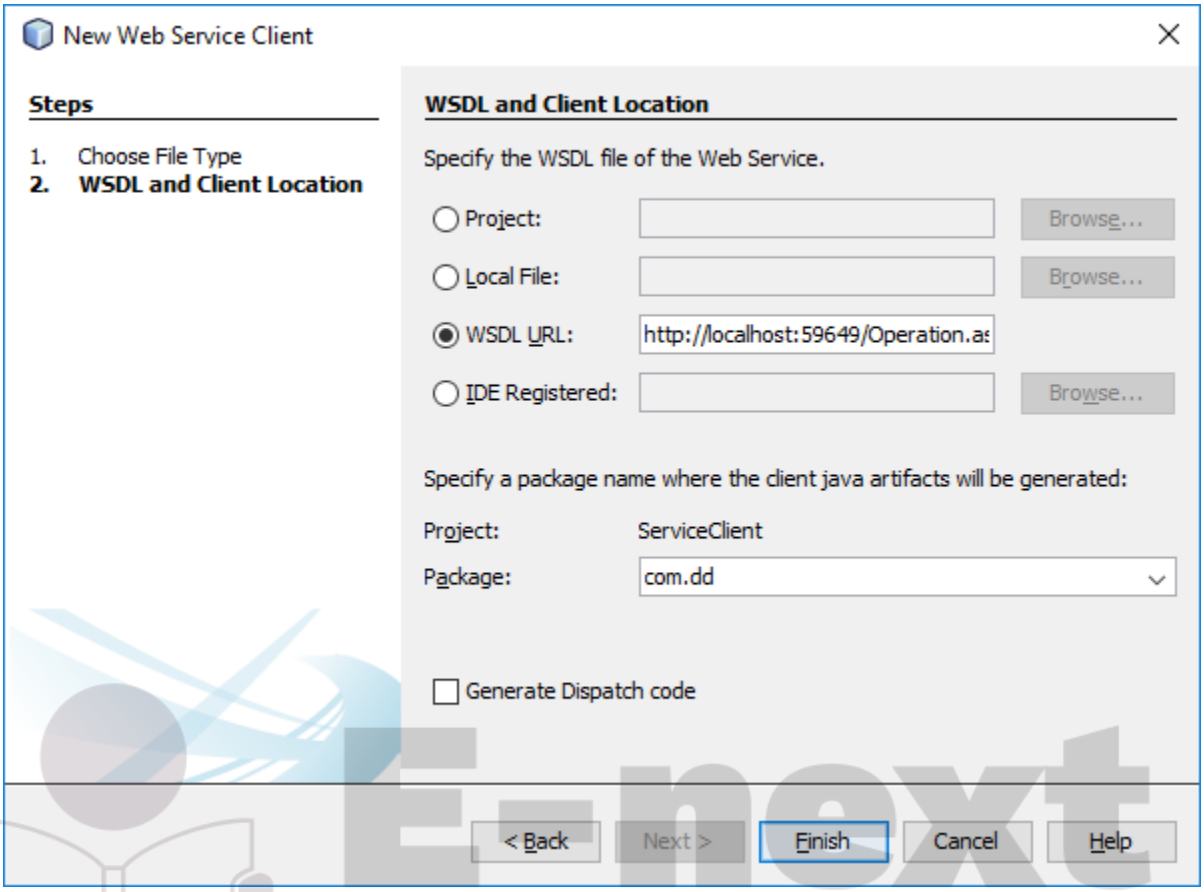
19. Now **select Web Services in Categories** section.



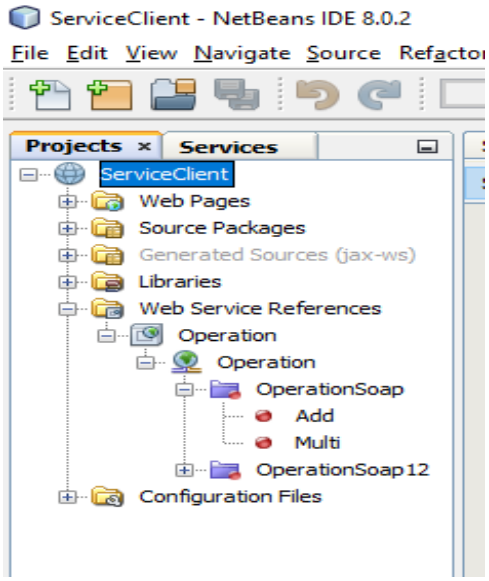
20. Select Web Service Client in File Types and **Click on Next** button.



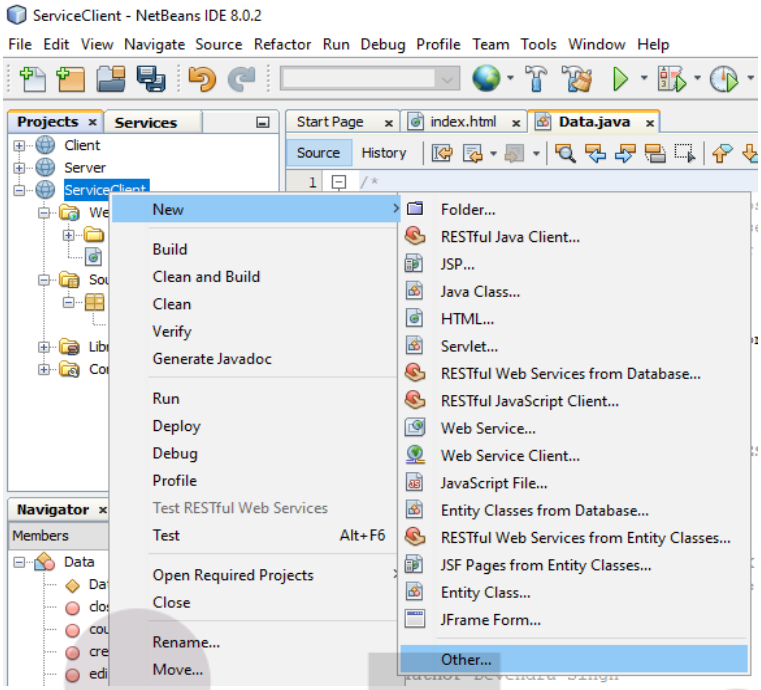
21. Select WSDL URL and paste the link that you have copied from browser on run of Visual Studio **enter package name com.dd**. After that **click on Finish** button.



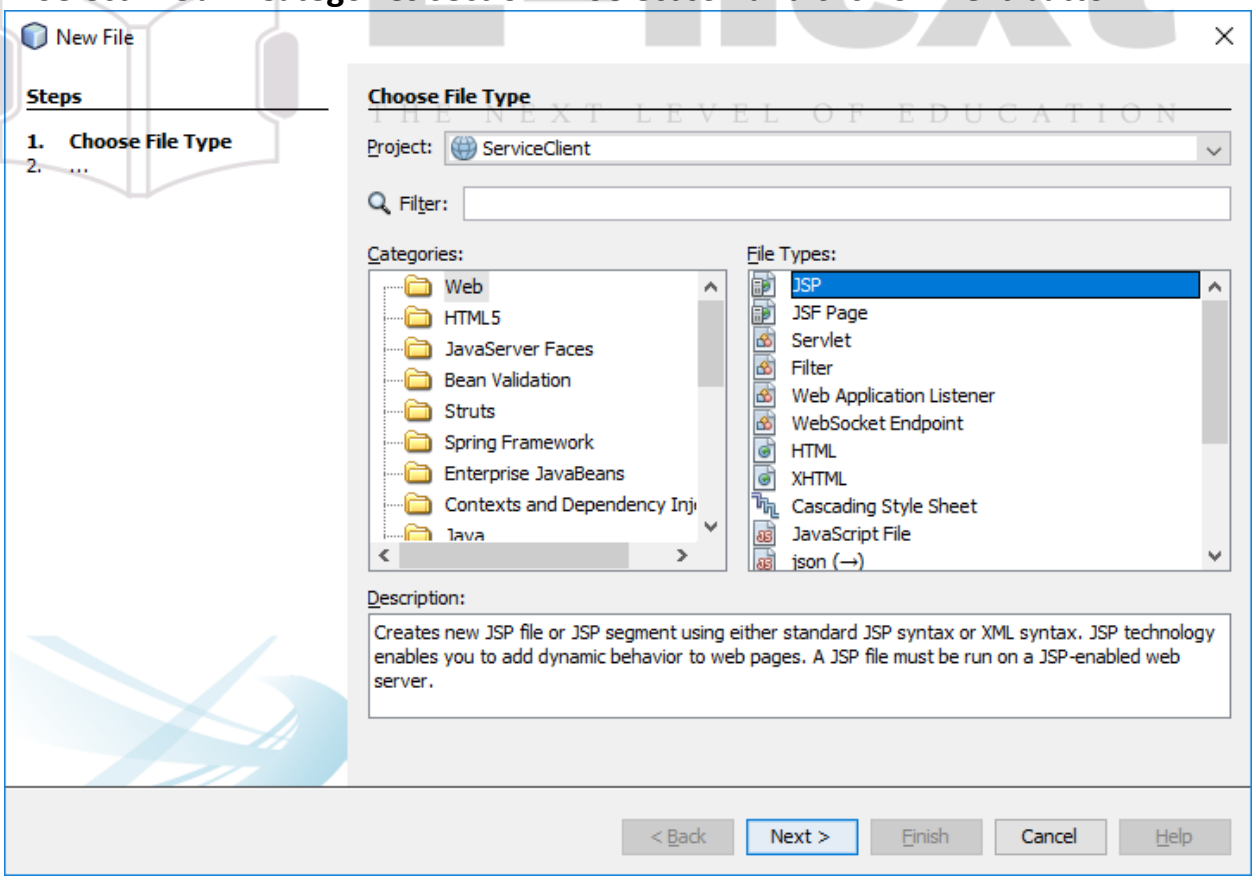
22. As you can see, we got both the service methods i.e. Add & Multi. But in this practical I'm going to use only one service method. You can do for both also.



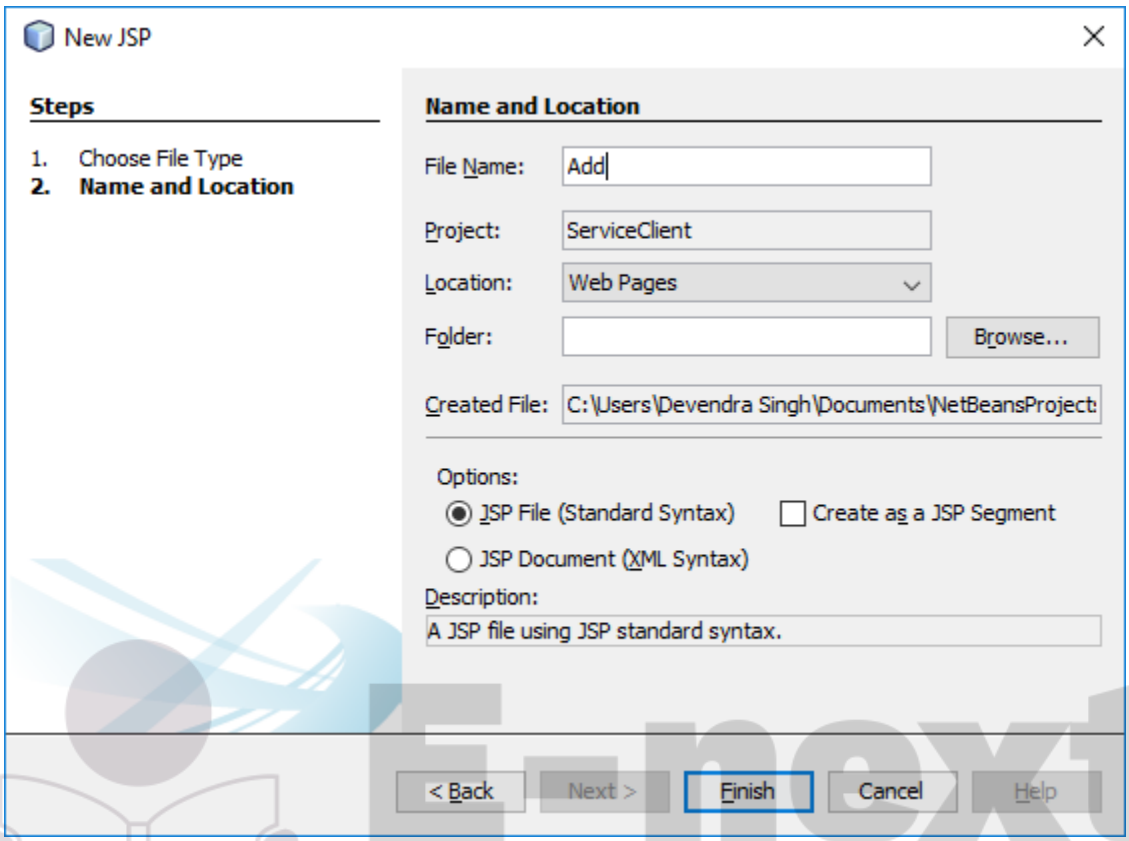
23. Now create a JSP page. **Right click on ServiceClient -> New -> Other**



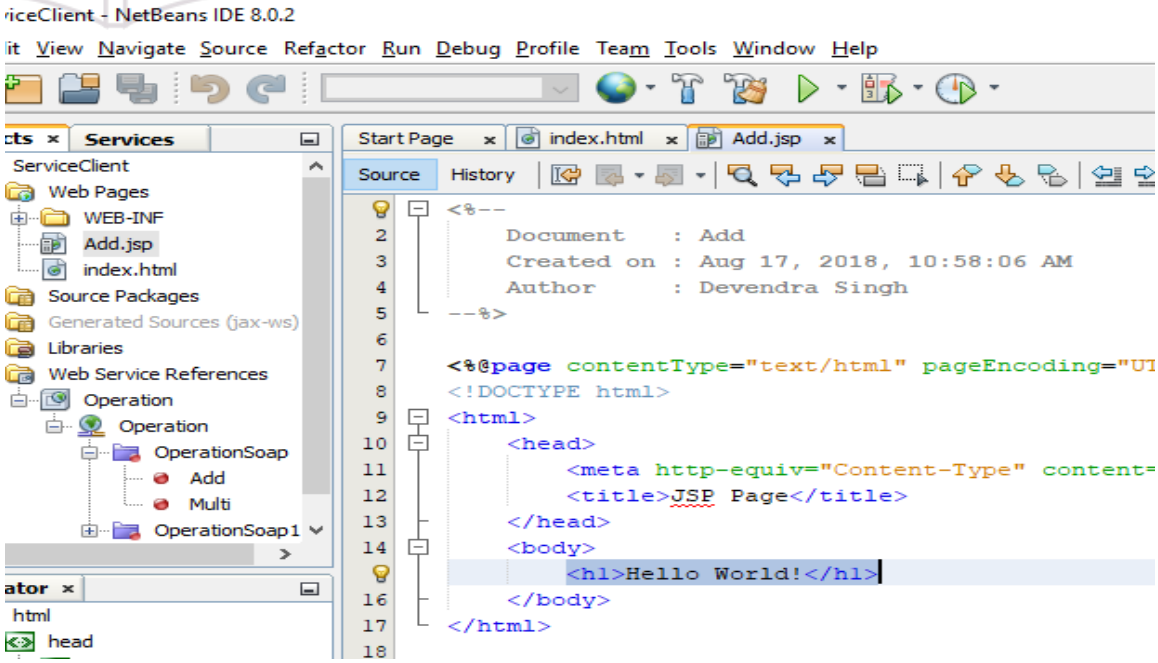
24. Select Web in Categories section -> Select JSP and click on Next button.



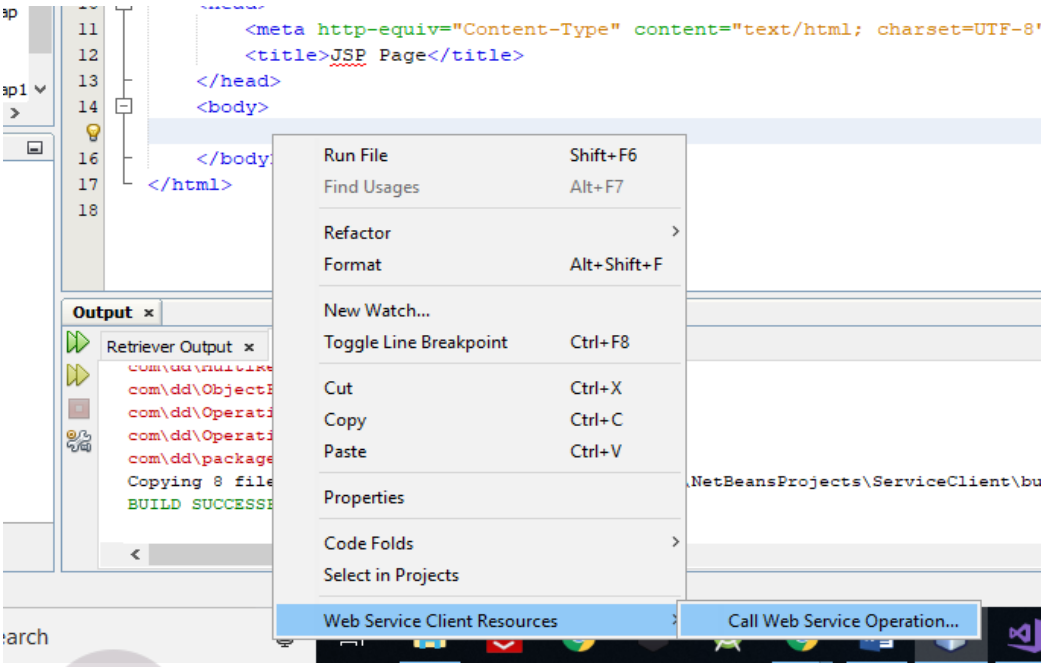
25. Enter File Name Add and **click on Finish button**.



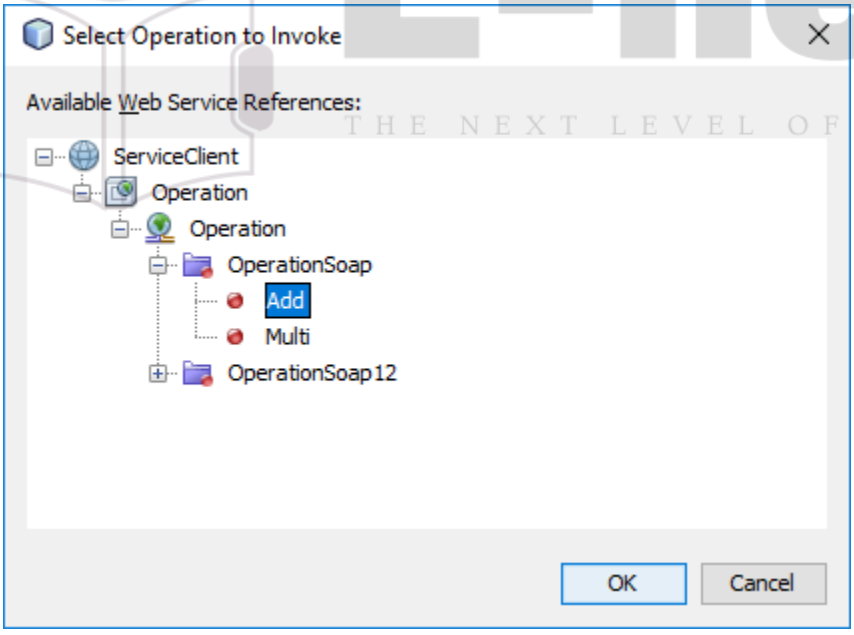
26. In Add.jsp file delete the selected part in body tag, because we don't need this.



27. Right click between body tag and **select Call Web Service Operation**.

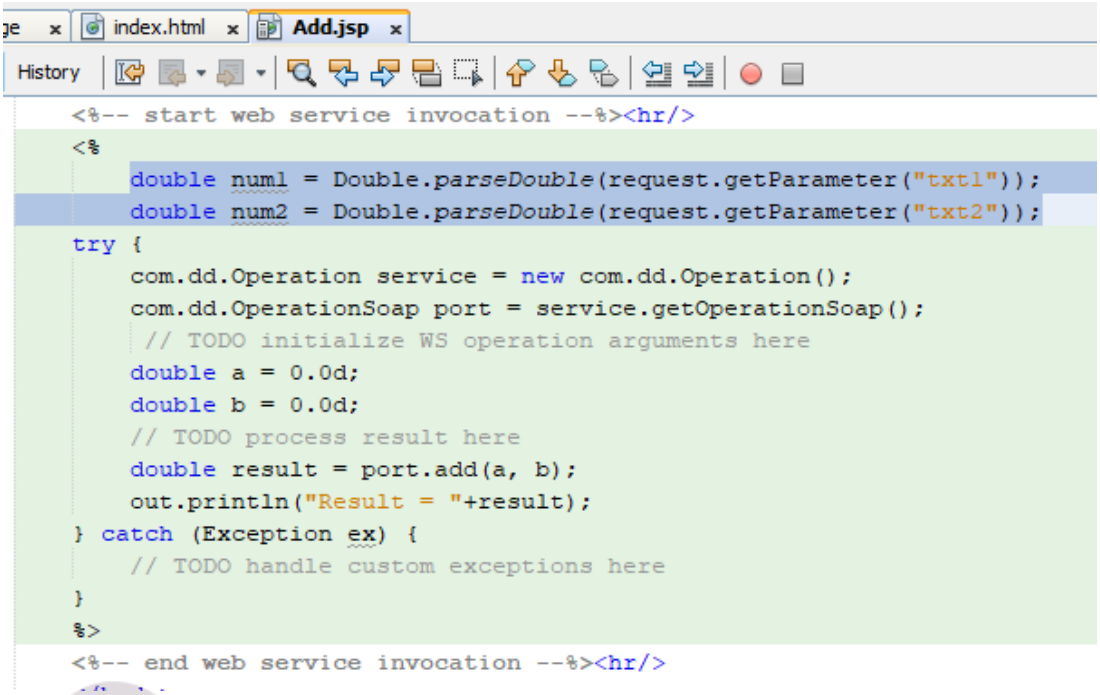


28. **Expand and select Add**. After select, **click on OK** button.

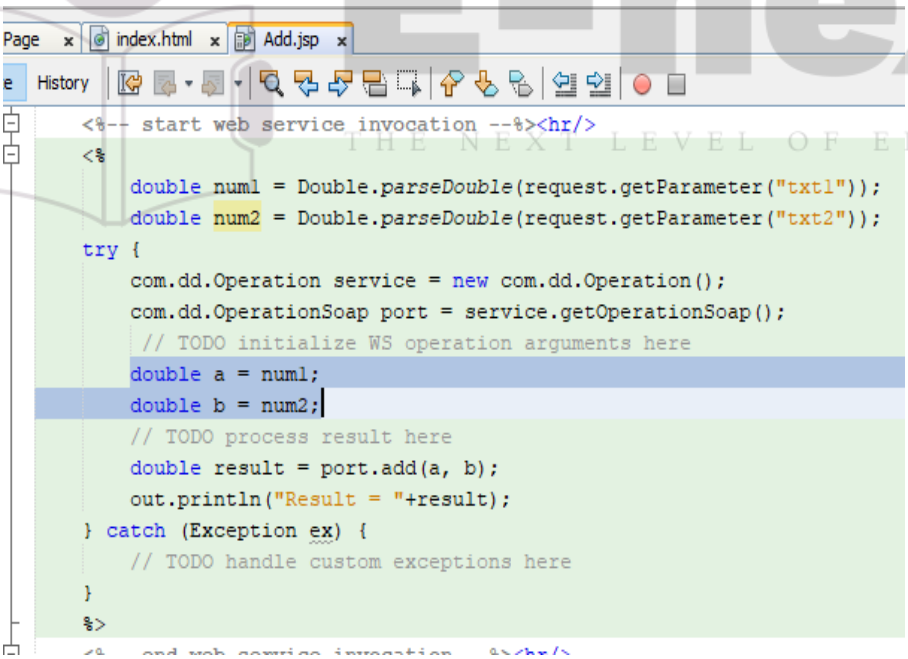


29. Now **add the following code outside of try block**.

```
double num1 = Double.parseDouble(request.getParameter("txt1"));
double num2 = Double.parseDouble(request.getParameter("txt2"));
```

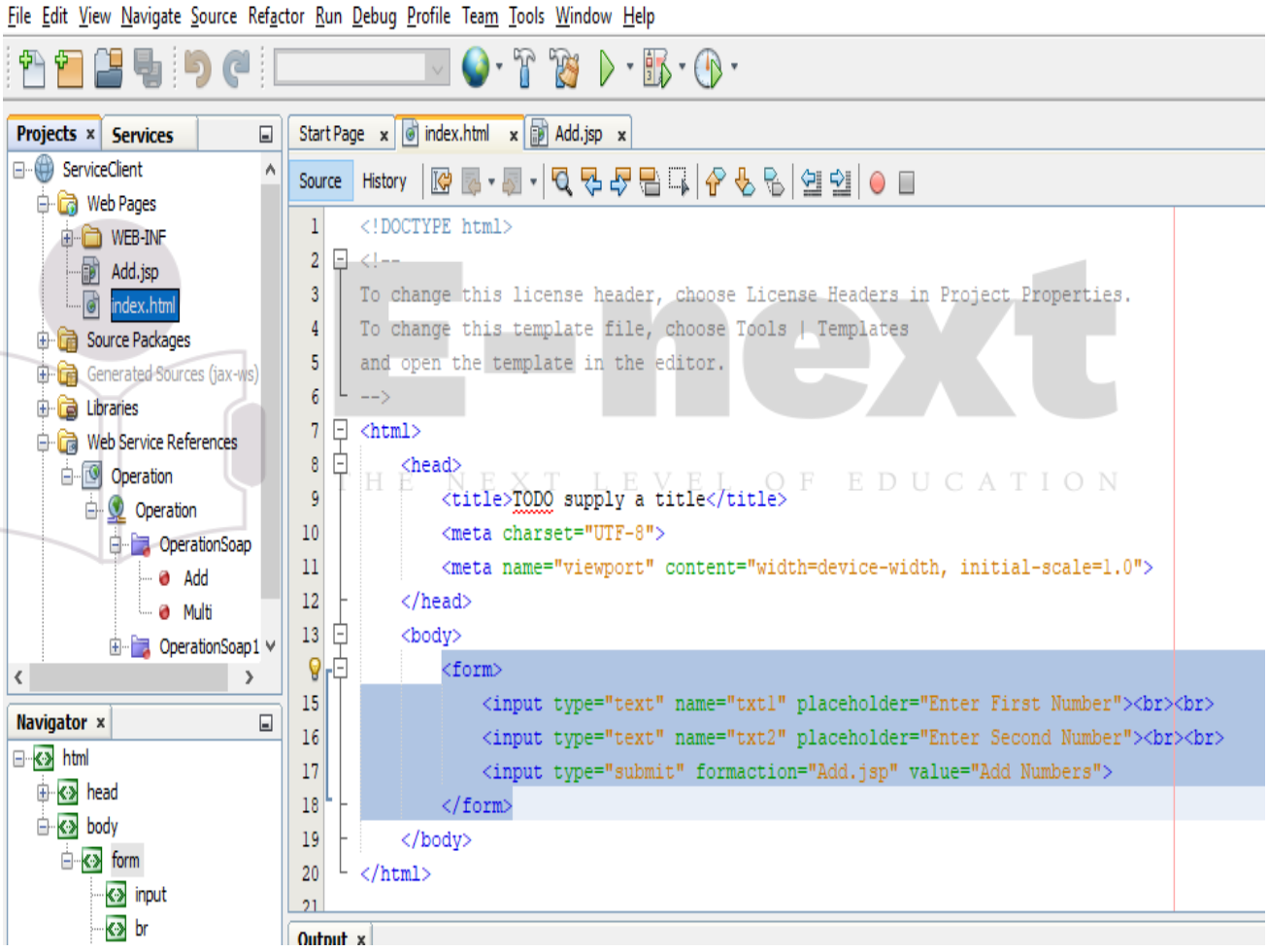


30. Now **pass num1 & num2 to a & b variable respectively**. After that **press Ctrl+S** to save this code.

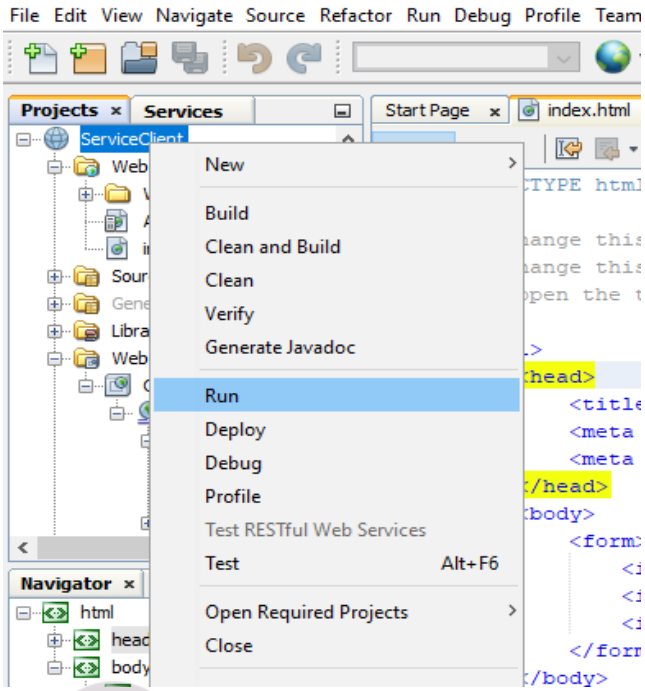


31. Now **open index.html file of ServiceClient** project and **replace the contents of body tag with following code**. **After that press Ctrl+S** to save it.

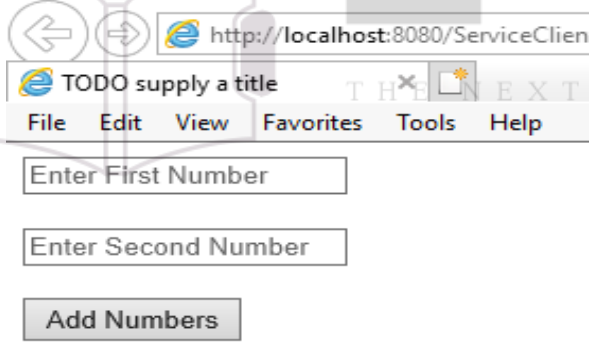
```
<form>
        <input type="text" name="txt1" placeholder="Enter First
        Number"><br><br>
        <input type="text" name="txt2" placeholder="Enter Second
        Number"><br><br>
        <input type="submit" formaction="Add.jsp" value="Add Numbers">
</form>
```



32. Now **run the ServerClient** web application.

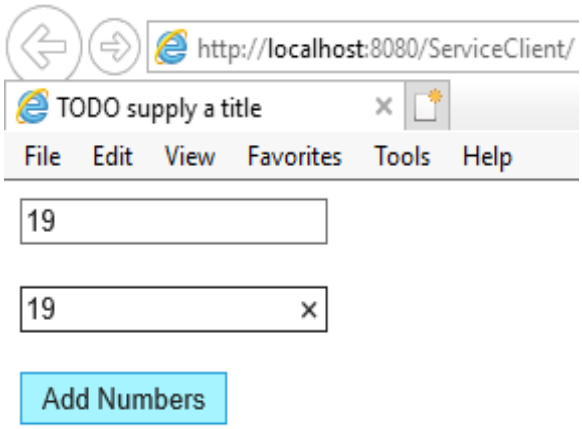


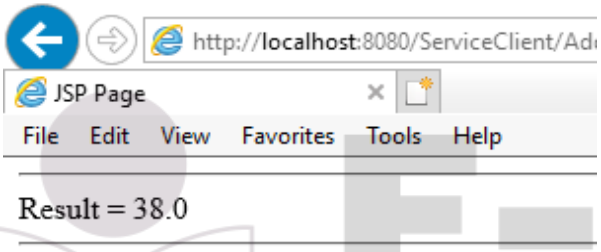
33. A window will open in browser as below.



34. Now **enter two numbers and click on Add Numbers button**. **Wait to get result**……..

You will get result on a new page.

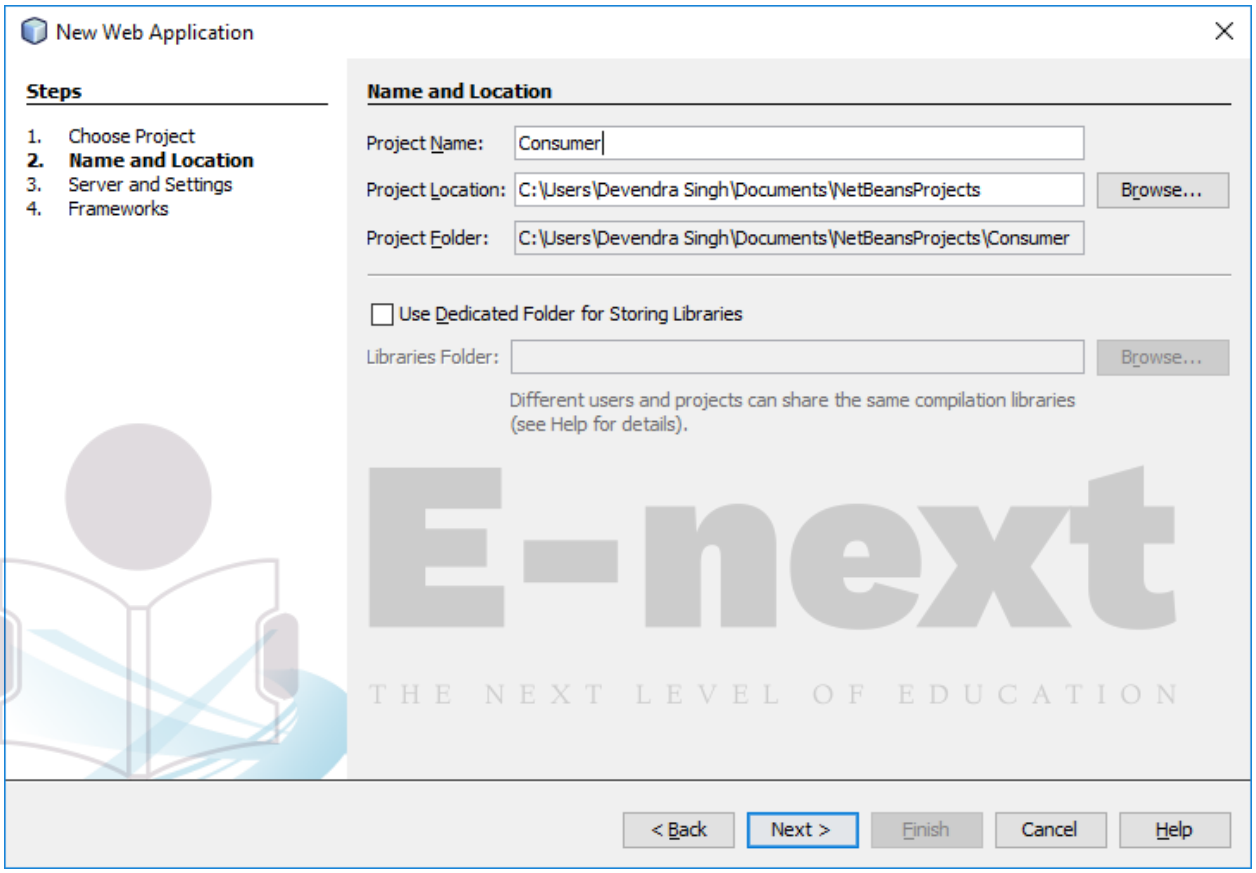




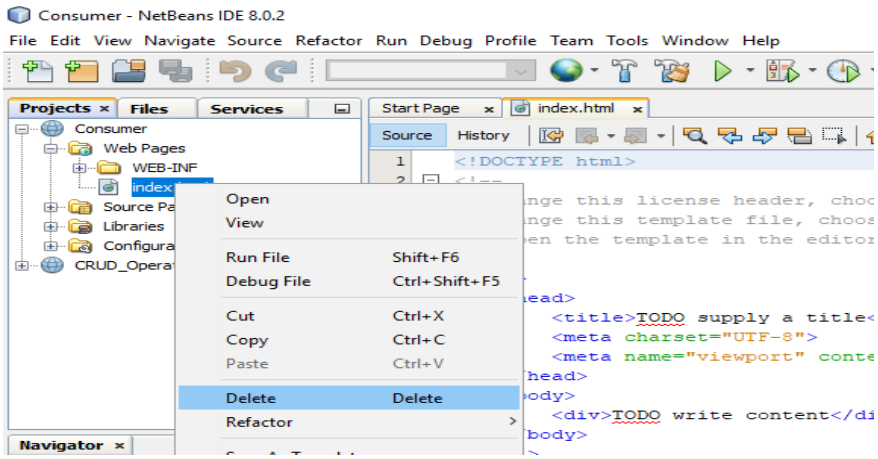
Note: To do this practical, follow the steps from 1 to 18 present in practical – 7.

After that do not close or restart the NetBeans.

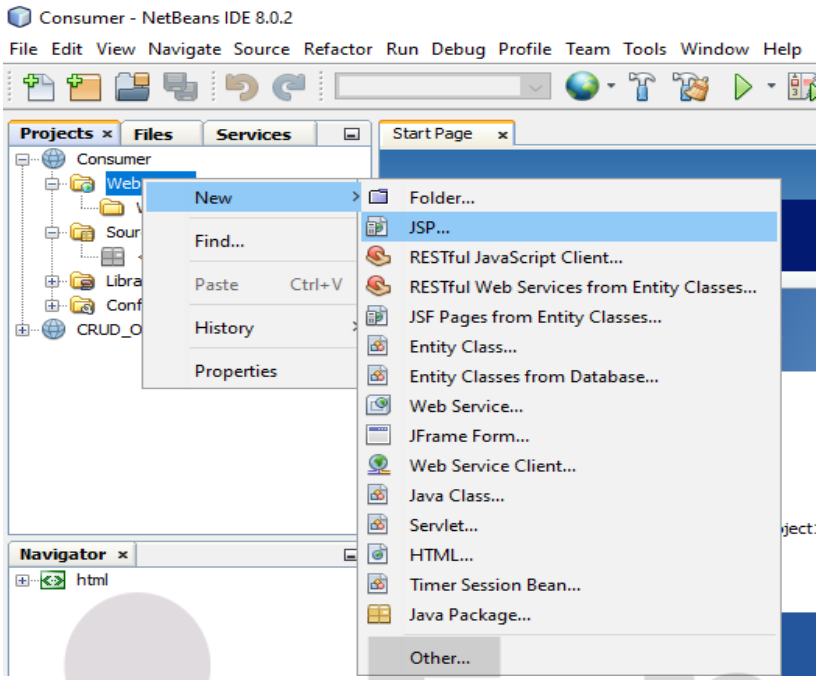
1. Create an another Web Application project and Give name as **Consumer**.



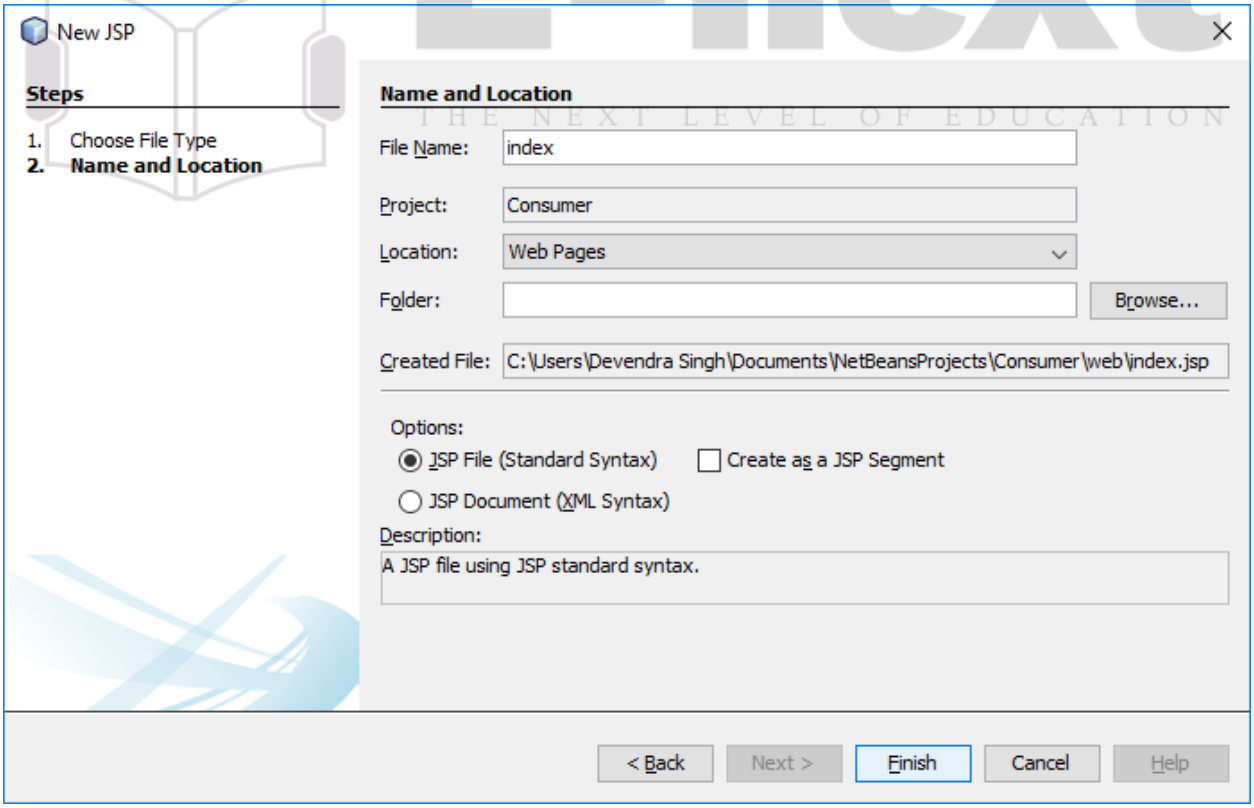
2. And create it**. Next -> Finish**.
3. Now **right click on index.html and delete it**.



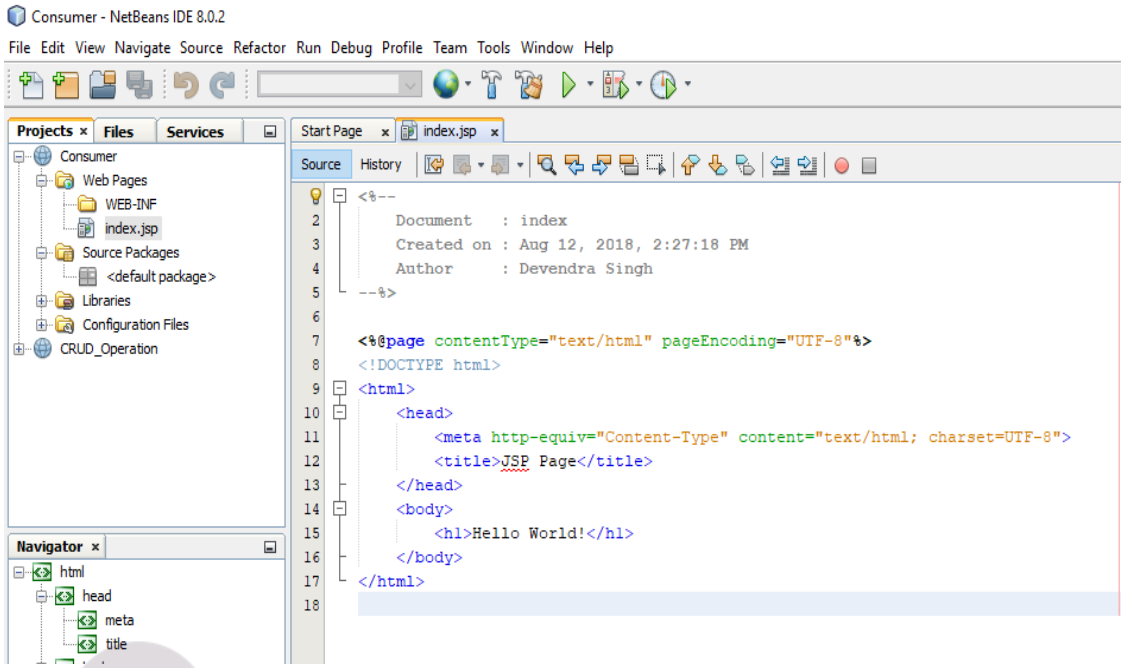
4. Now **right click on Web Pages and select JSP** to add a JSP page.



5. Give **index** name to it and then **click on Finish**.



6. Now index.jsp page will open like below.



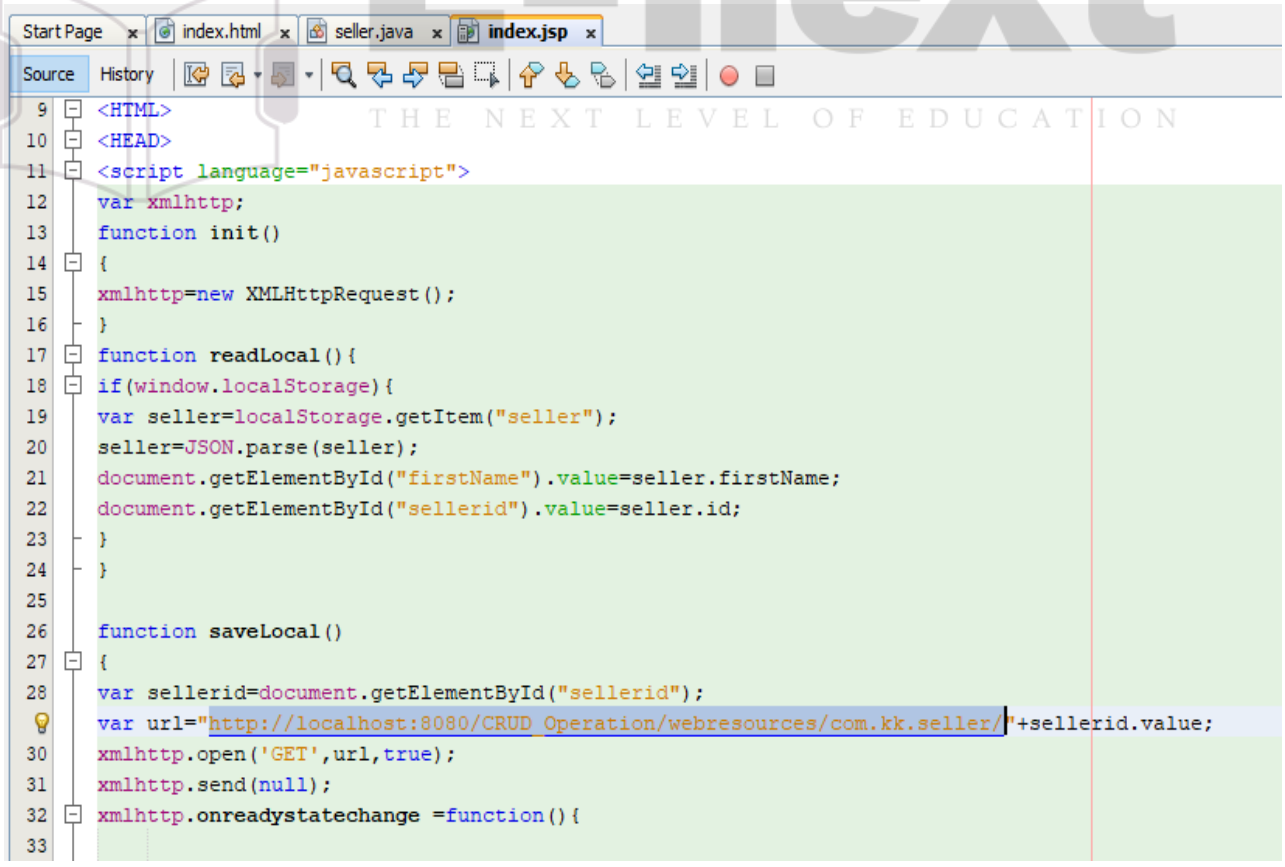
7. Now **select HTML content of index.jsp** file and **replace it** with following bold letter codes.

```
<HTML>
<HEAD>
<script language="javascript">
var xmlhttp;
function init()
{
xmlhttp=new XMLHttpRequest();
}
function readLocal(){
if(window.localStorage){
var seller=localStorage.getItem("seller");
seller=JSON.parse(seller);
document.getElementById("firstName").value=seller.firstName;
document.getElementById("sellerid").value=seller.id;
}
}
```

```
function saveLocal()
{
var sellerid=document.getElementById("sellerid");
var
url="http://localhost:8080/CRUD_Operation/webresources/com.kk.seller/"+
sellerid.value;
xmlhttp.open('GET',url,true);
xmlhttp.send(null);
xmlhttp.onreadystatechange =function(){

    if(xmlhttp.readyState===4){alert("6"+sellerid);
            if(xmlhttp.status===200){alert("7"+sellerid);
                    var seller =eval("("+xmlhttp.responseText+")");
                            if(window.localStorage){

    localStorage.setItem("seller",JSON.stringify(seller));
                                    alert("information stored
successfully"+seller.firstName);
                                            }
                                            else{
                                                    alert("notStored");
                                            }
                                    }
                                    else
                                       alert("error");
                                    }
                                    };
                                    }
</script>
</head>
<body onLoad ="init()">
<table>
<tr>
<td>Enter id:</td>
<td><input type="text" id="sellerid"/>
```

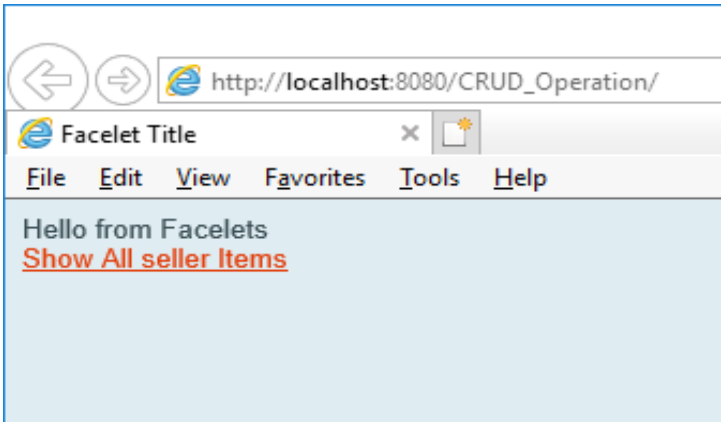
```
<input type="button" value="load employee in local browser"
onClick="saveLocal()"/>
</td>
</tr>
<tr>
<td>read from local</td>
<td><input type="button" value="Send values" onClick="readLocal()"/></td>
</tr>
<tr>
<td>first Name:</td>
<td> <input type="text" id="firstName"/></td>
</tr>
</table>
</body>
</html>
```

8. Now in the following pic, **highlighted URL is most important**. I'm explaining it next step.



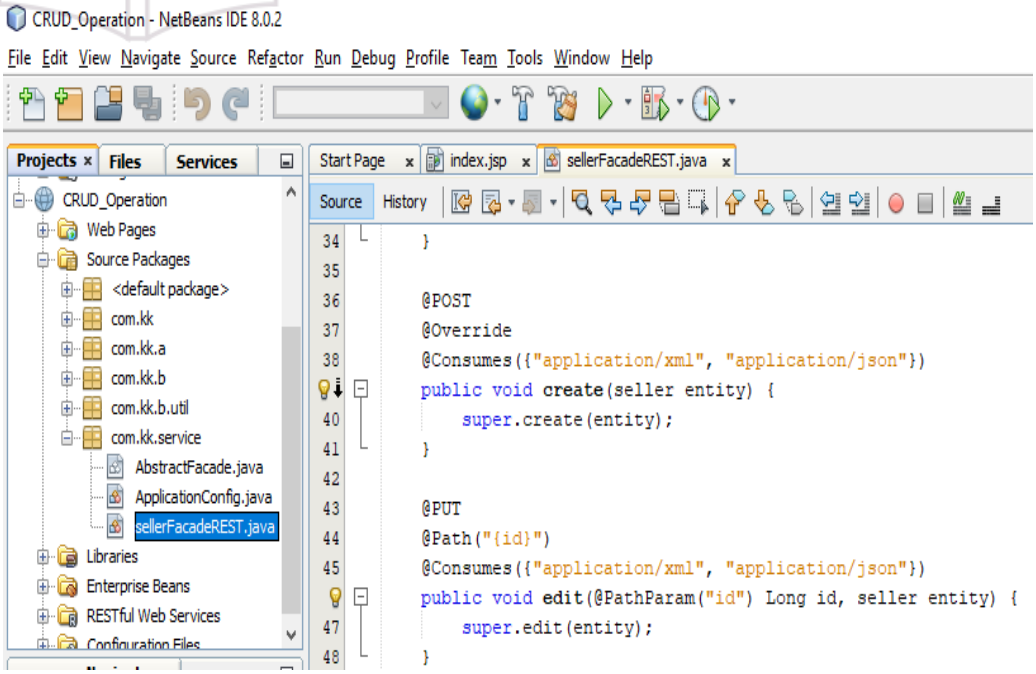
9. **http://localhost:8080/CRUD_Operation/webresources/com.kk.seller/**

Red part is the URL of which is obtained in browser by running of CRUD_Operation web application.

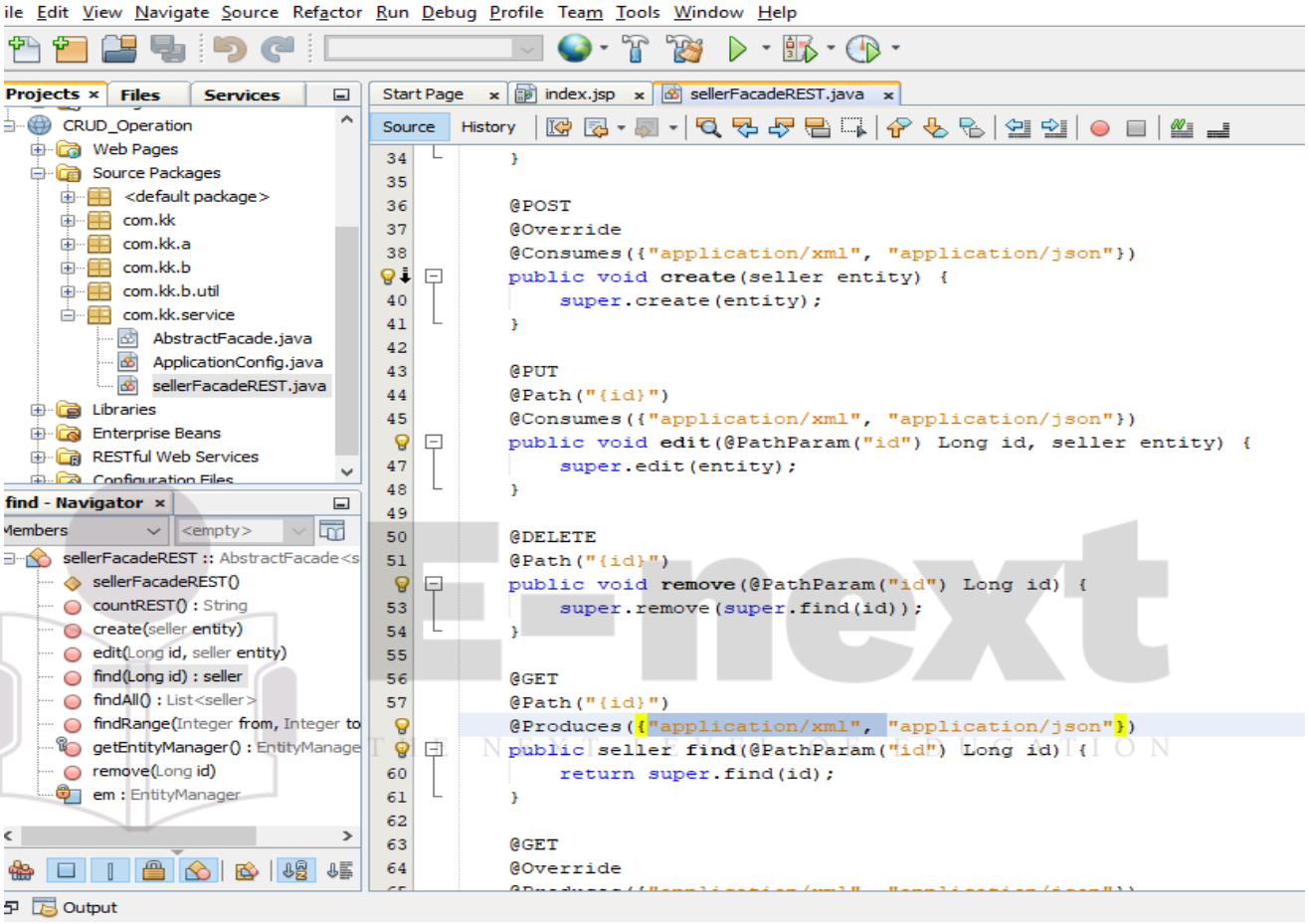


Blue part in above link is **static**. But **red part** is **dynamic** which is based on practical 7. So if you have changed name anywhere then the above URL will change accordingly.

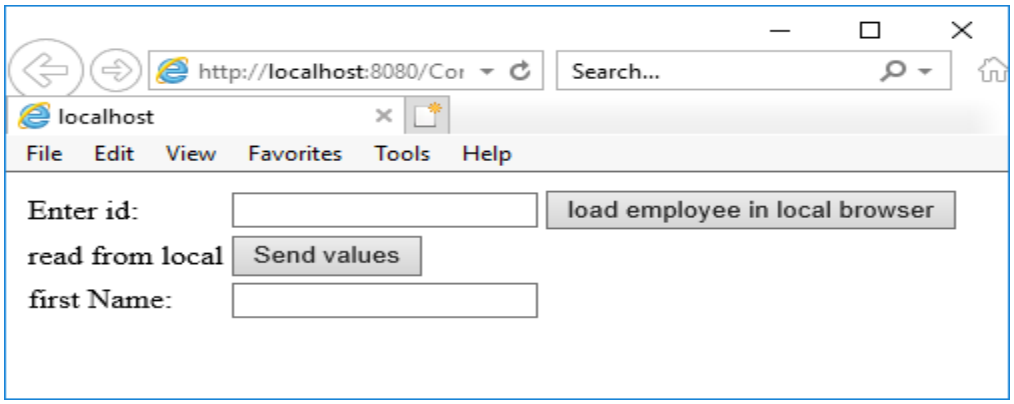
10. Now **open sellerFacadeREST.java** file by follow below pic **available in CRUD_Operation web application**.



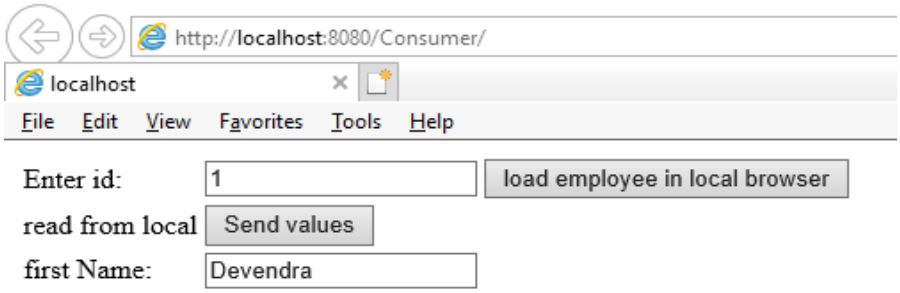
11. Now **delete the selected part**. Because it will return data in XML format to the consumer. But we have written javascript in **index.jsp** for JSON data format only. After that **deploy the CRUD_Operation** web application; so that it will update the changes.



12. Now **run the Consumer** application. A window will open in browser like below.



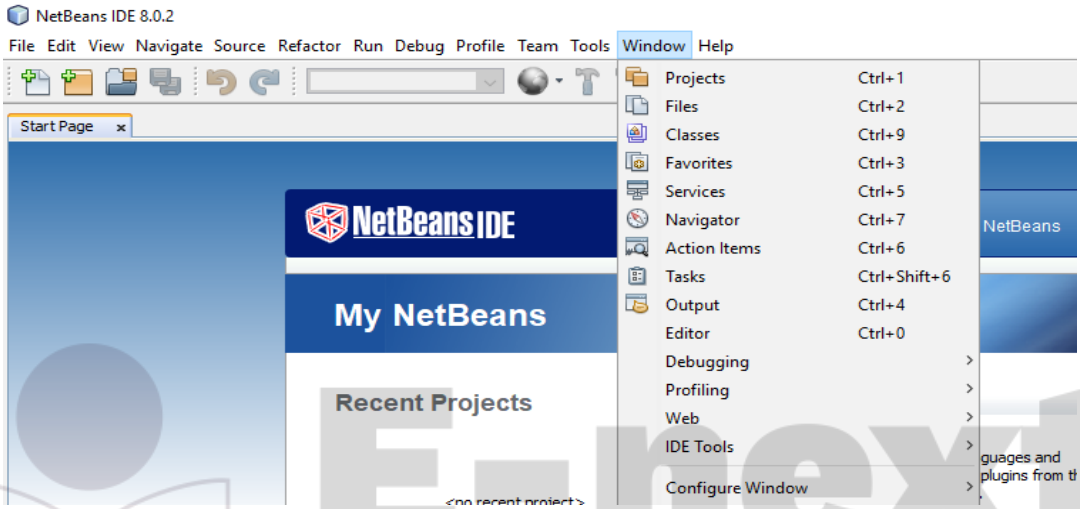
13. Now if you will **enter an id** into **Enter id** textbox **which you have created in database (id from data in last step of practical 7)** and then click on **load employee in local browser** button. It will get the detail of particular entered id and will store it into local storage of browser. Now click on **Send Values** to get the first name of entered id.



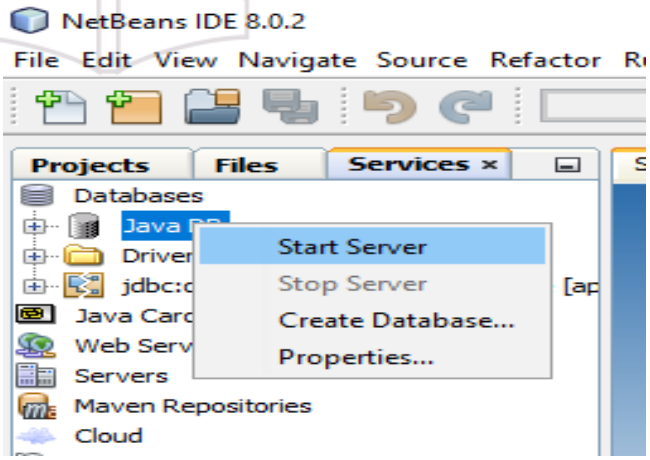
PRACTICAL-6

Aim: Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.

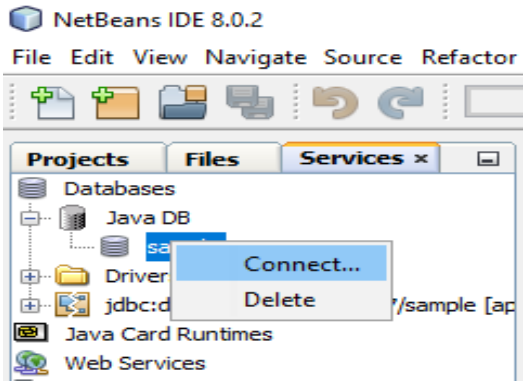
1. Click on Window menu and click on **Projects, Files** & **Services** to open it.



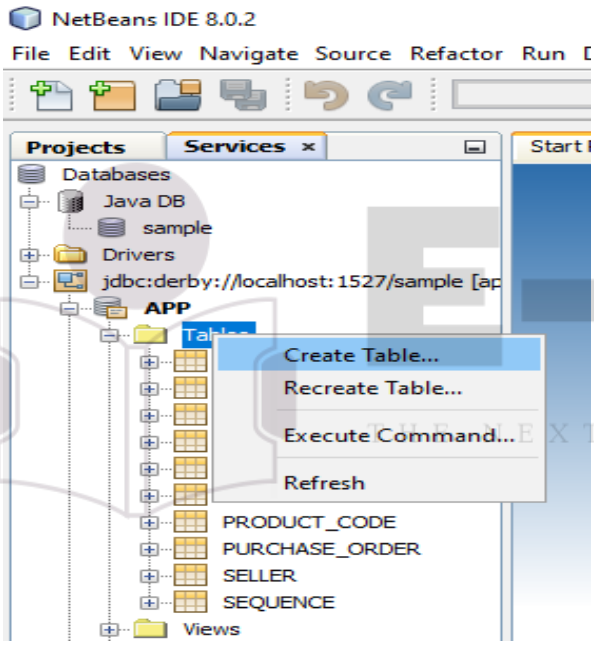
2. **Right click on Java DB** and then **click on Start Server** to start the server .



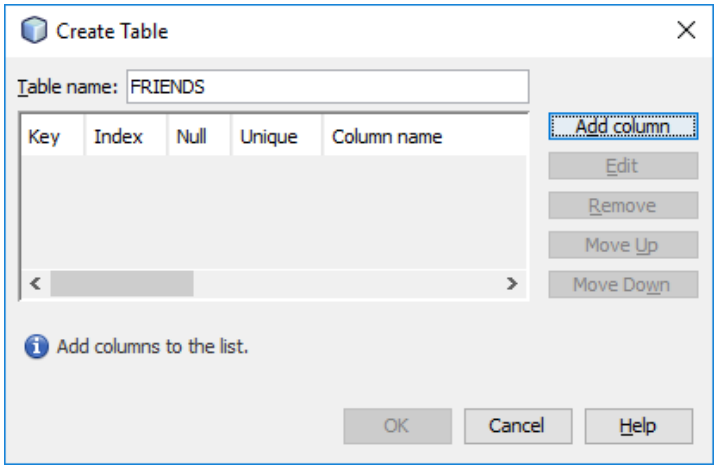
3. Now expand Java DB and **right click on sample** and then **click on connect** to connect the sample database with server.



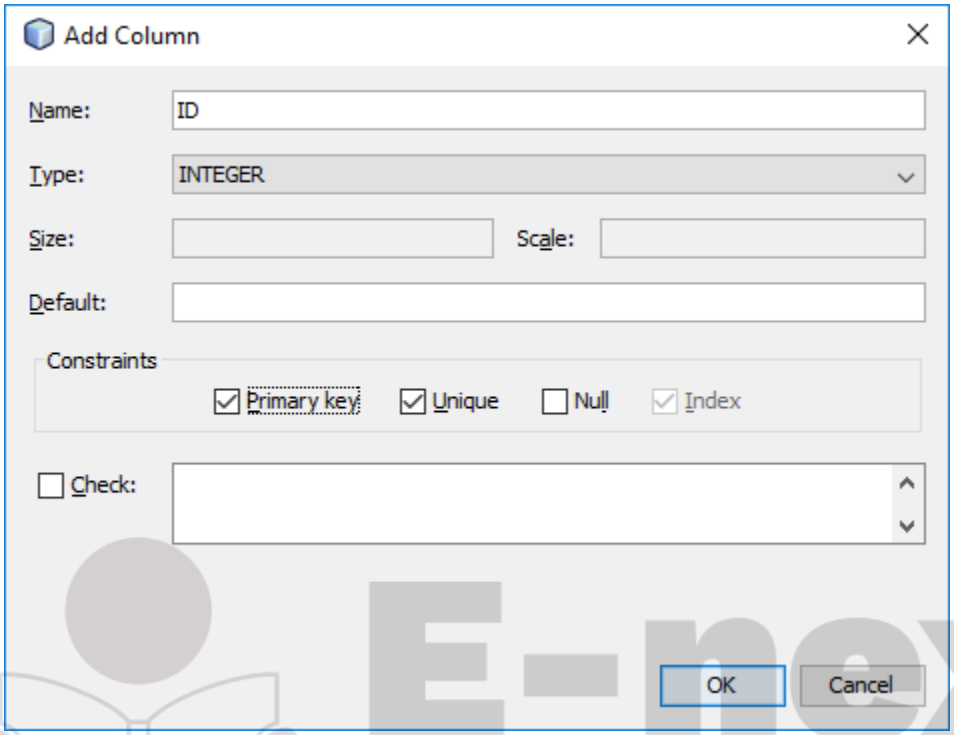
4. Now we are going to create a table in default database **sample**.
   Right click on Table -> Create Table



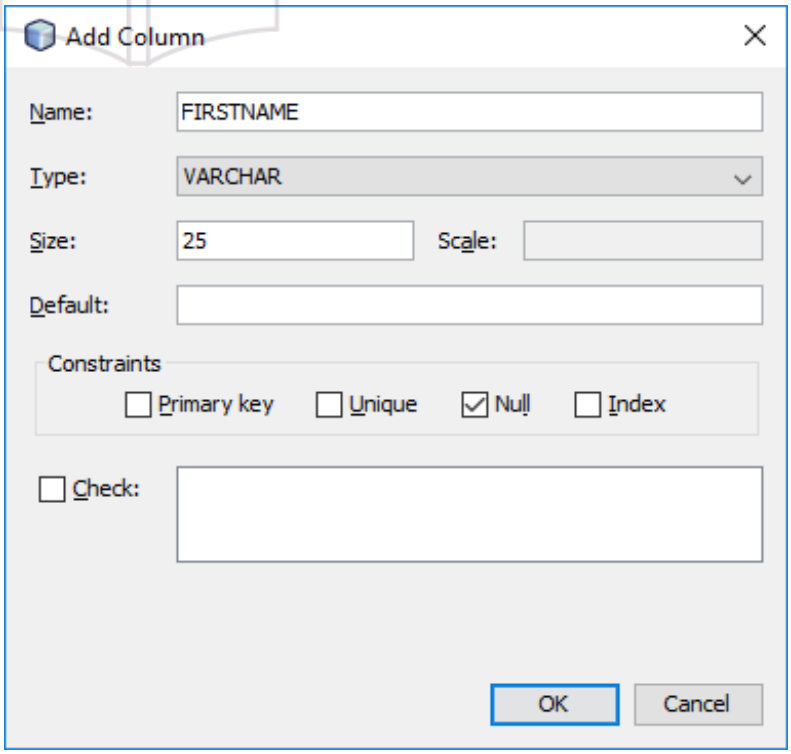
5. Give **table name** as **FRIENDS**.



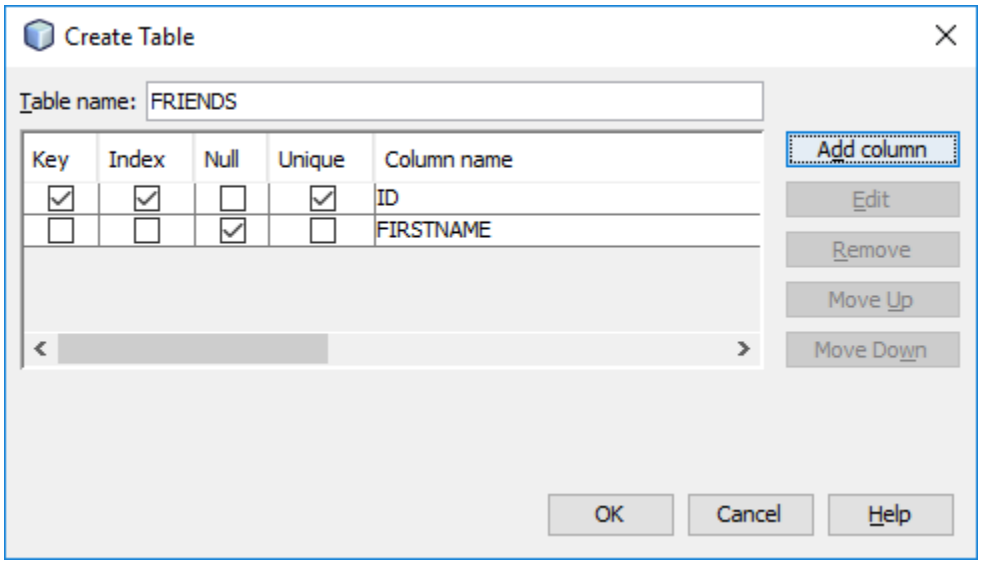
6. Now **click on Add column** button **to add columns** in table.
   Enter details as in below pic and **select Primary key**. After that **click on OK** button.



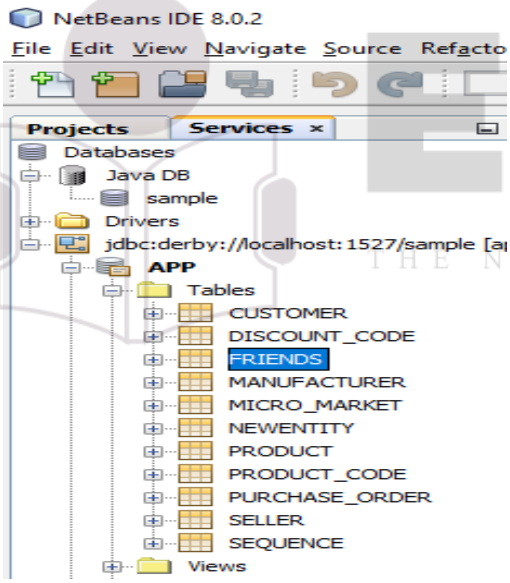
7. Now **add second column** with following detail. But **don't select primary** and click on OK button.



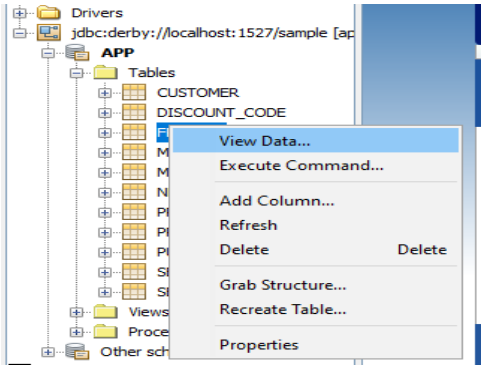
8. Now **click on OK** button.



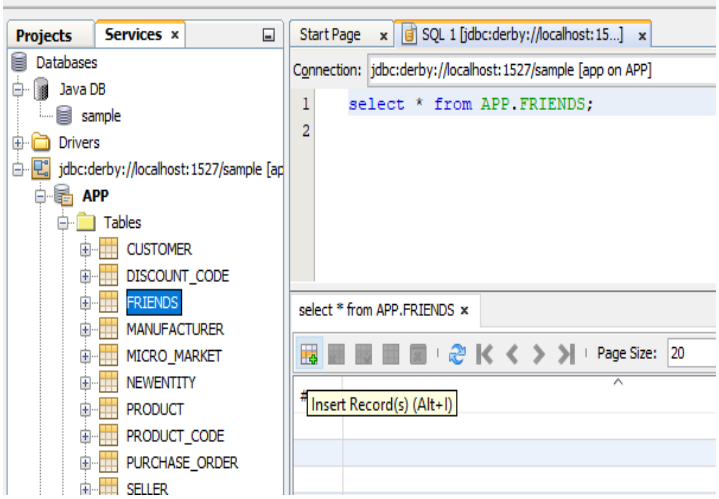
9. Now you can see a table with name **FRIENDS** in the table.



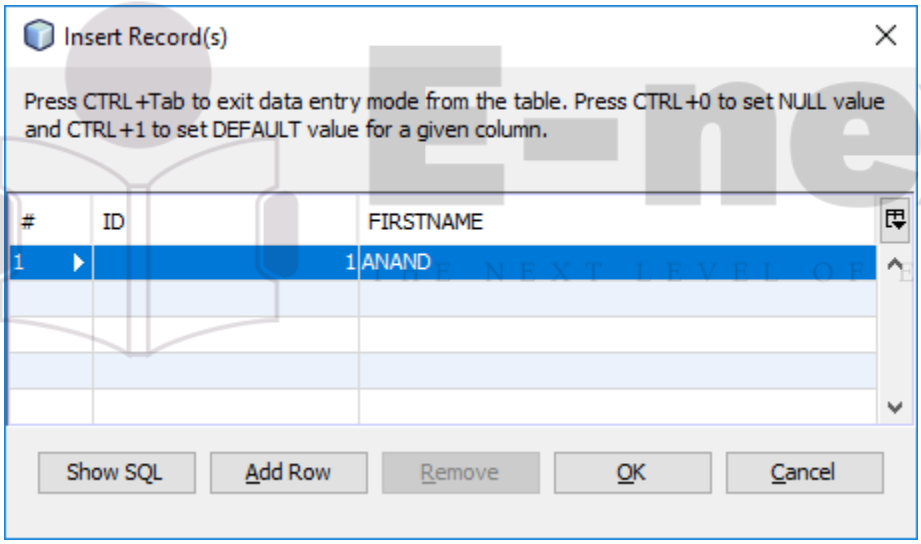
10. Right click on FRIENDS to view and add records into it.



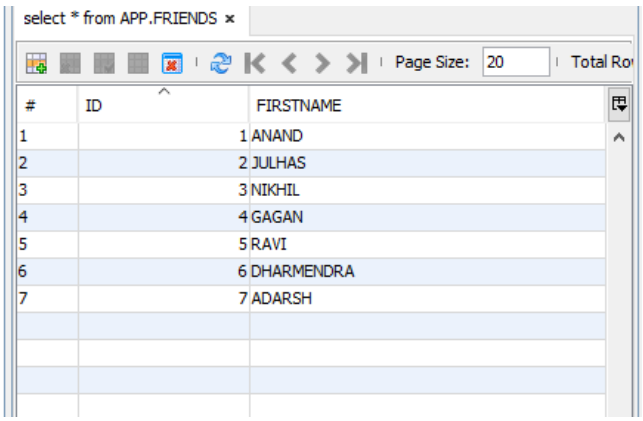
11. Now **click on the leftmost icon in second panel to insert some record**.



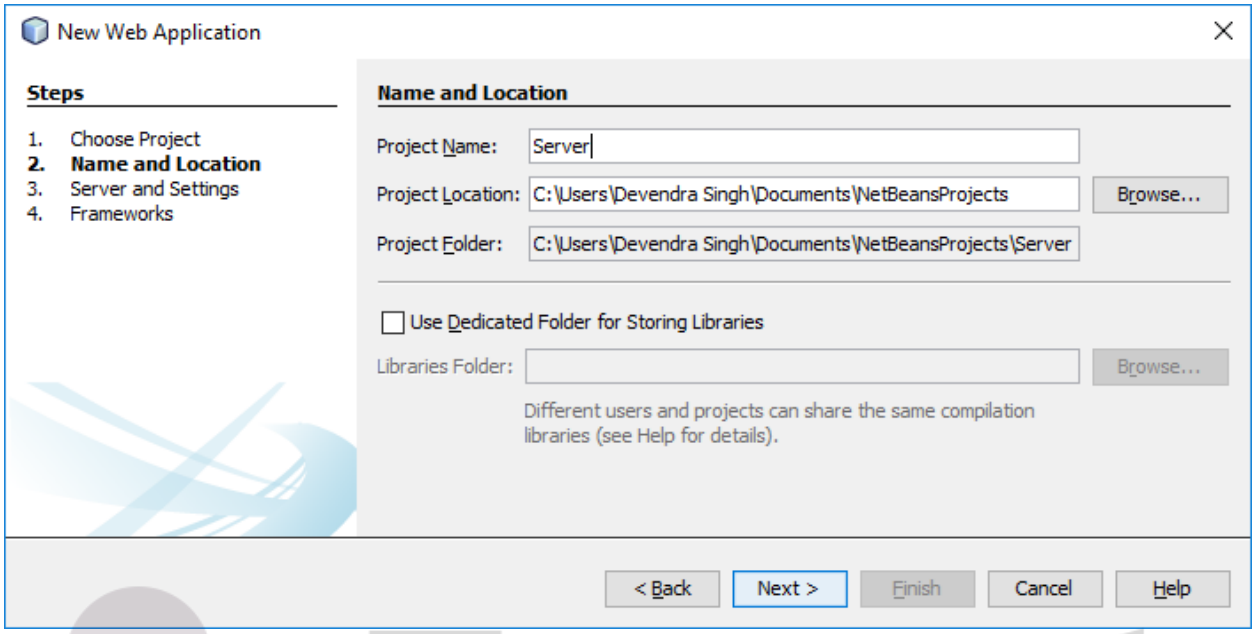
12. Insert a record and then **click on Add Row button to insert more record**. After that **click on OK button** to finish.



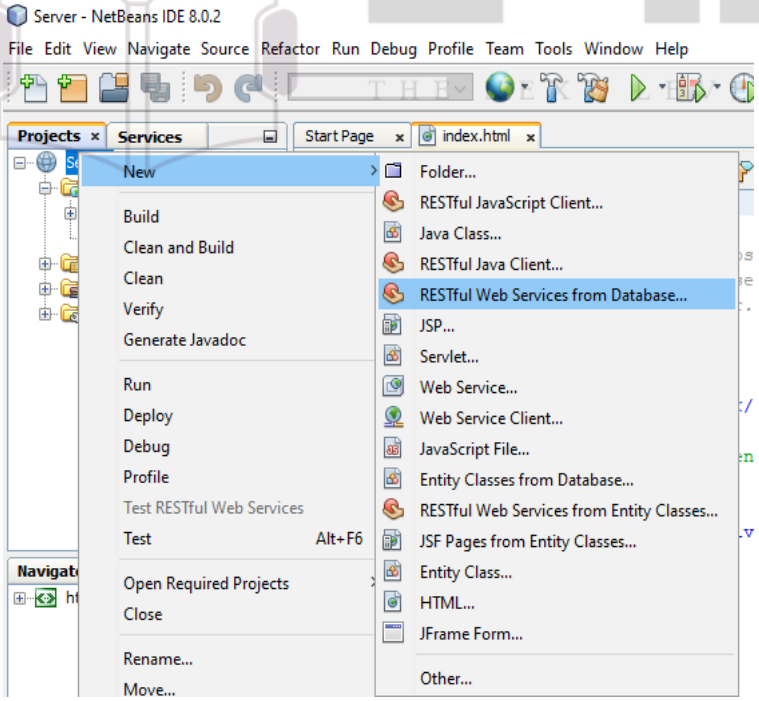
13. As you can see, I have entered 7 records.



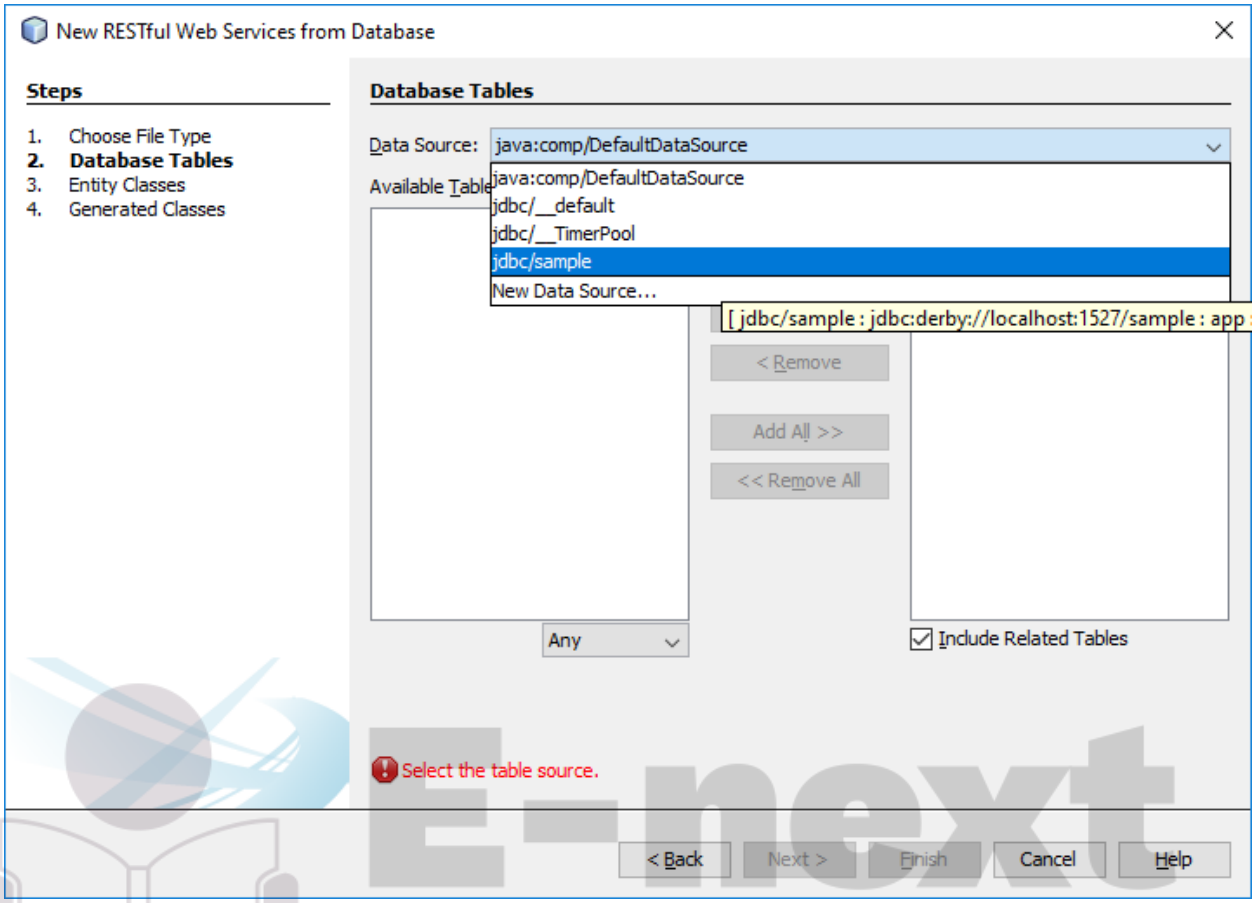
14. Now **create a web application** with name **Server**. After that **click on Next and then Finish** button.



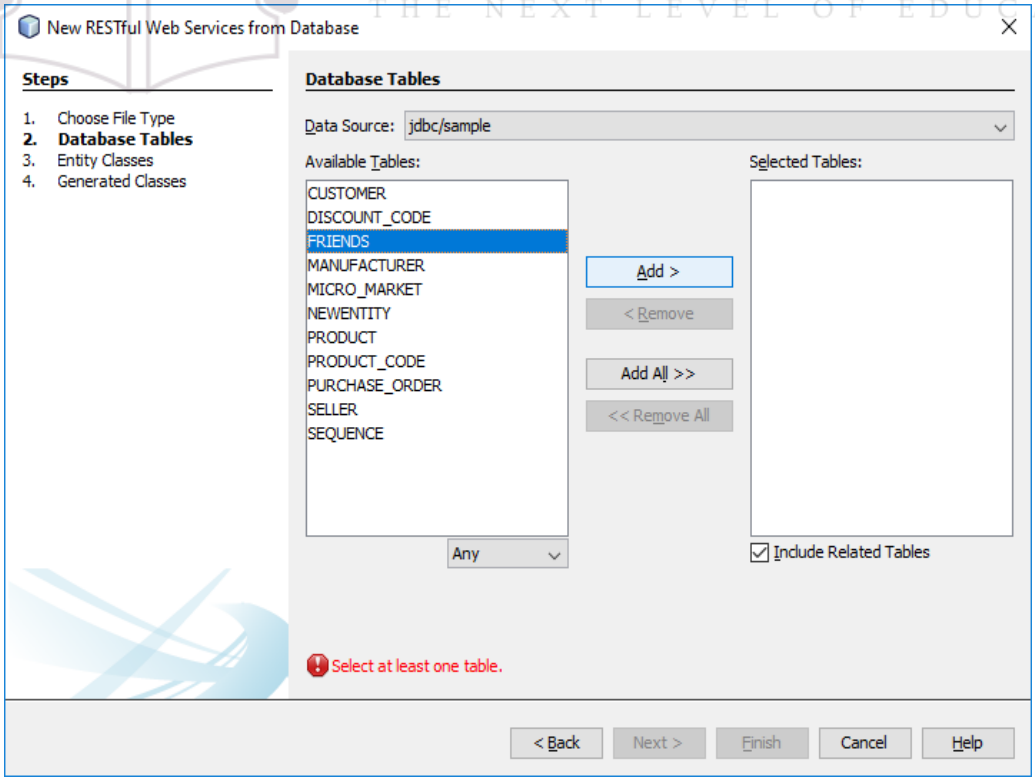
15. Now **create a RESTful Web Service from Database by right click on project** name.



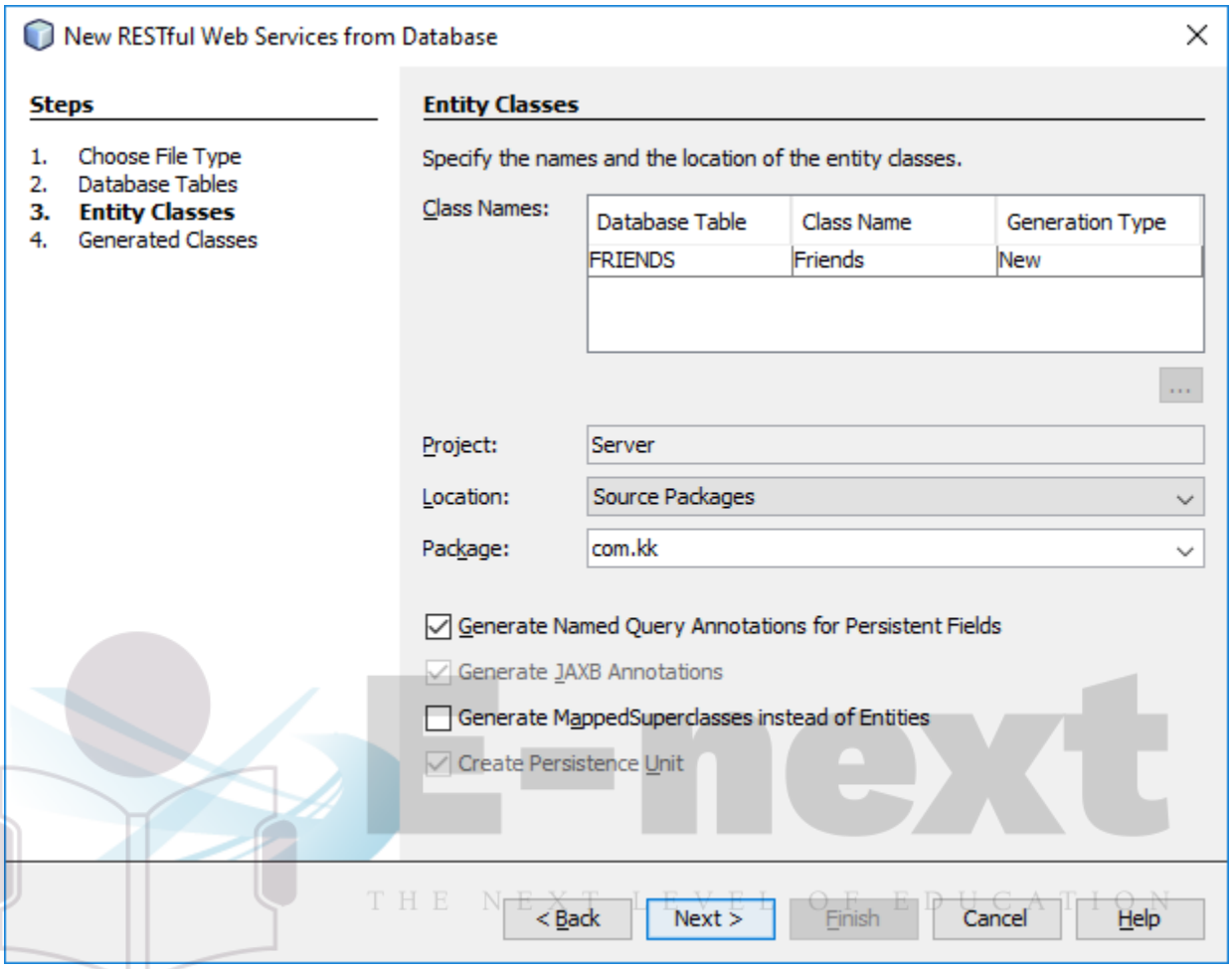
16. Choose **Data Source jdbc/sample**.



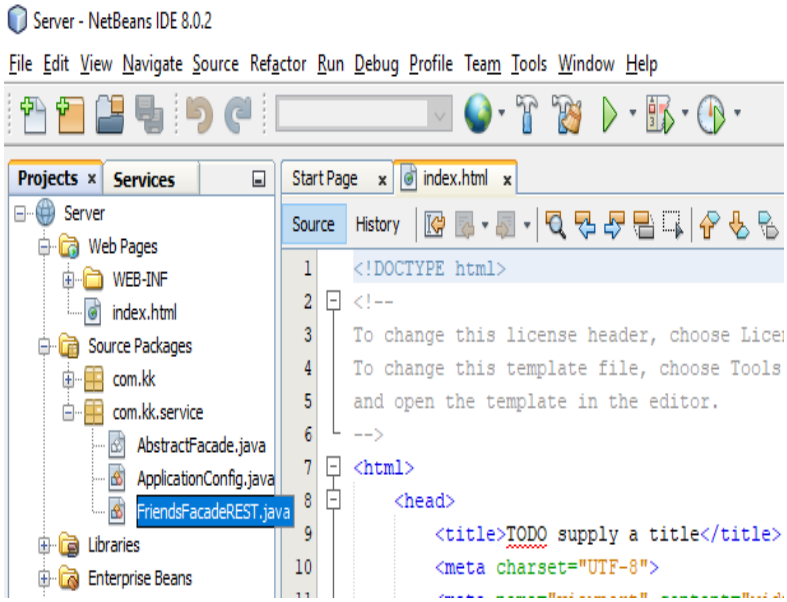
17. Now **select FRIENDS** and **click on Add button**. After that **click on Next** button.



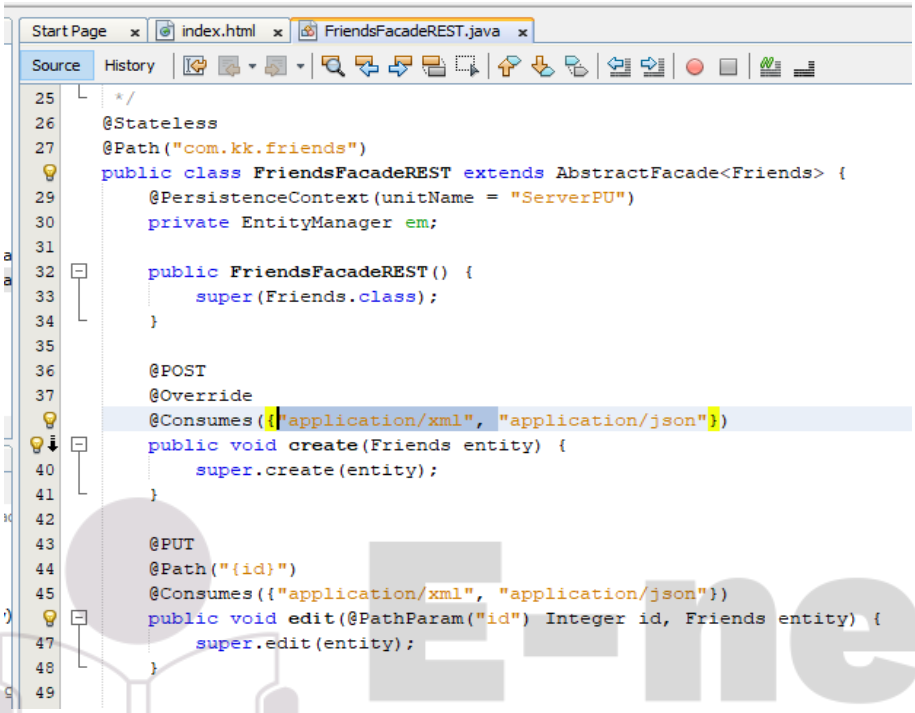
18. Enter Package name as **com.kk** and **click on Next** button and **then Finish**.



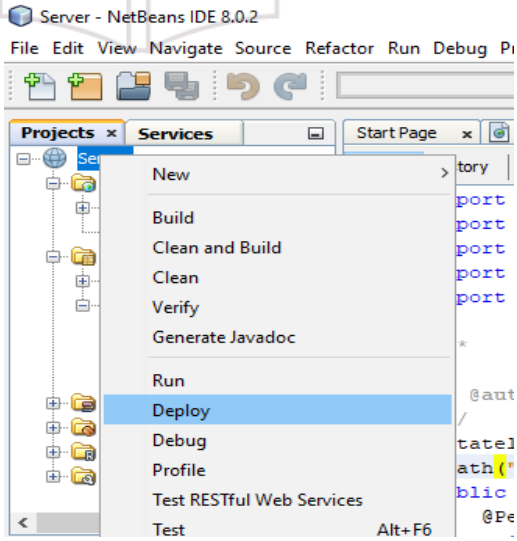
19. Now **open selected file by double click** on it.



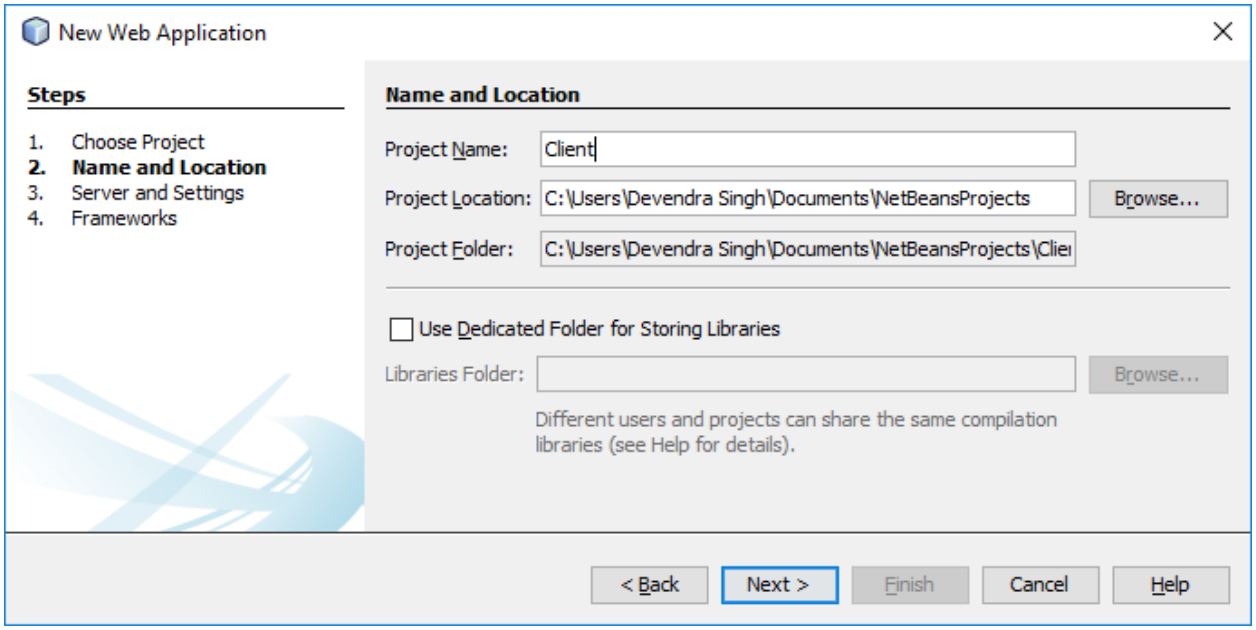
20. Now **remove the selected part from every method in this file**. So that it will communicate only in JSON format. You can also use methods to convert it. But this is easiest method.



21. After that **right click on project name and Deploy** it.



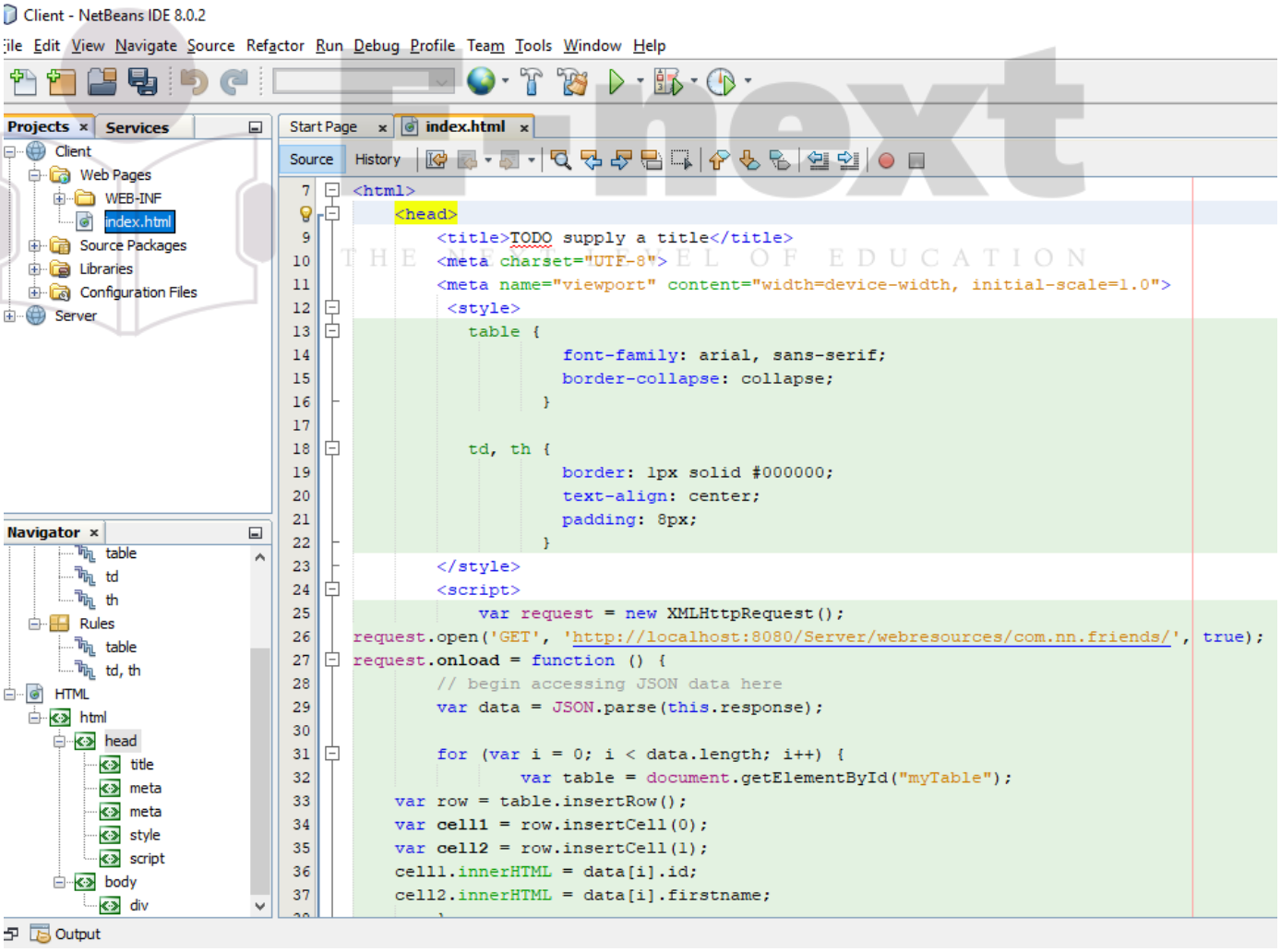
22. Now **create one more Web Application** as **Client**. After that **click on Next** and then **Finish** button.



23. Now **open the index.html file of Client project** and **add the following code in between HEAD tag**.

```
<style>
        table {
                font-family: arial, sans-serif;
                border-collapse: collapse;
               }

        td, th {
                border: 1px solid #000000;
                text-align: center;
                padding: 8px;
               }
      </style>
      <script>
         var request = new XMLHttpRequest();
request.open('GET',
'http://localhost:8080/Server/webresources/com.kk.friends/', true);
request.onload = function () {
    // begin accessing JSON data here
    var data = JSON.parse(this.response);
```

```
    for (var i = 0; i < data.length; i++) {
            var table = document.getElementById("myTable");
   var row = table.insertRow();
   var cell1 = row.insertCell(0);
   var cell2 = row.insertCell(1);
   cell1.innerHTML = data[i].id;
   cell2.innerHTML = data[i].firstname;
    }
};

request.send();
      </script>
```



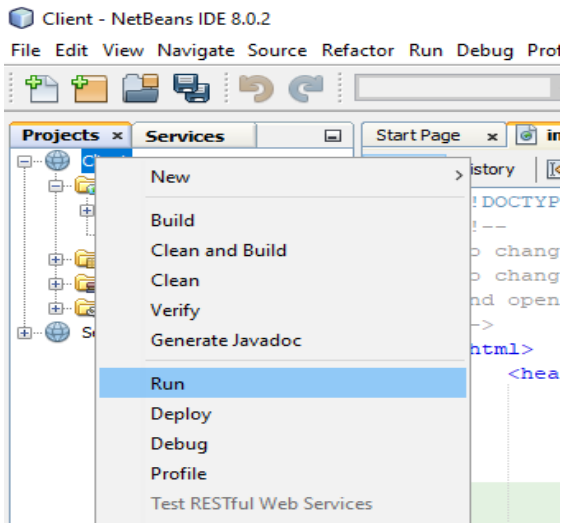
URL in the red font is sensitive. It will change accordingly if you have not given the project and file names as of mine.

24. Replace the content of body tag **with following code**.

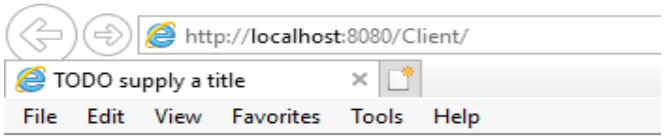
```
<table id="myTable">
        <tr>
          <th> ID</th>
          <th>NAME</th>
        </tr>
      </table>
```

```
        var cell2 = row.insertCell(1);
        cell1.innerHTML = data[i].id;
        cell2.innerHTML = data[i].firstname;
            }
};

request.send();
        </script>
    </head>
    <body>
        <table id="myTable">
            <tr>
                <th> ID</th>
                <th>NAME</th>
            </tr>
        </table>
    </body>
</html>
```

25. Now **run** the **Client** Web Application.

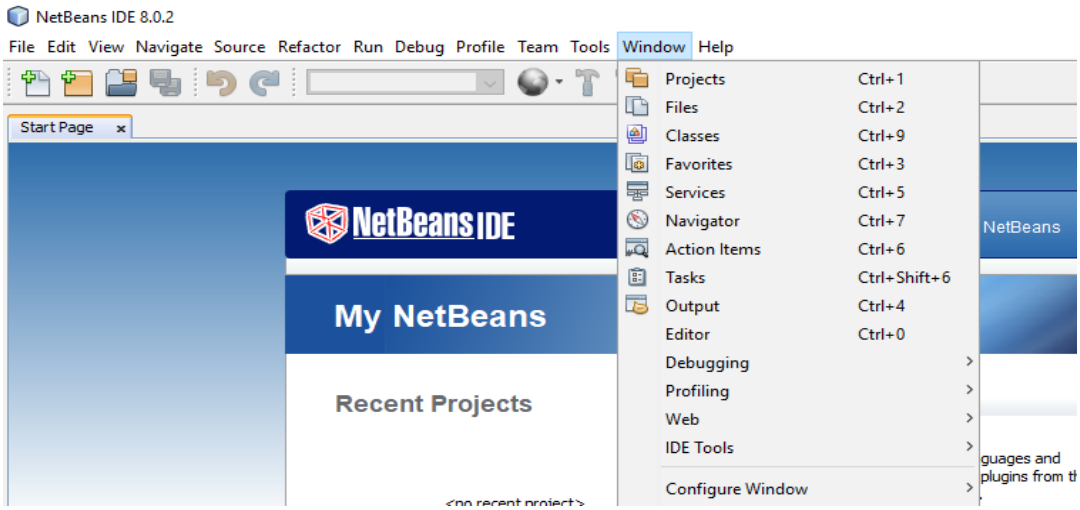


26. A window will open in browser which represents a data in tabular format. These **data are the records entered in FRIEND table**.

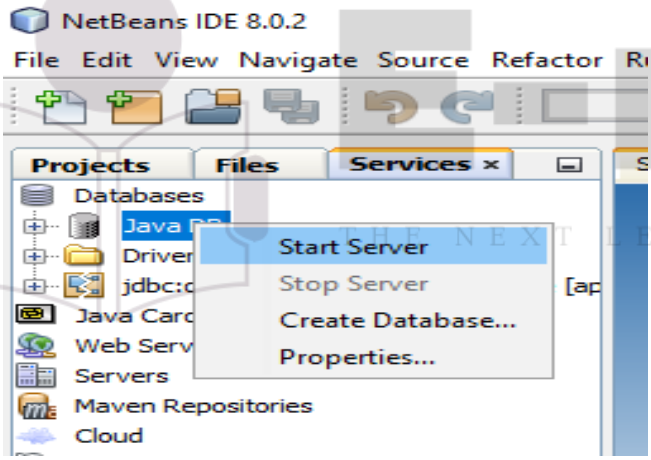


ID	NAME
1	ANAND
2	JULHAS
3	NIKHIL
4	GAGAN
5	RAVI
6	DHARMENDRA
7	ADARSH

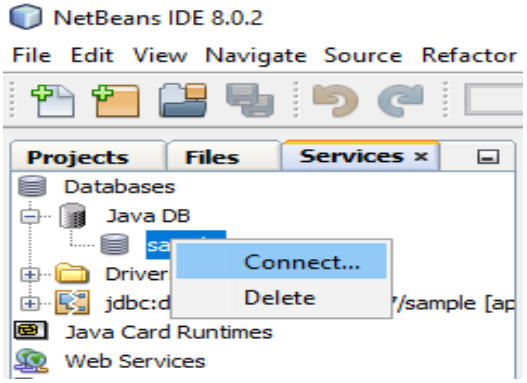
1. Click on Window menu and click on **Projects, Files** & **Services** to open it.



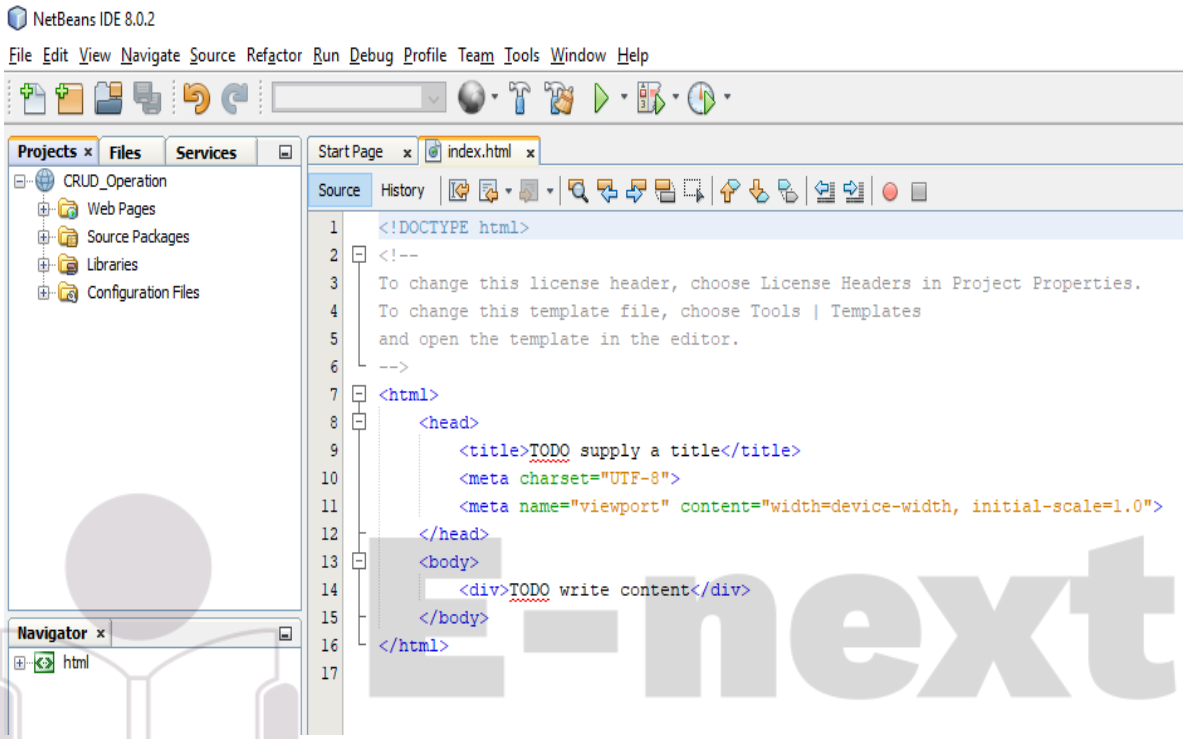
2. **Right click on Java DB** and then **click on Start Server** to start the server .



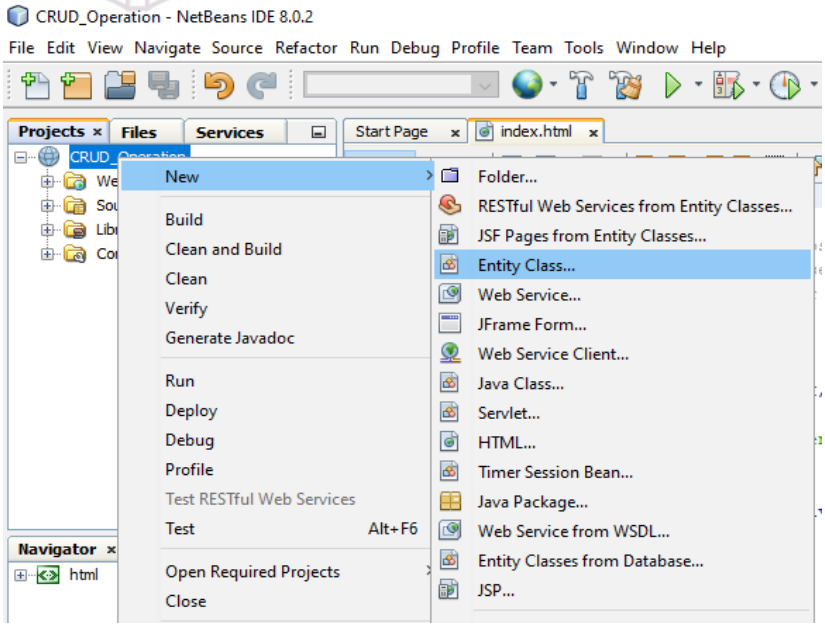
3. Now expand Java DB and **right click on sample** and then **click on connect** to connect the sample database with server.



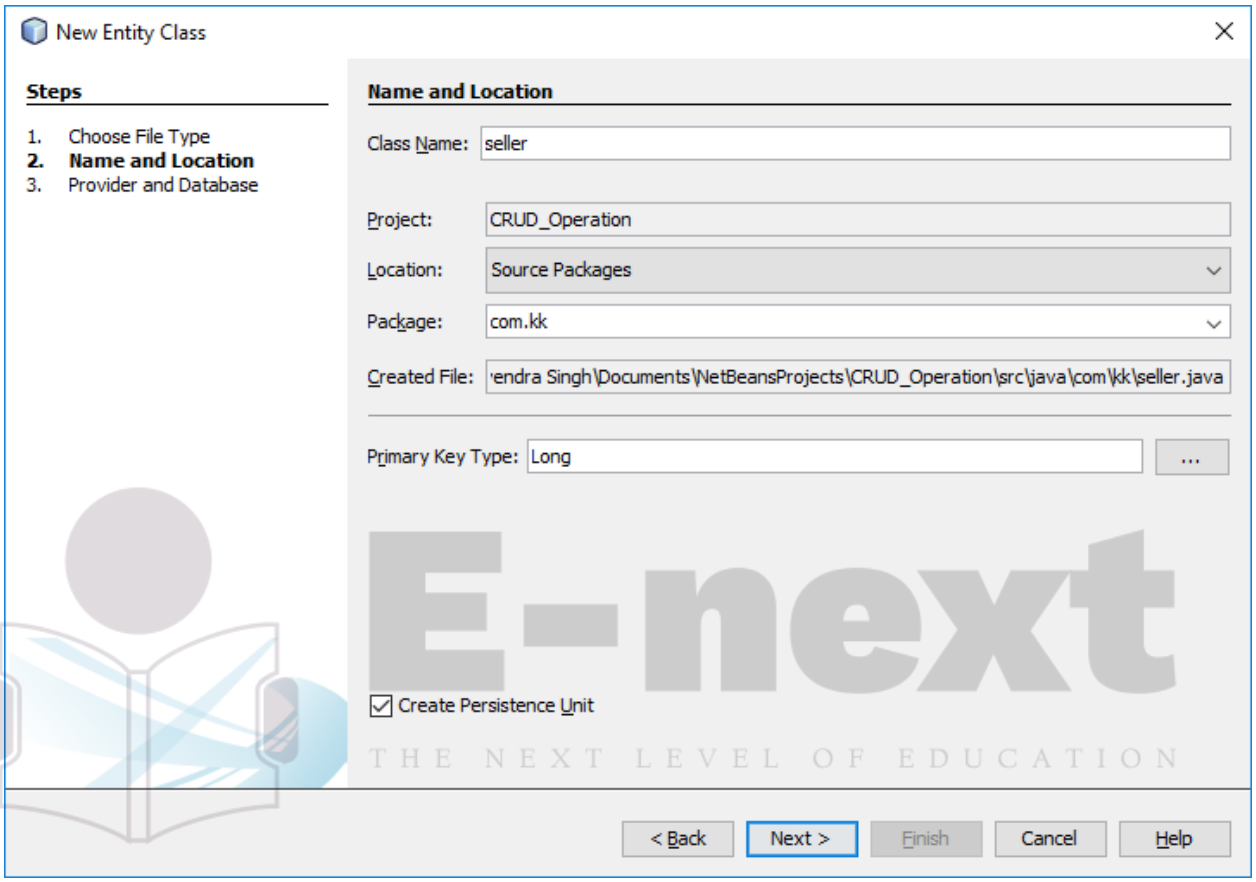
4. Now create a web application with the name **CRUD_Operation**. A window will open like following pic.



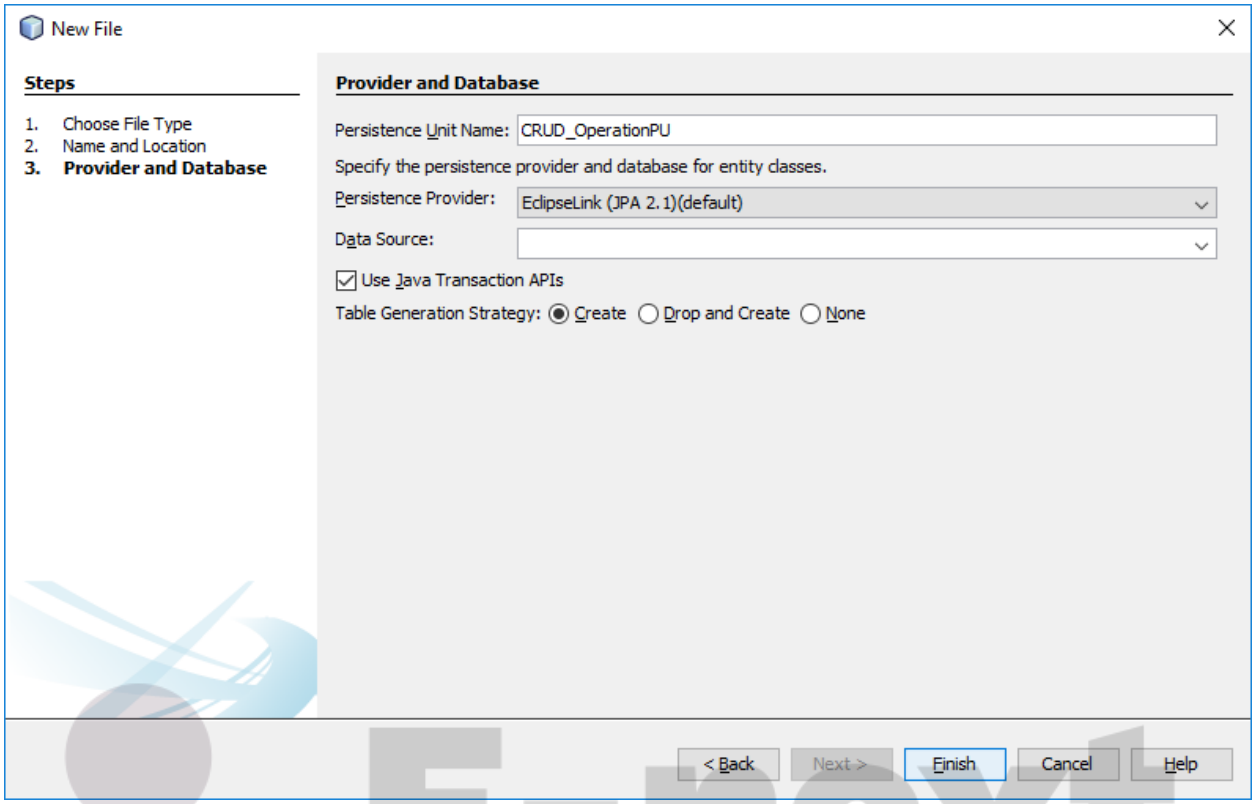
5. Create an entity class. **Right click on project name -> New -> Entity Class.**



6. A window will appear like bellow pic. Enter following data and click on Next ….
   Class Name -> seller
   Package Name -> com.kk

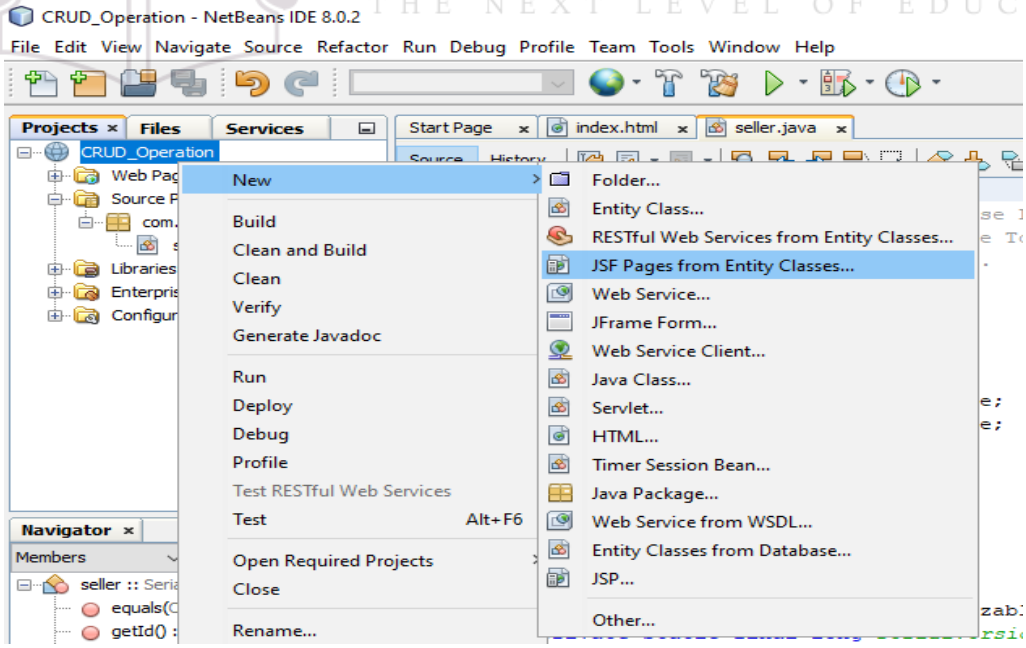


7. **Click on Finish**.

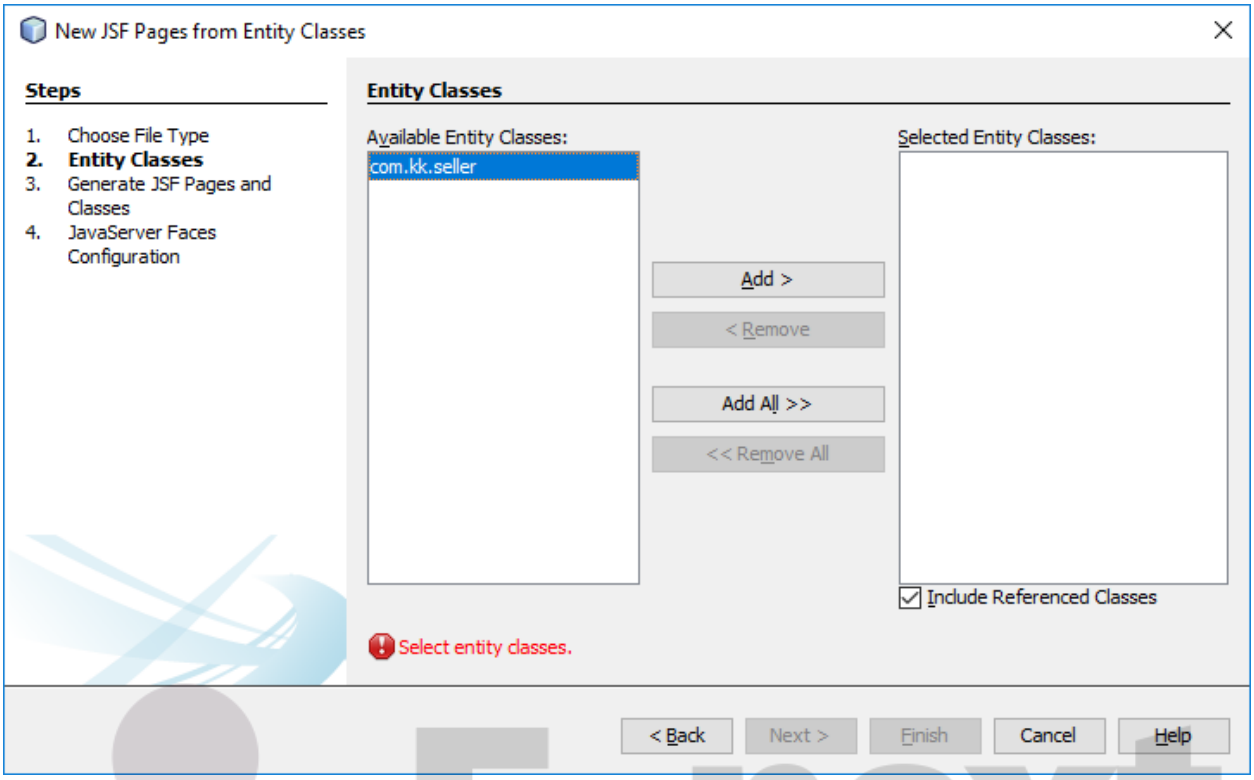


8. Right click on project name and create JSF Pages from Entity Classes.

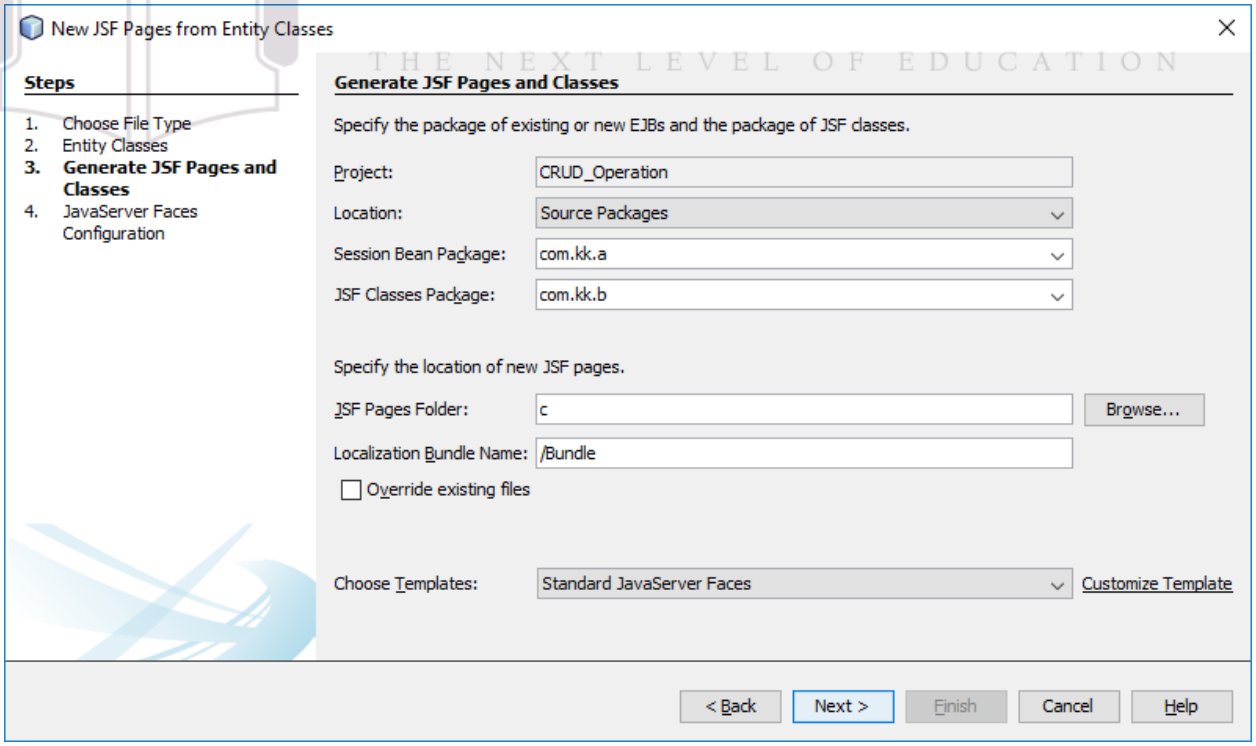
Right click on project name -> New -> JSF Pages from Entity Classes



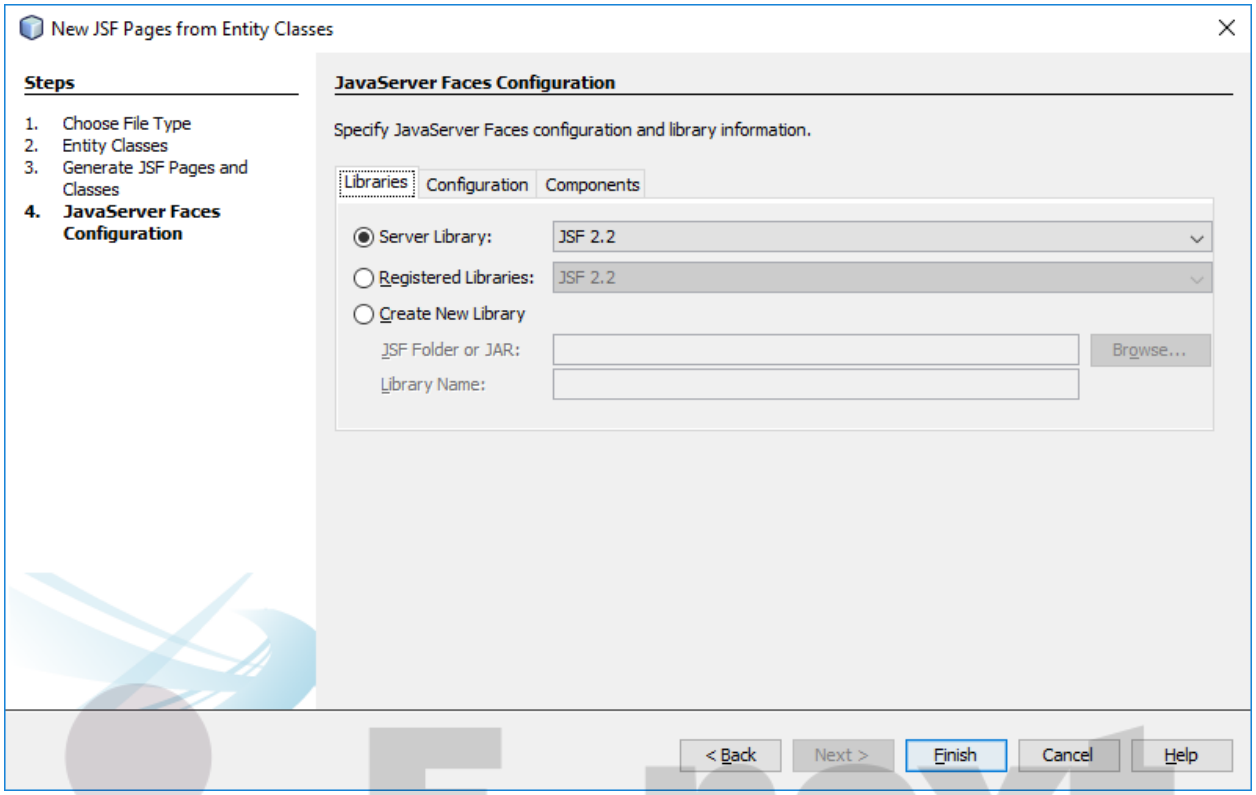
9. **Select com.kk.seller** and **click on Add button and then Next button** on below.



10. A window like below will appear on the screen. **Enter the data into that window as entered in below pic** and **click on Next button**.

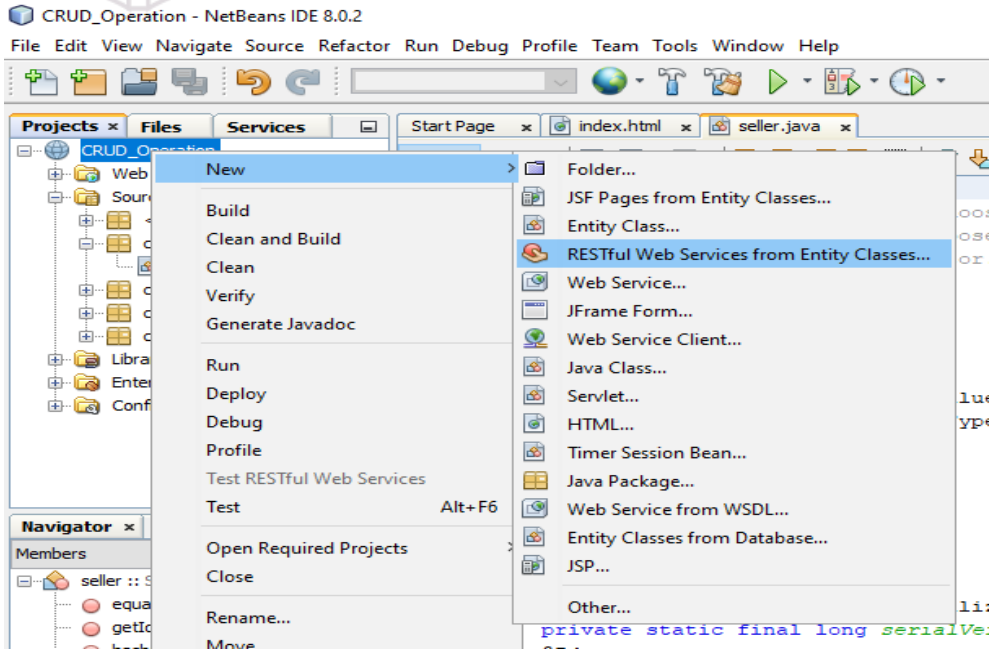


11. Now **click on Finish**.

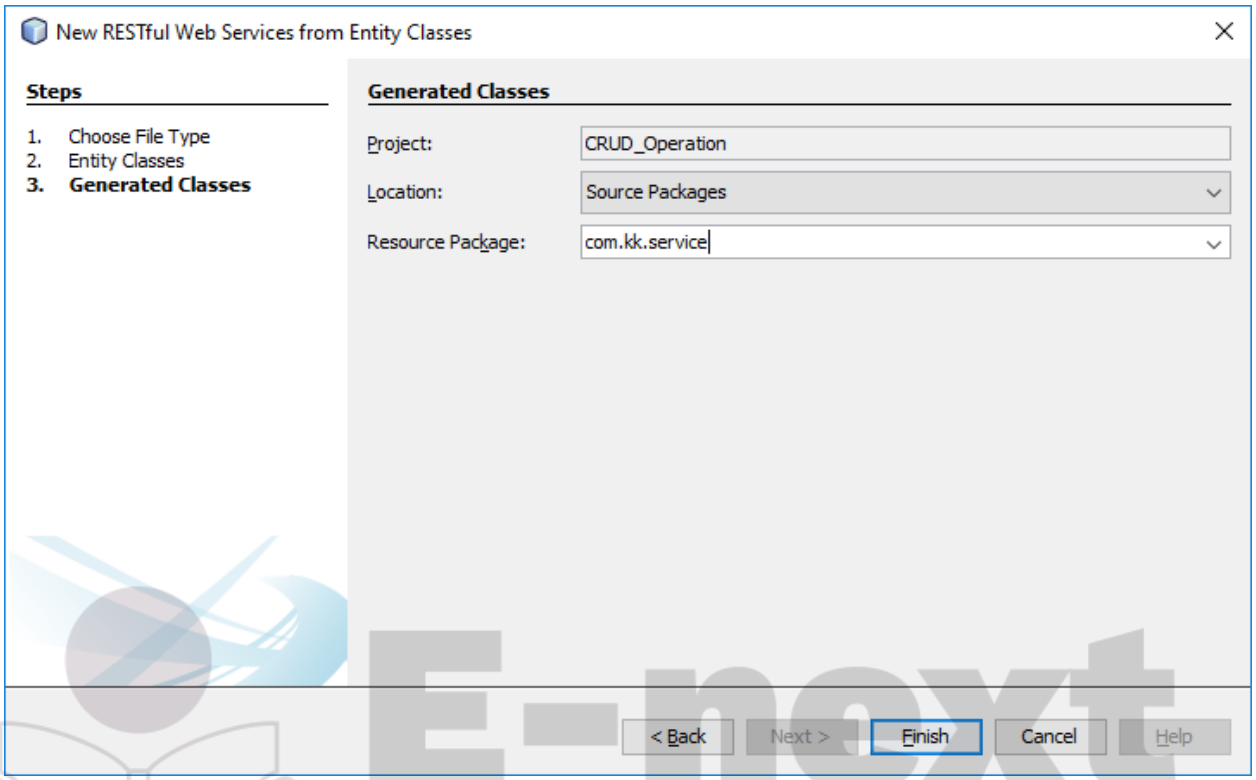


12. Right click on project name and create RESTful Web Services from Entity Classes.

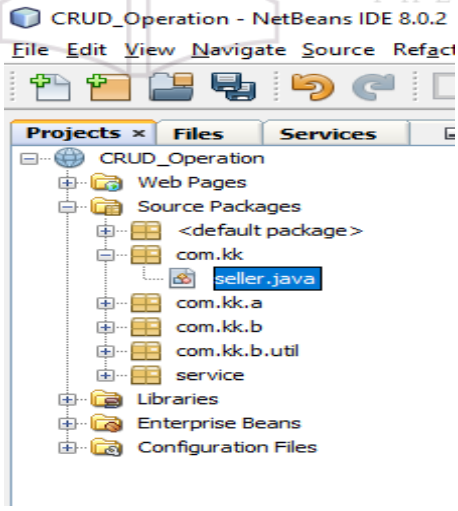
Right click on project name -> New -> RESTful Web Services from Entity Classes



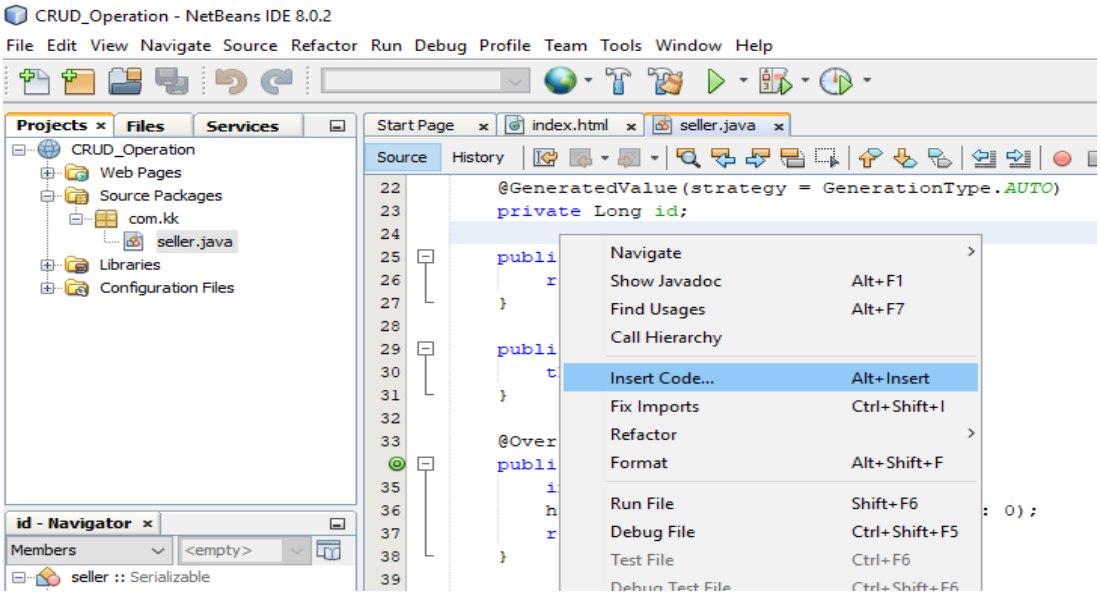
13. Repeat step 9 and then it will go on next page. Then **enter the com.kk.service** in Resource Package and then **click on Finish button**.



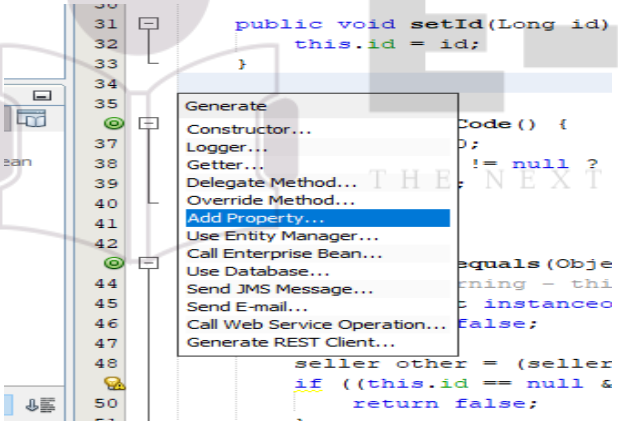
14. Now **open seller.java file under com.kk package**.



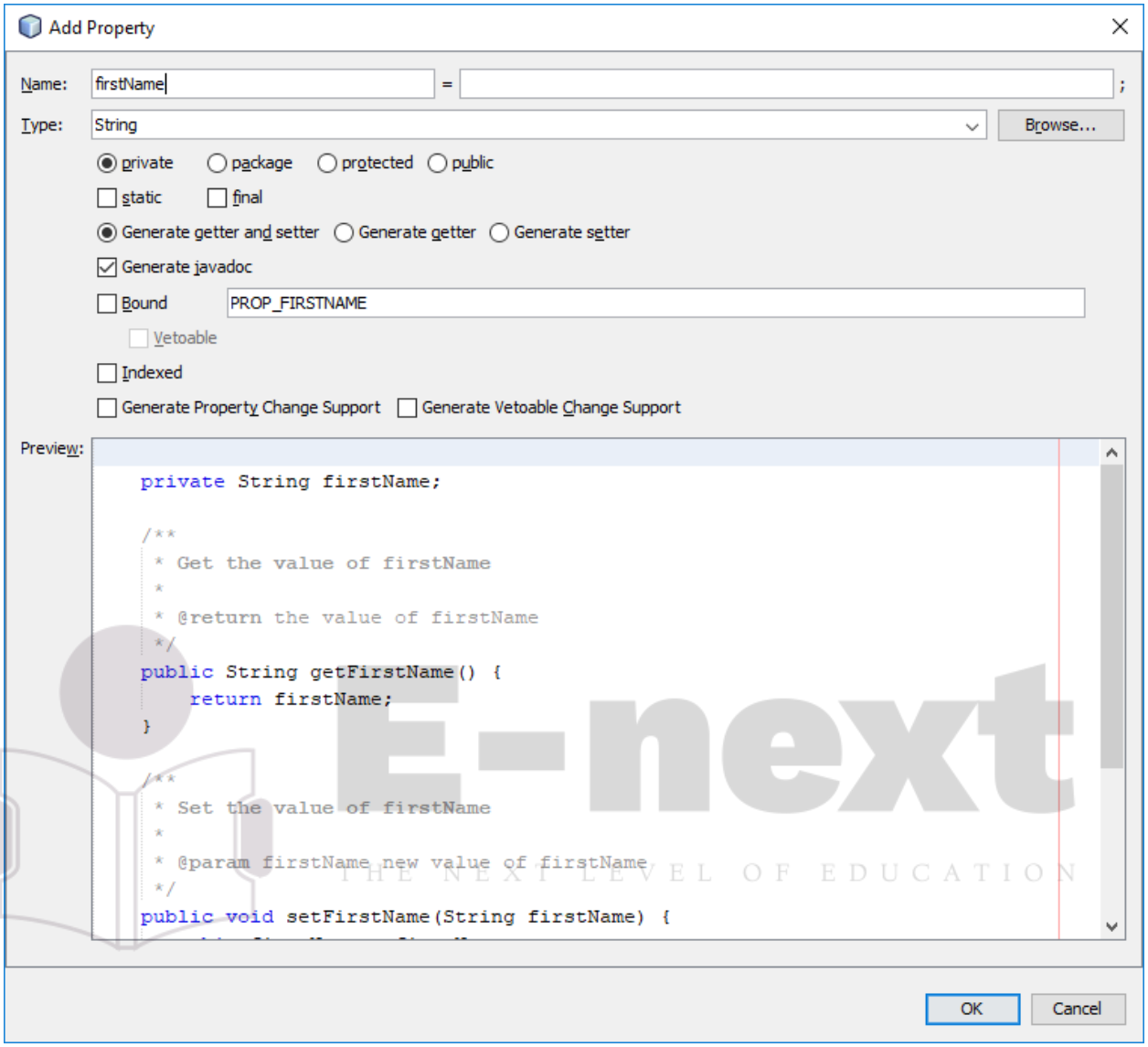
15. In this file **at line number 24**, do the **right click and select Insert Code**.



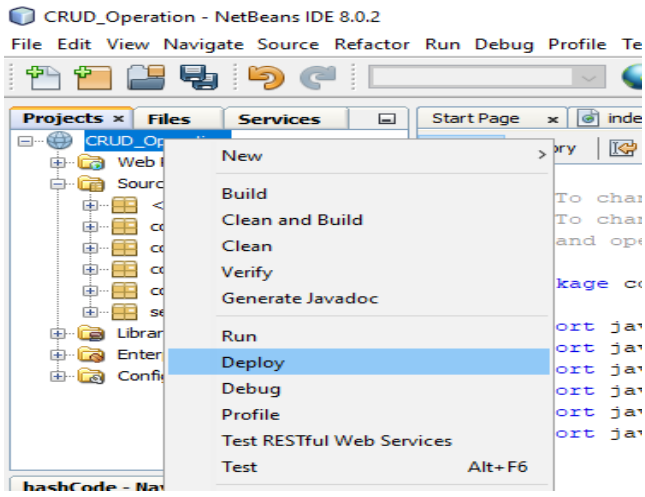
16. A new list will appear. **Click on Add Property**.



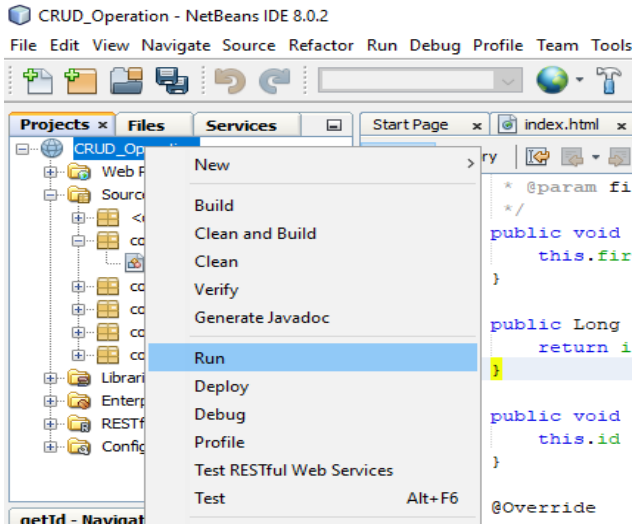
17. A new window will open. Enter name as **firstName**. Make sure name should be exact same as of mine and then **click on OK button**. Actually we are setting getter and setter method for firstName.



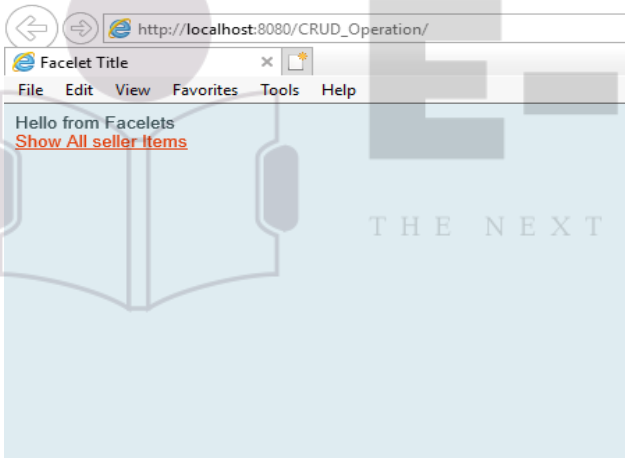
18. Now **right click on web application name and Deploy** it.



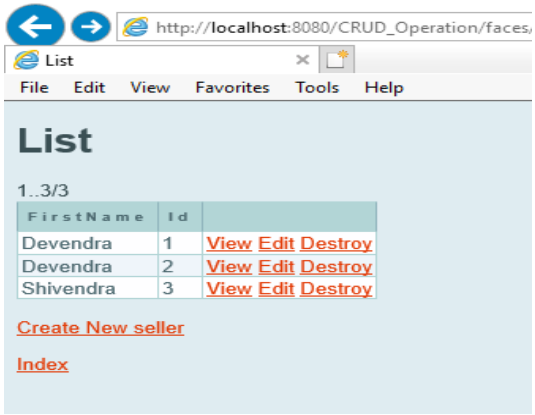
19. Now **right click on project name** and **run** it.



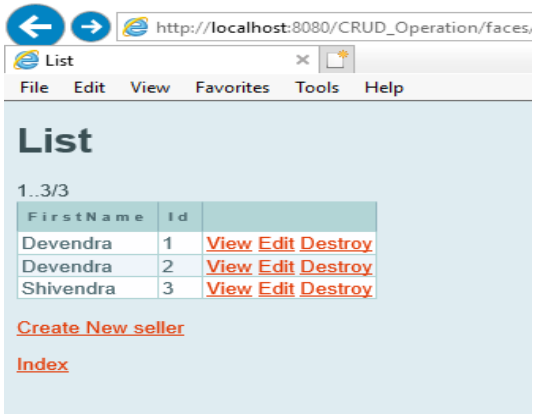
20. A window will open in browser like below….



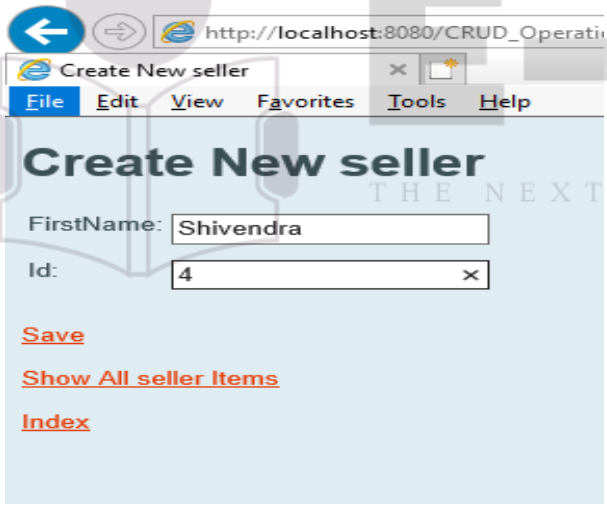
21. Now **click on Show All sellers** Items for CRUD operation.



22. As I have added three data into database. **You can add more data by click on Create New seller and can view, edit and delete by click on View, Edit and Destroy option**.



23. Adding one more data into database for demo. Just click on Create New seller. **Enter a name into FirstName and id into Id**. Now **click on Save** option to save the data.



24. Now **click on Show All seller Items to view all records** whether our data is entered or not. We can see that one more Shivendra name is appearing.