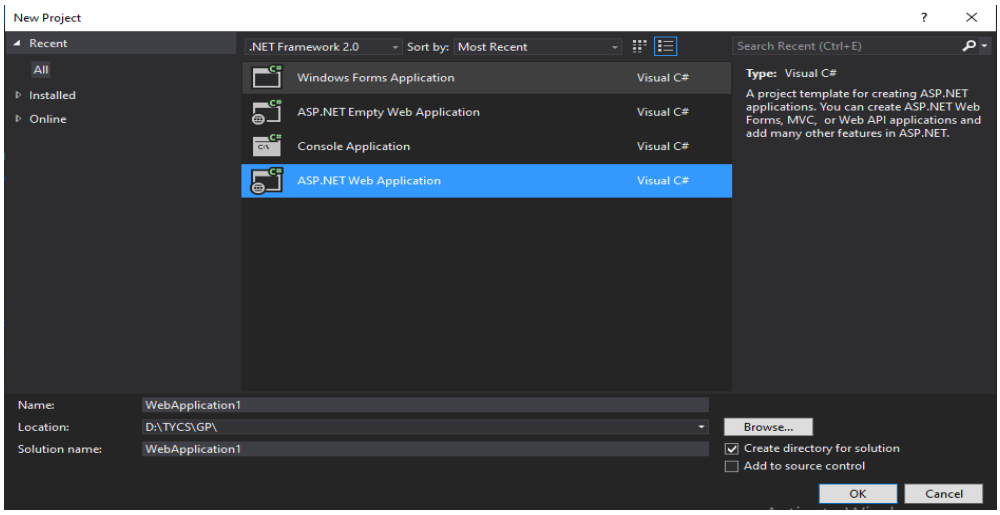
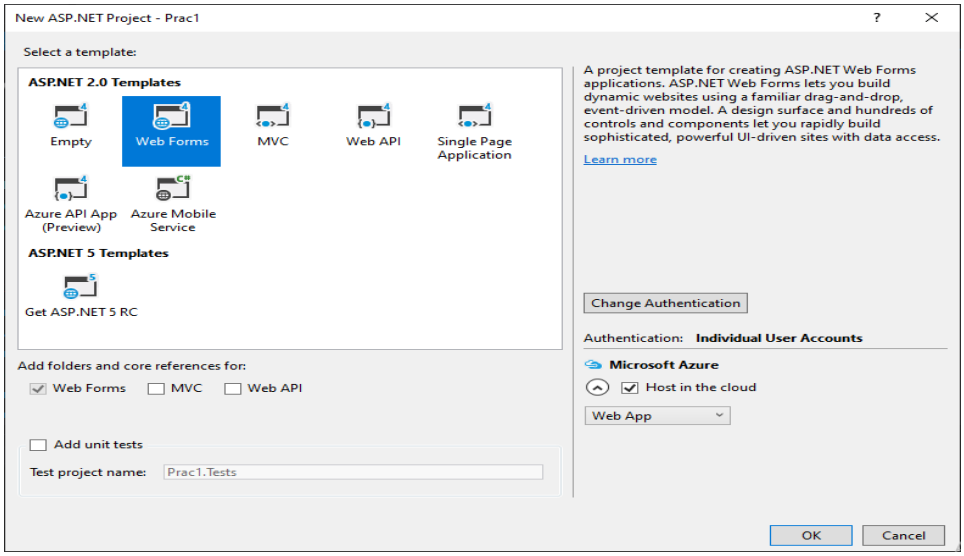
Practical 1 - A

Aim: Write a program to implement to create a simple web service that converts the temperature from Fahrenheit to Celsius and vice a versa

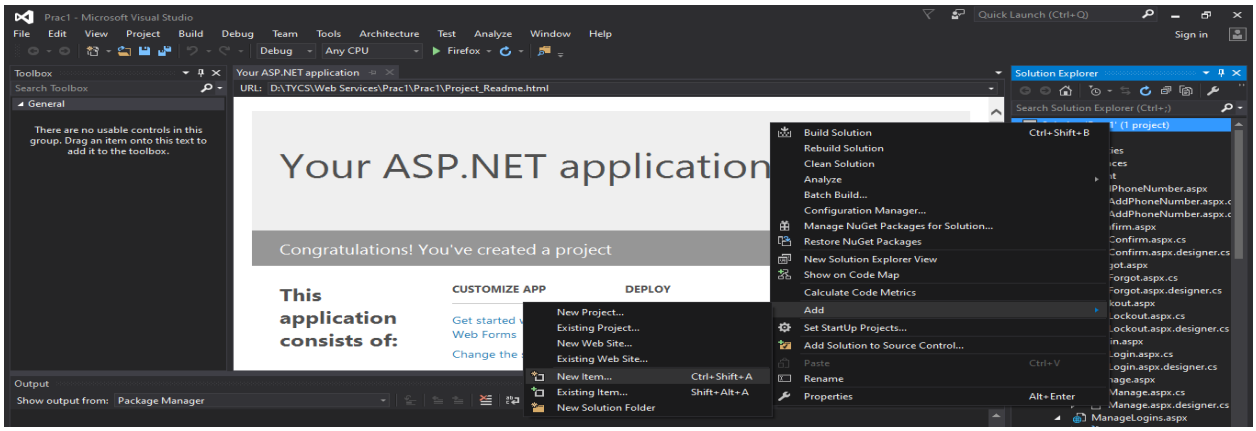
Make a new Project → ASP.NET.Web Application



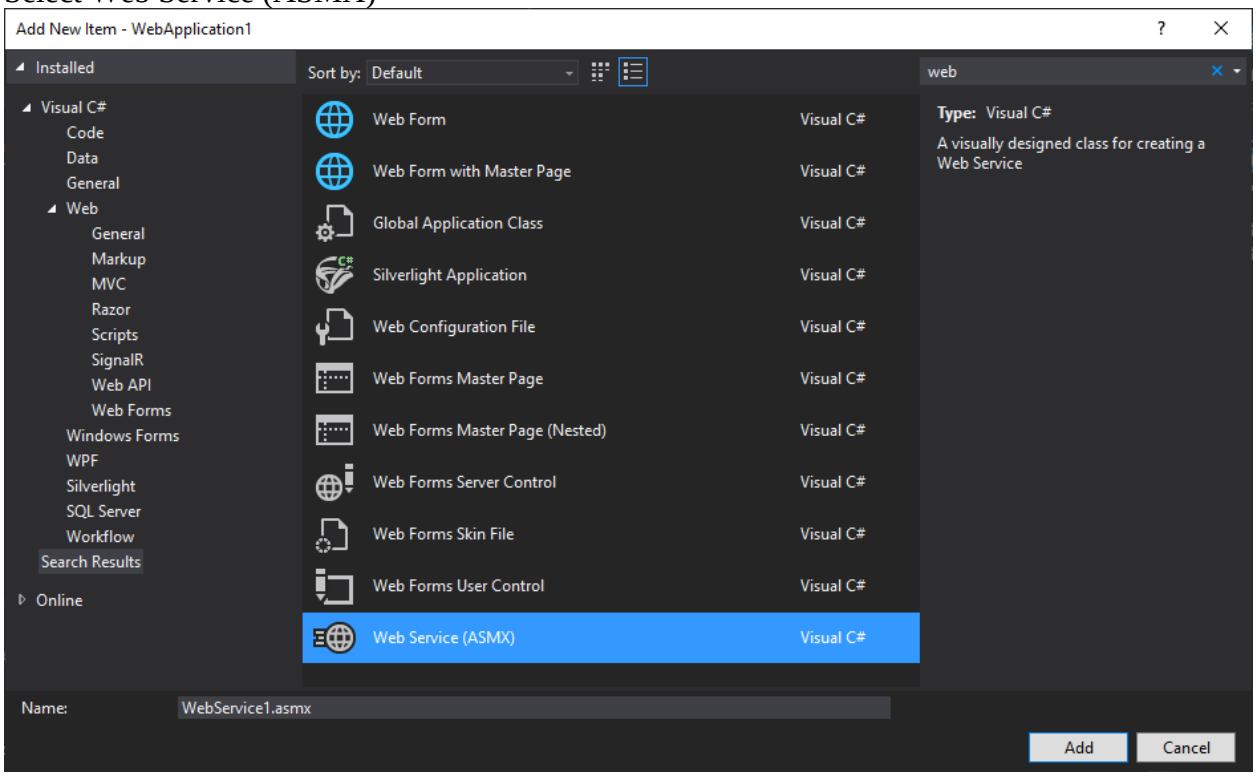
Select Web Forms



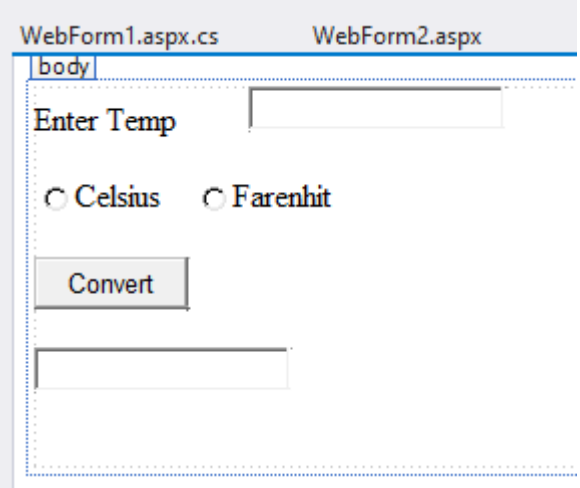
Add → New item



Select Web Service (ASMX)



Design



Source Code

Html Design Code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Prac1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body style="height: 285px">
```

```
    <form id="form1" runat="server">
    <div style="margin-left: 120px">
        <asp:Label ID="Label1" runat="server" Text="Enter Temp:"></asp:Label>
          
        <asp:TextBox ID="TextBox1" runat="server"
OnTextChanged="TextBox1_TextChanged"></asp:TextBox>
        <asp:RadioButton ID="RadioButton1" runat="server" GroupName="fc"
OnCheckedChanged="RadioButton1_CheckedChanged" Text="Fahrenheit" />
        <asp:RadioButton ID="RadioButton2" runat="server" GroupName="fc"
OnCheckedChanged="RadioButton2_CheckedChanged" Text="Celsius" />
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Convert" />
        <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
        <br />
    </div>
    </form>
</body>
</html>
```

WebForm1.aspx.cs

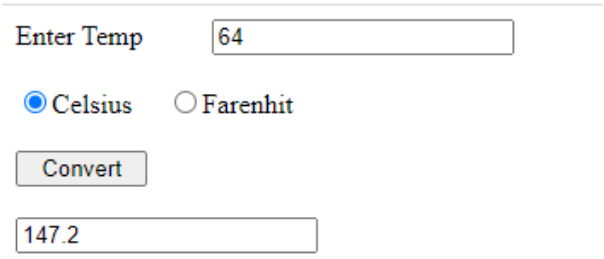
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Prac1
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void TextBox1_TextChanged(object sender, EventArgs e)
        {

        }

        protected void RadioButton1_CheckedChanged(object sender, EventArgs e)
        {

        }

        protected void RadioButton2_CheckedChanged(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
```

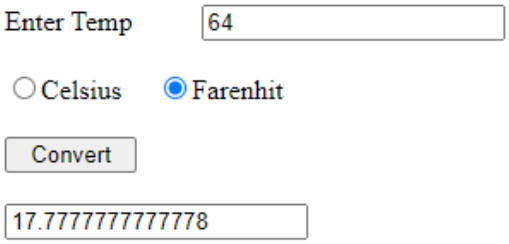
```
        {
            WebService1 w = new WebService1();

            if (RadioButton1.Checked)
            {
                Label2.Text = w.ctof(Convert.ToDouble(TextBox1.Text)).ToString();
            }
            else if (RadioButton2.Checked)
            {
                Label2.Text = w.ftoc(Convert.ToDouble(TextBox1.Text)).ToString();
            }
        }
    }
}
```

WebService1.asmx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace Prac1
{
    /// <summary>
    /// Summary description for WebService1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the
following line.
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {

        [WebMethod]
        public double ctof(double c)
        {
            return ((c * 1.8) + 32);
        }

        [WebMethod]
        public double ftoc(double f)
        {
            return ((f - 32) / 1.8);
        }
    }

}
```

Output


Enter Temp
64
Celsius
Farenhit
Convert
147.2


Enter Temp
64
Celsius
Farenhit
Convert
17.7777777777778

Practical 1 - B

Aim: Write a program to implement to create a simple web service that calculates the square of a number

Design

Source Code

Html Design Code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication1.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div style="height: 287px">
     
      <asp:Label ID="Label1" runat="server" style="font-weight: 700" Text="Enter the
Number:"></asp:Label>
           
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
             &nb
sp;         
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Calculate" />
  <strong>
      <asp:Label ID="Label2" runat="server" Text="Square :"></asp:Label>
     
</strong>            &
nbsp;    
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>

   </div>
   </form>
</body>
</html>
```

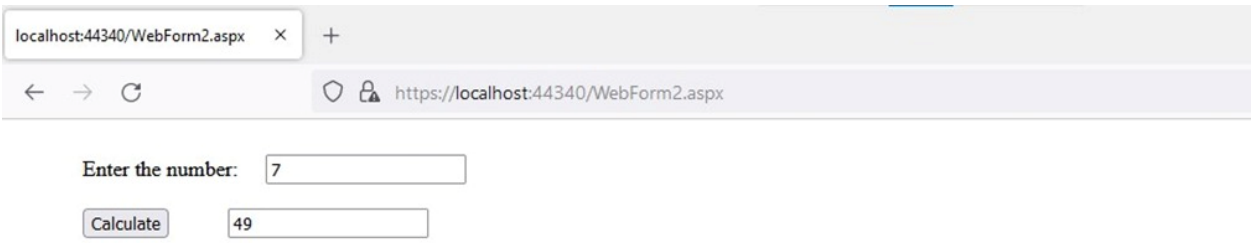
WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication1
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            WebService1 w = new WebService1();

            TextBox2.Text = w.square(Convert.ToDouble(TextBox1.Text)).ToString();
        }
    }
}
```

WebService1.asmx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace WebApplication1
{
    /// <summary>
    /// Summary description for WebService1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the
following line.
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {
        [WebMethod]
        public double square(double a)
        {
            return (a * a);
        }
    }
}
```

Output


localhost:44340/WebForm2.aspx
https://localhost:44340/WebForm2.aspx
Enter the number:
7
Calculate
49

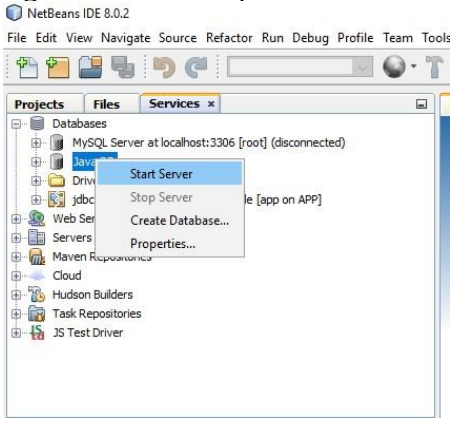
Practical 2

AIM: Write a program to implement the operation can receive request and will return a response in two ways.

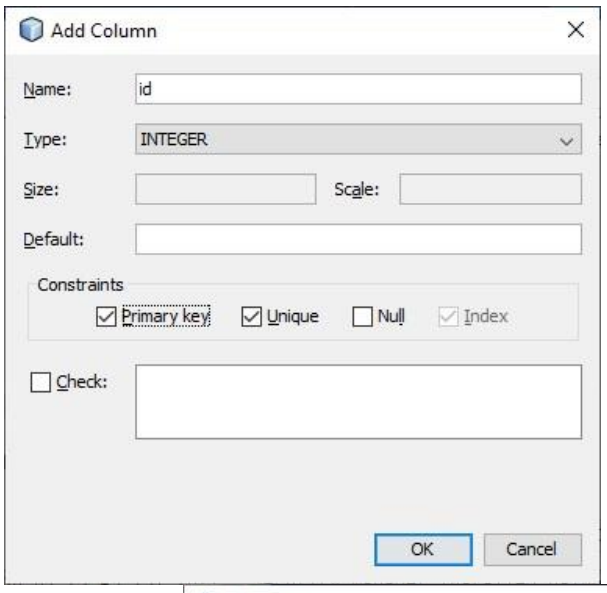
One - Way operation

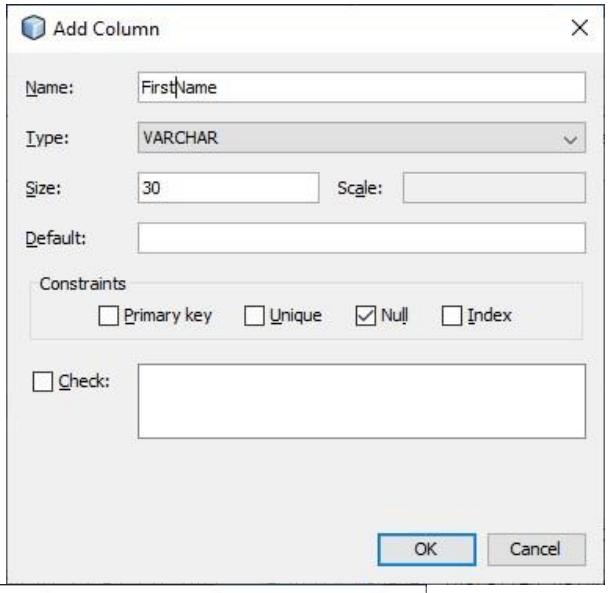
Request –Response

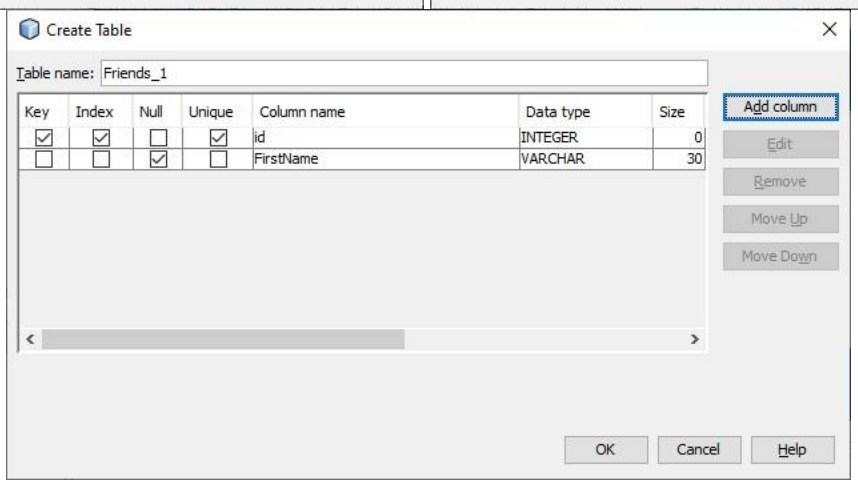
STEP 1: Go to Services – Databases – Java DB right click and Start Server. Expand Java DB and right click on Sample and Connect



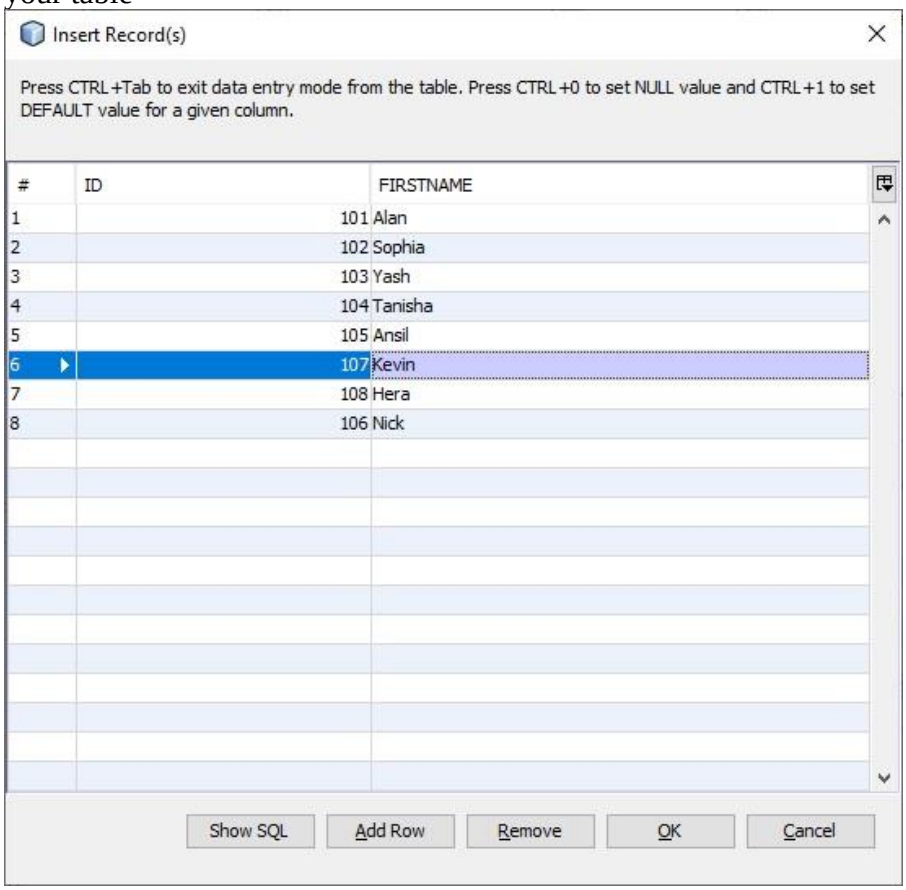
STEP 2: Expand jdbc:derby://localhost:1527/sample [app on APP] right click on Tables and Create Table. Name it and add columns



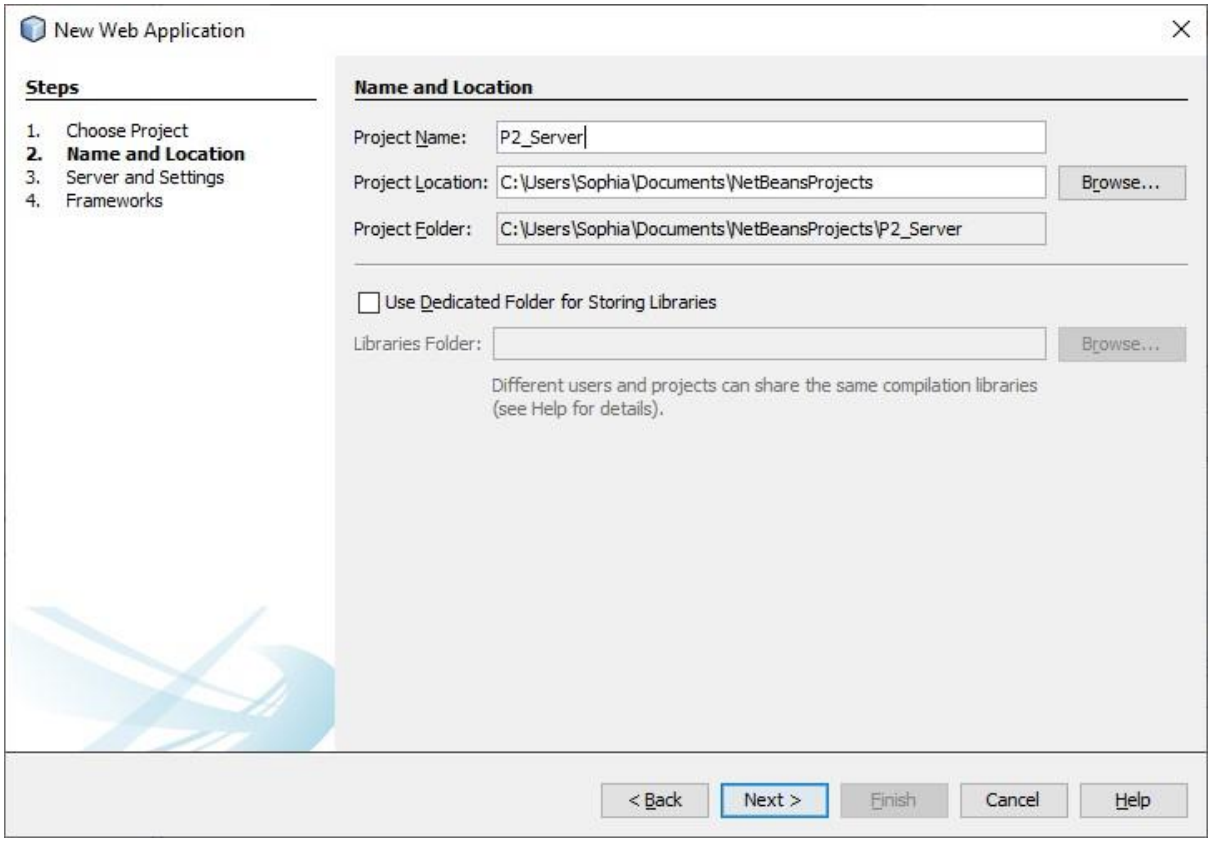




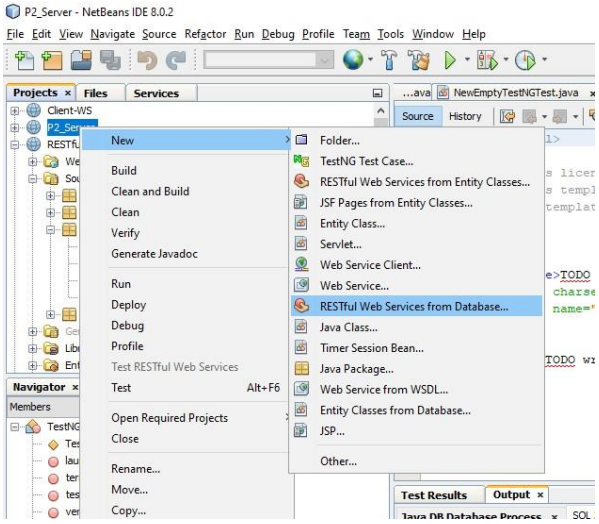
STEP 3: Expand Tables and right click on Friends_1 click View Data and here insert records into your table



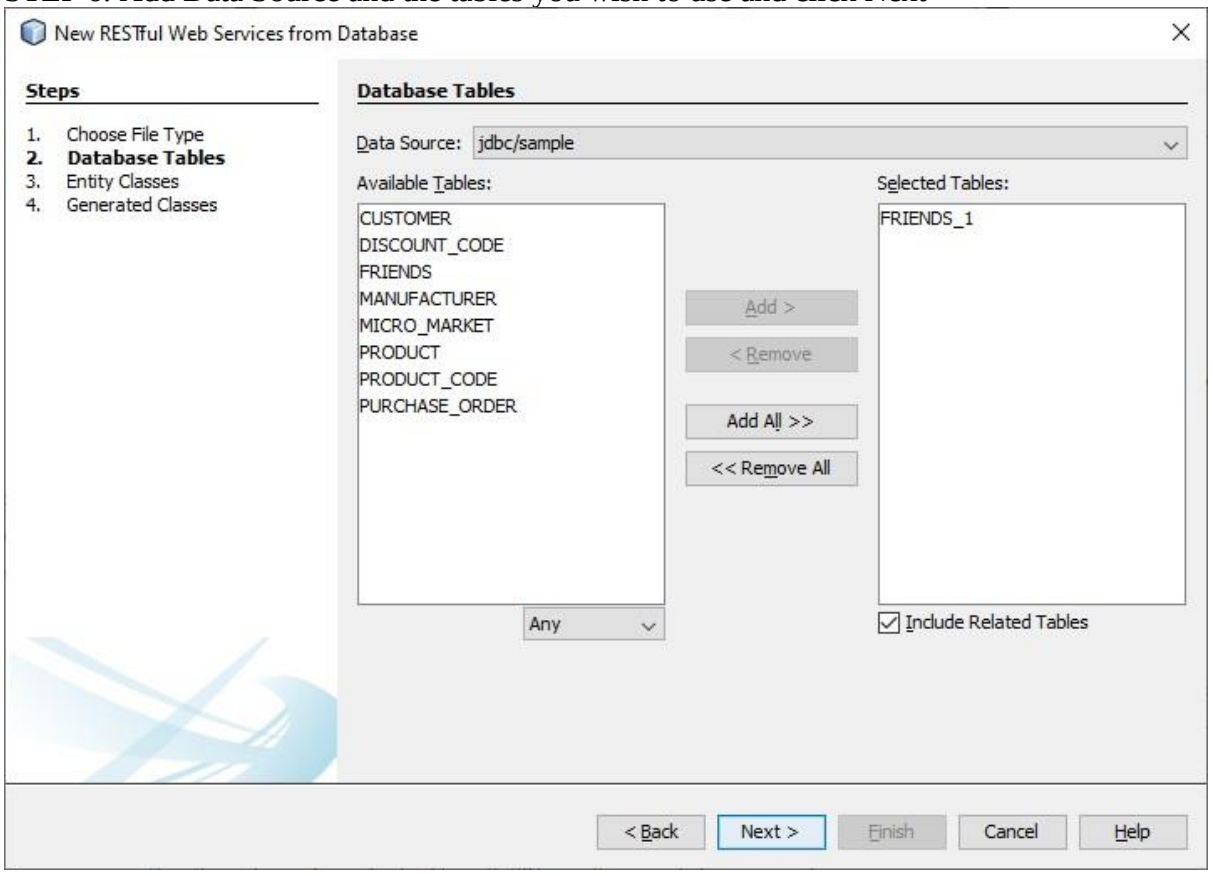
STEP 4: Create a New Web Application Project. Name it as per your choice click on Next and Finish



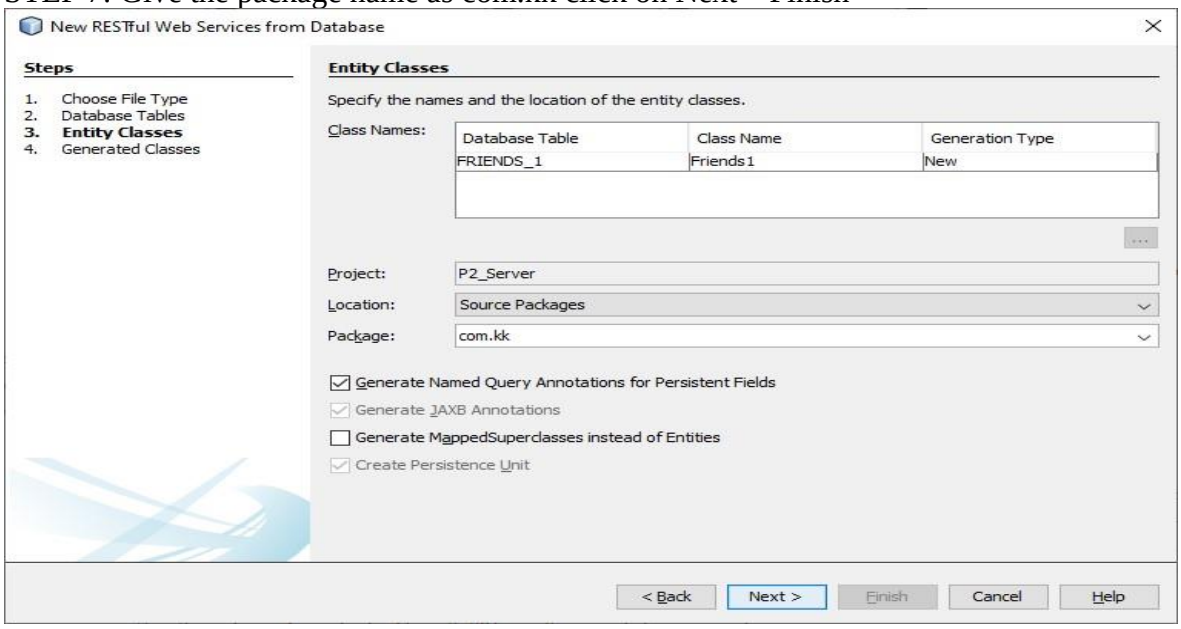
STEP 5: Right click on the newly created project and Add – New – RESTful Web Service from Database



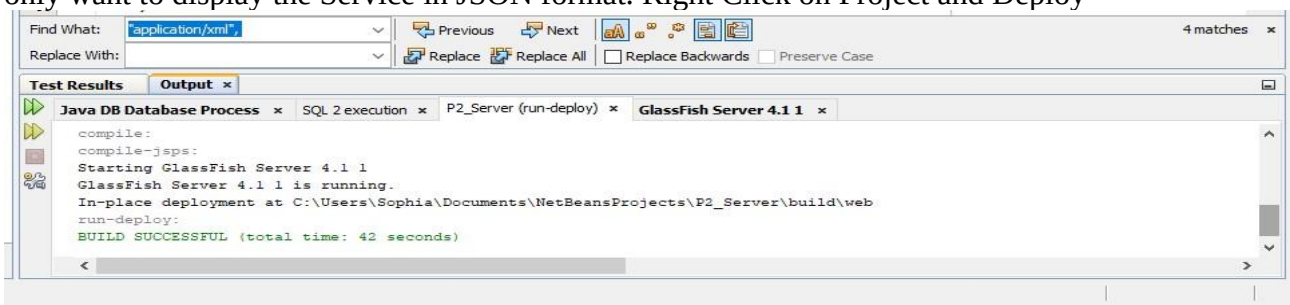
STEP 6: Add Data Source and the tables you wish to use and click Next



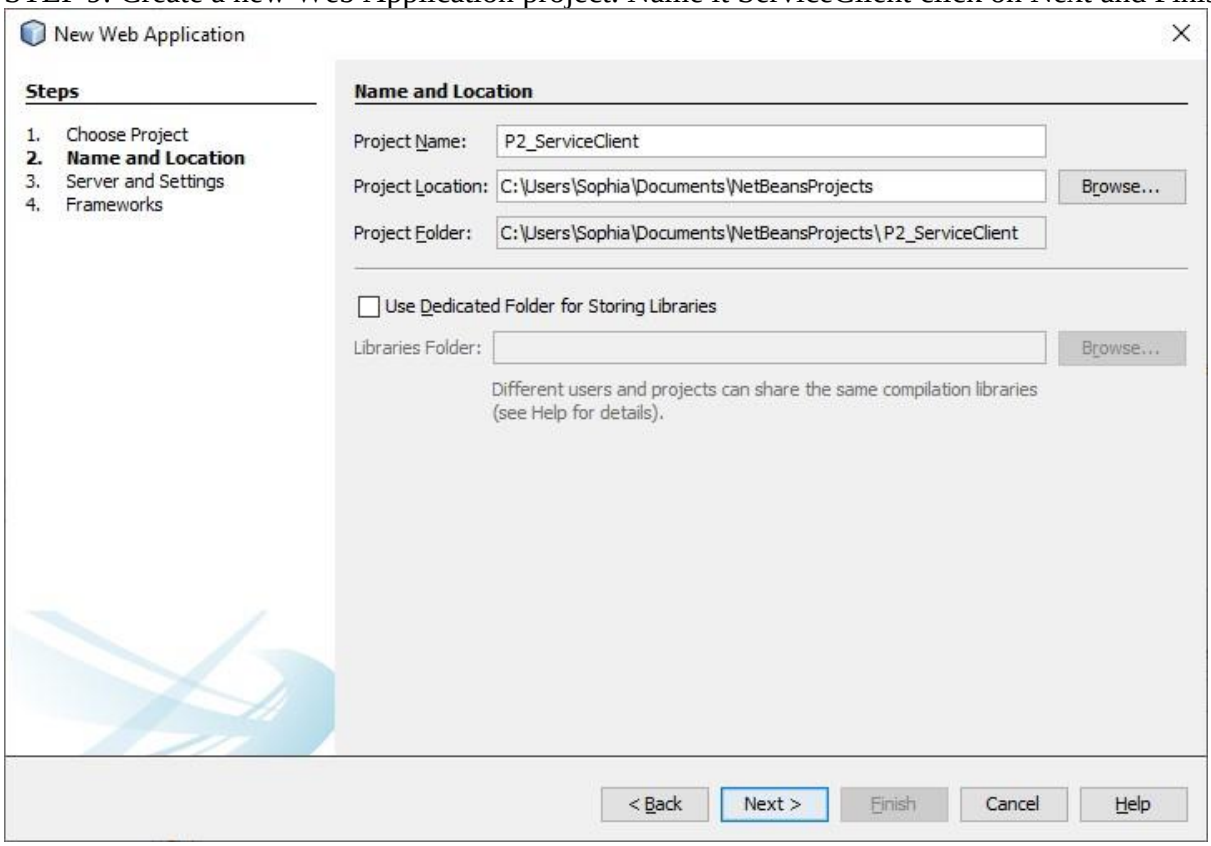
STEP 7: Give the package name as com.kk click on Next – Finish



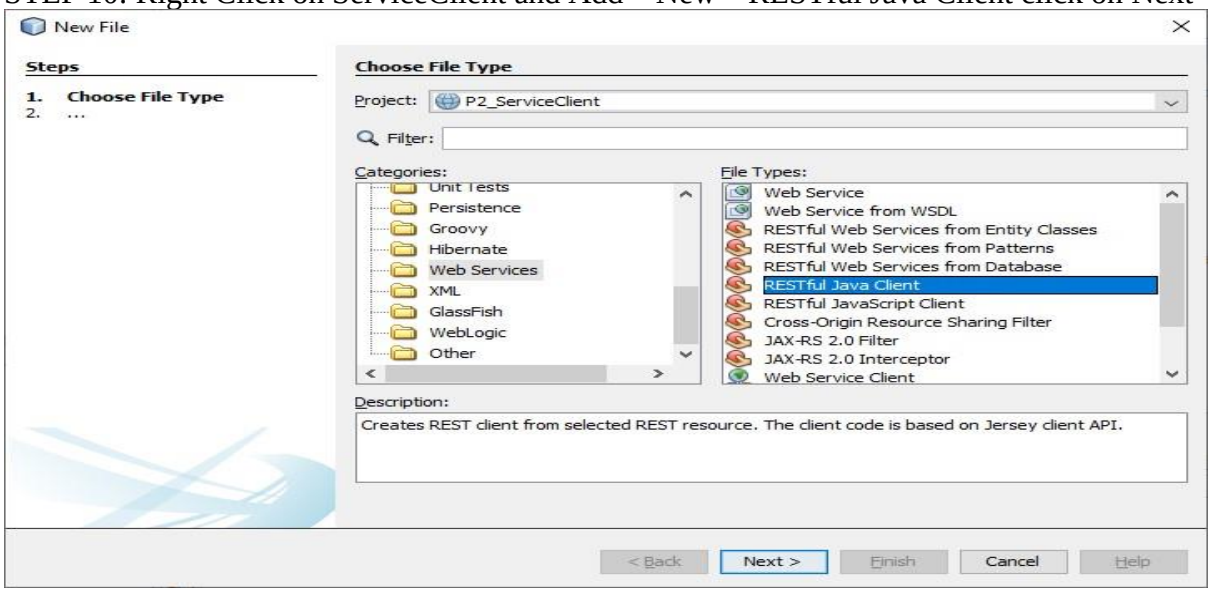
STEP 8: Open Friends1FacadeREST.java and remove all "application/xml", from the code as we only want to display the Service in JSON format. Right Click on Project and Deploy



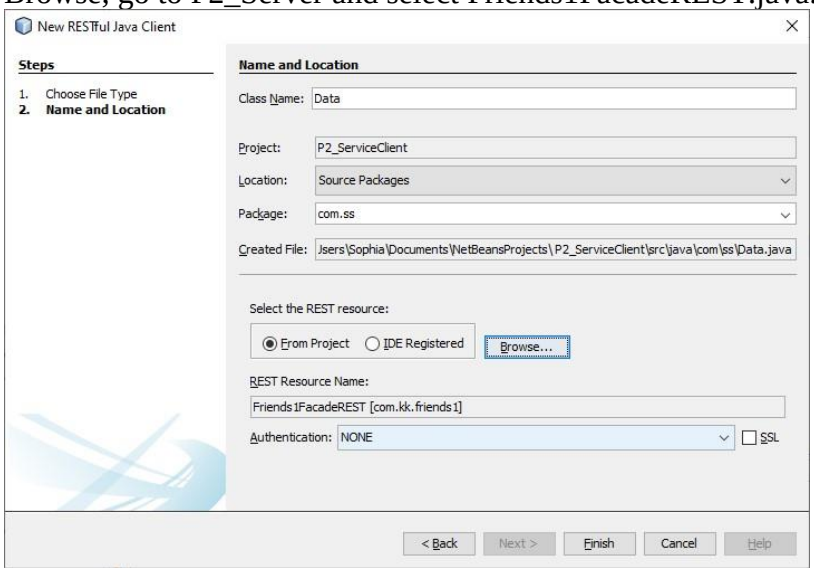
STEP 9: Create a new Web Application project. Name it ServiceClient click on Next and Finish



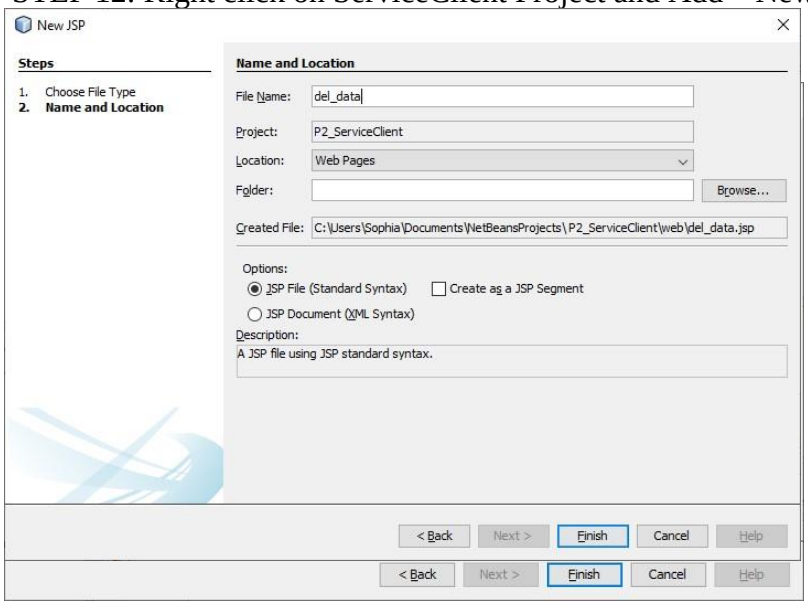
STEP 10: Right Click on ServiceClient and Add – New – RESTful Java Client click on Next



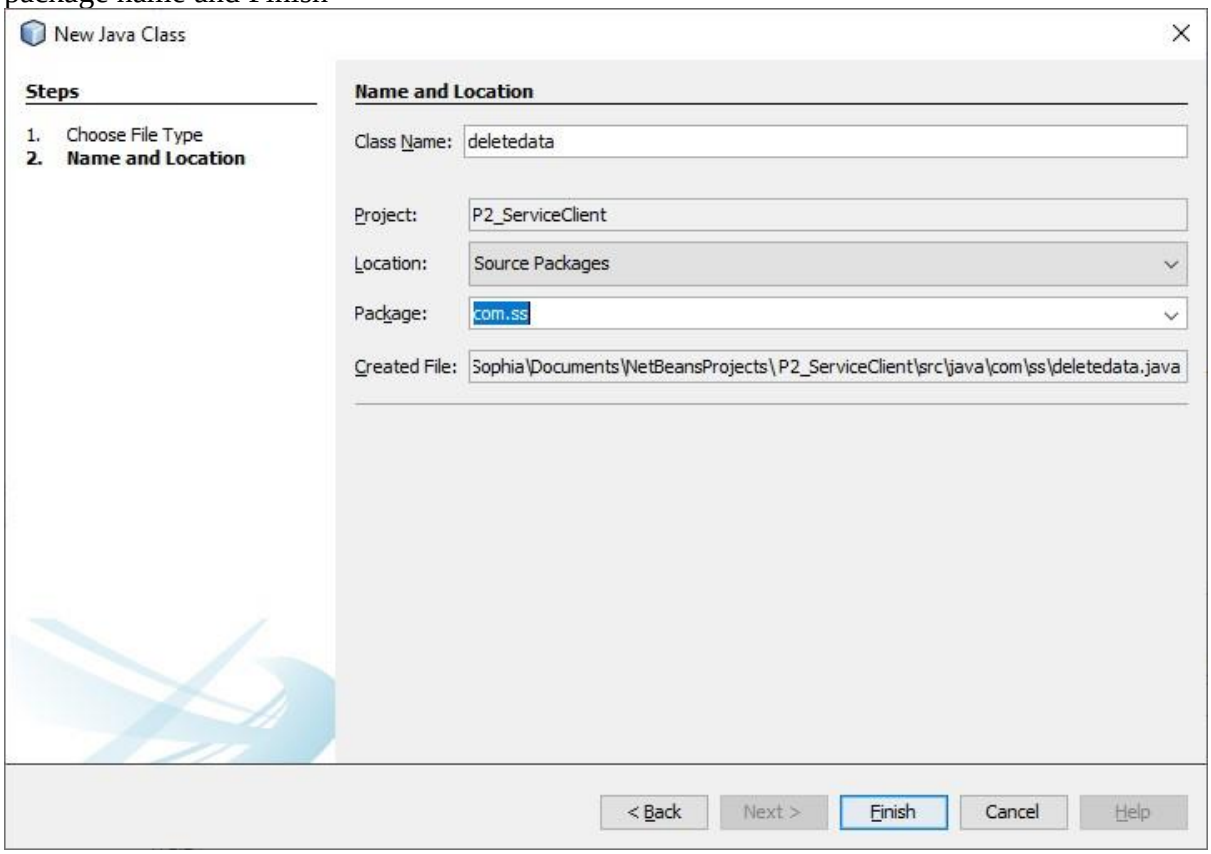
STEP 11 Give your Class Name and set Package as com.ss. Choose REST Resource Name. Click on Browse, go to P2_Server and select Friends1FacadeREST.java.Click on Finish



STEP 12: Right click on ServiceClient Project and Add – New – JSP Page. (2 pages)



STEP 13 Right click on ServiceClient Add – New – Java Class. Name the class and give the package name and Finish



STEP 14: Edit index.html to create front end for the service

```
<!DOCTYPE html>
<!--
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.
-->
<html>
   <head>
  <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <form>
      <h2>One-way Operation</h2><br>
      <table>
         <tr><td><input type="text" name="ID" placeholder="Enter ID"></td></tr>
         <tr><td><input type="submit" formaction="Del_data.jsp" value="Delete
Data"></td></tr>
      </table>
      <h1>-------------------------------------------</h1>
        <h2>Request-Response Operation</h2><br><br>
         <input type="submit" formaction="getData.jsp" value="Get Data">       </form>
   </body>
</html>
```

STEP 15 Edit deletedata.java to carry out One-Way operation of deleting data from the tables

```
/*
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.ss;

/**
 *
@author Sophia
 */
public class deletedata {
    public static void deldata(String id){
     String a = id;
     Data ob = new Data();      ob.remove(a);
     System.out.println("Data has been deleted..");
 }
}
```

STEP 16: Edit del_data.jsp to delete data

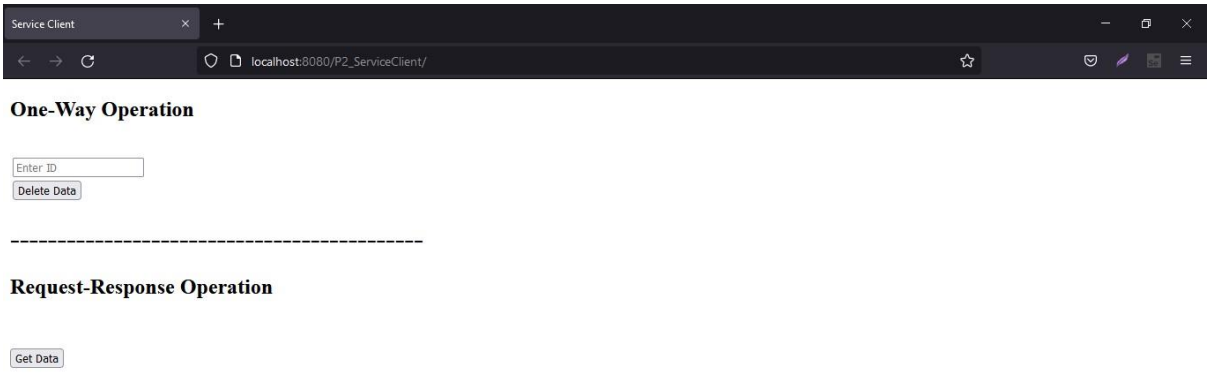
```
<%--
    Document   : del_data
    Created on : 30 Aug, 2021, 8:22:29 AM
    Author     : Sophia
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.ss.deletedata"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Delete Data</title>
    </head>
    <body>
     <%
     String id = request.getParameter("ID");       deletedata.deldata(id);
     %>
    </body>
</html>
```

STEP 17 Edit get_data.jsp to extract data from the table in our database

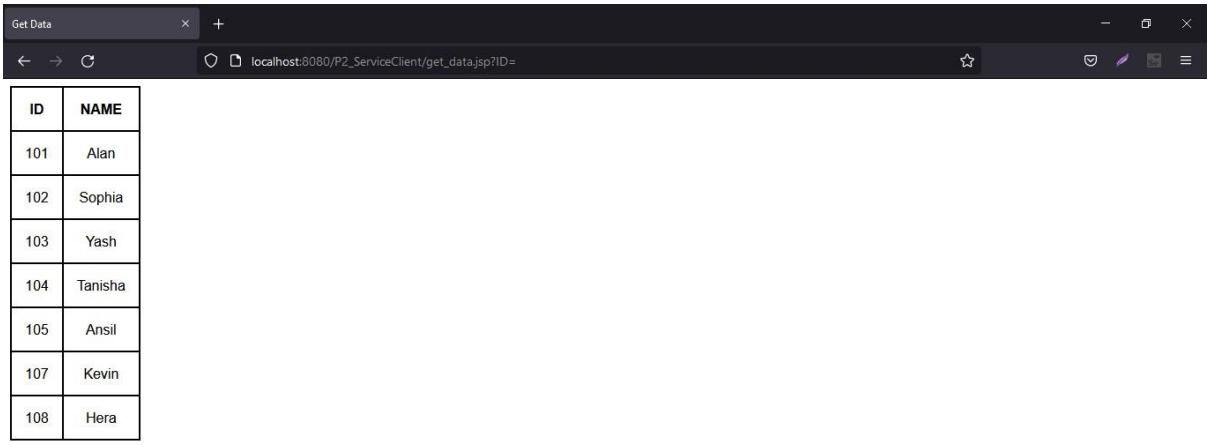
```
<%--
    Document   : get_data
    Created on : 30 Aug, 2021, 8:26:06 AM
    Author     : Sophia
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
 <title>Get Data</title>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initialscale=1.0">
 <style>  table {  font-family: arial, sans-serif;
 border-collapse: collapse;
 }
 td, th {
 border: 2px solid #000000;  text-align: center;
 padding: 15px;
 }
```

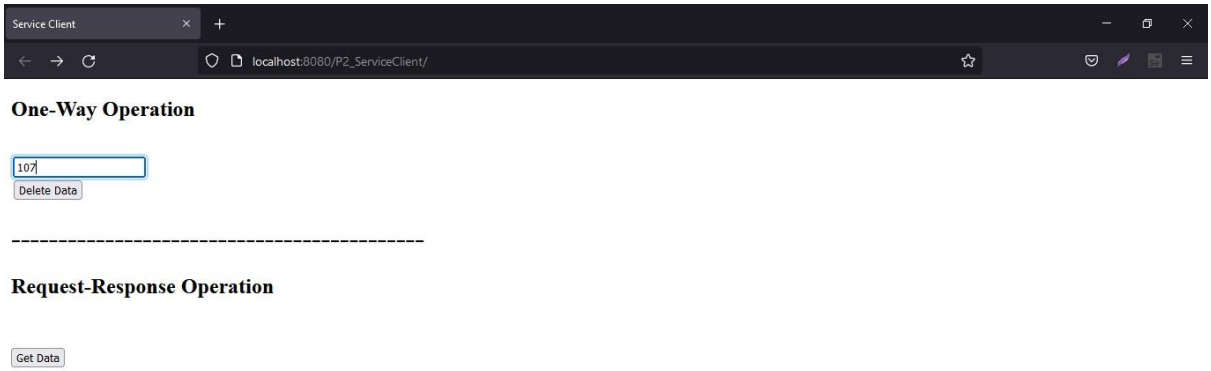
```
</style>  <script>
 var request = new XMLHttpRequest();
 request.open('GET', 'http://localhost:8080/P2_Server/webresources/com.kk.friends1/', true);
request.onload = function () { // begin accessing JSON data here  var data =
JSON.parse(this.response);
 for (var i = 0; i < data.length; i++) {
var 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>
</html>
```

OUTPUT :

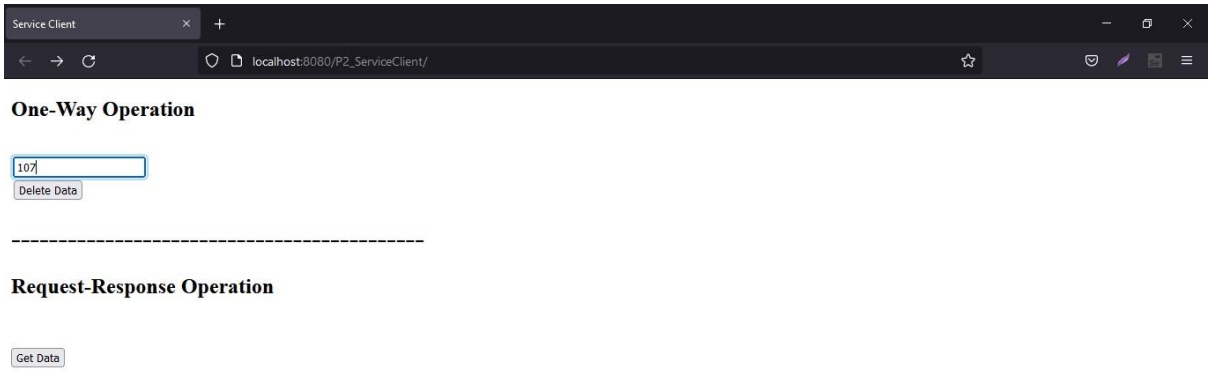

Service Client
localhost:8080/P2_ServiceClient/
One-Way Operation
Enter ID
Delete Data

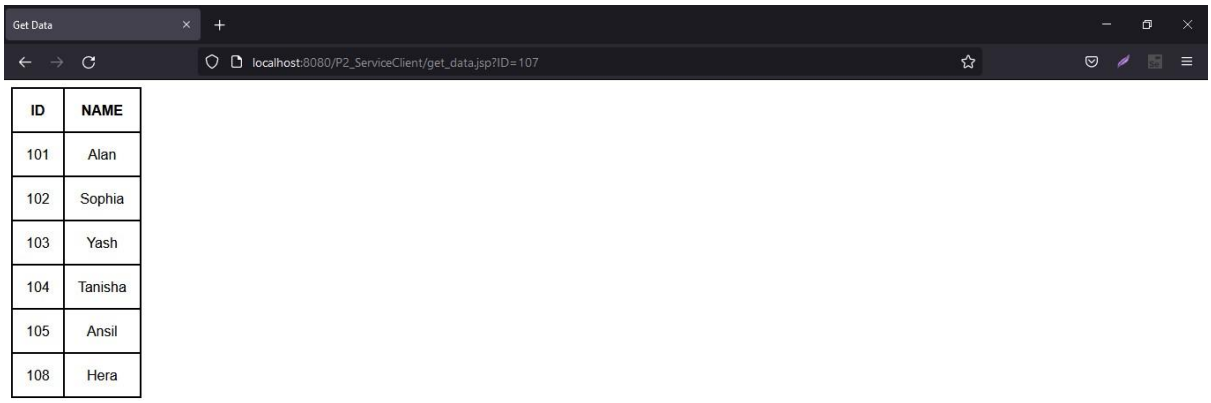
Request-Response Operation
Get Data


Get Data
localhost:8080/P2_ServiceClient/get_data.jsp?ID=
ID
NAME
101
Alan
102
Sophia
103
Yash
104
Tanisha
105
Ansil
107
Kevin
108
Hera


Service Client
localhost:8080/P2_ServiceClient/
One-Way Operation
107
Delete Data

Request-Response Operation
Get Data



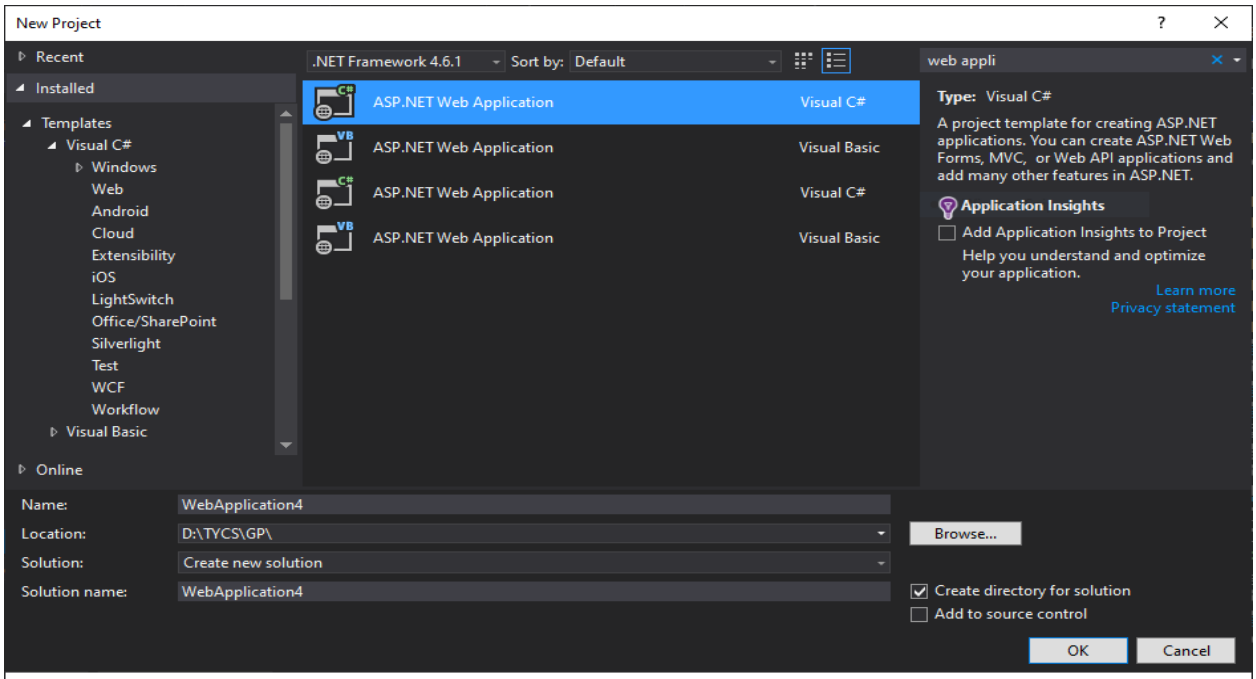


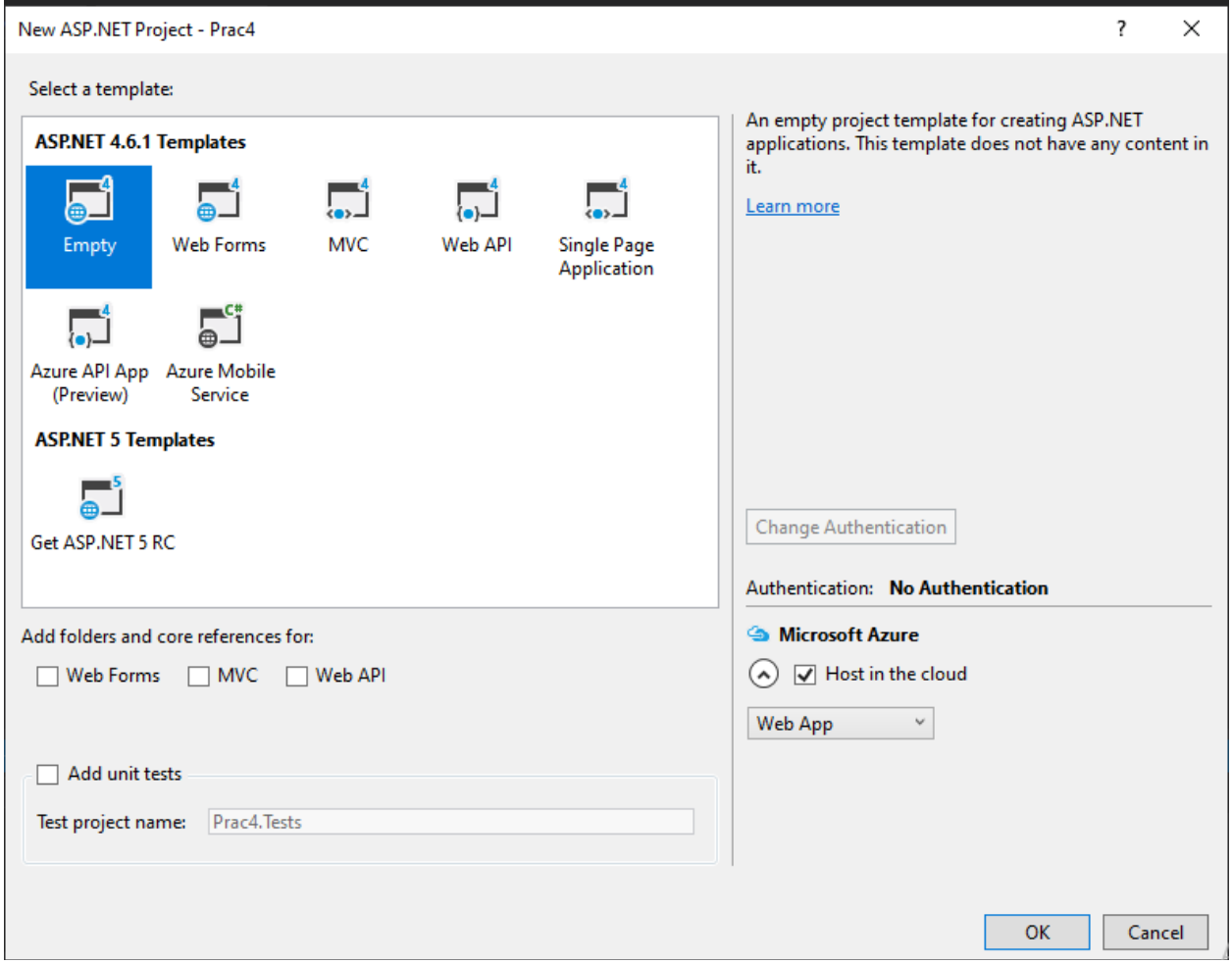
ID	NAME
101	Alan
102	Sophia
103	Yash
104	Tanisha
105	Ansil
108	Hera

Practical 3

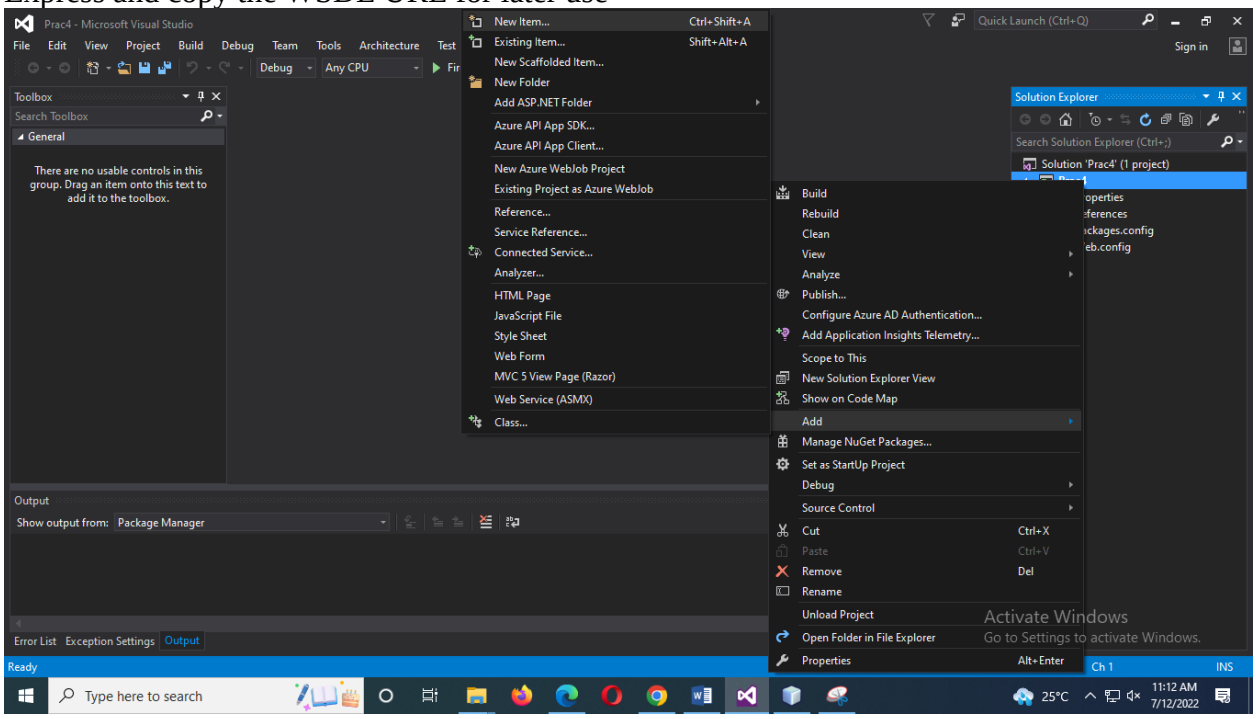
Develop client which consumes web services developed in different platform

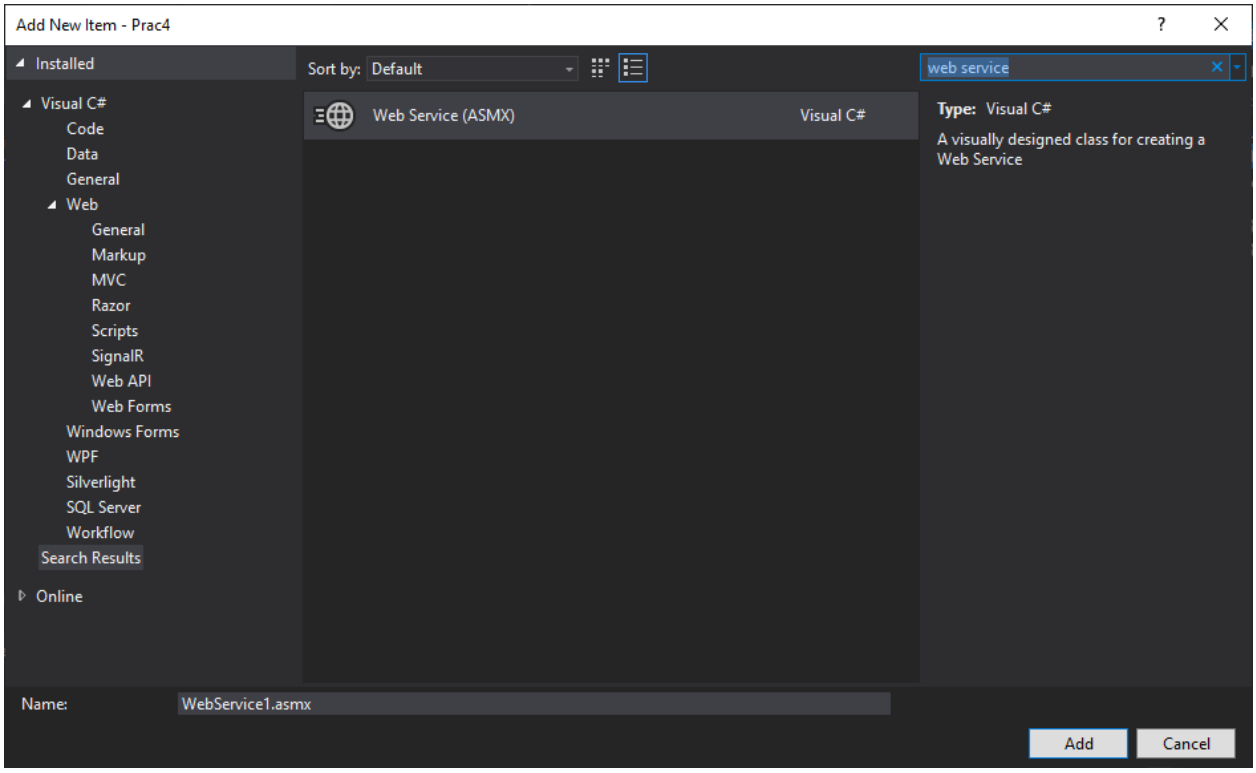
Step1: Open Visual Studio. Create a New ASP .net Web Application Project. Name it





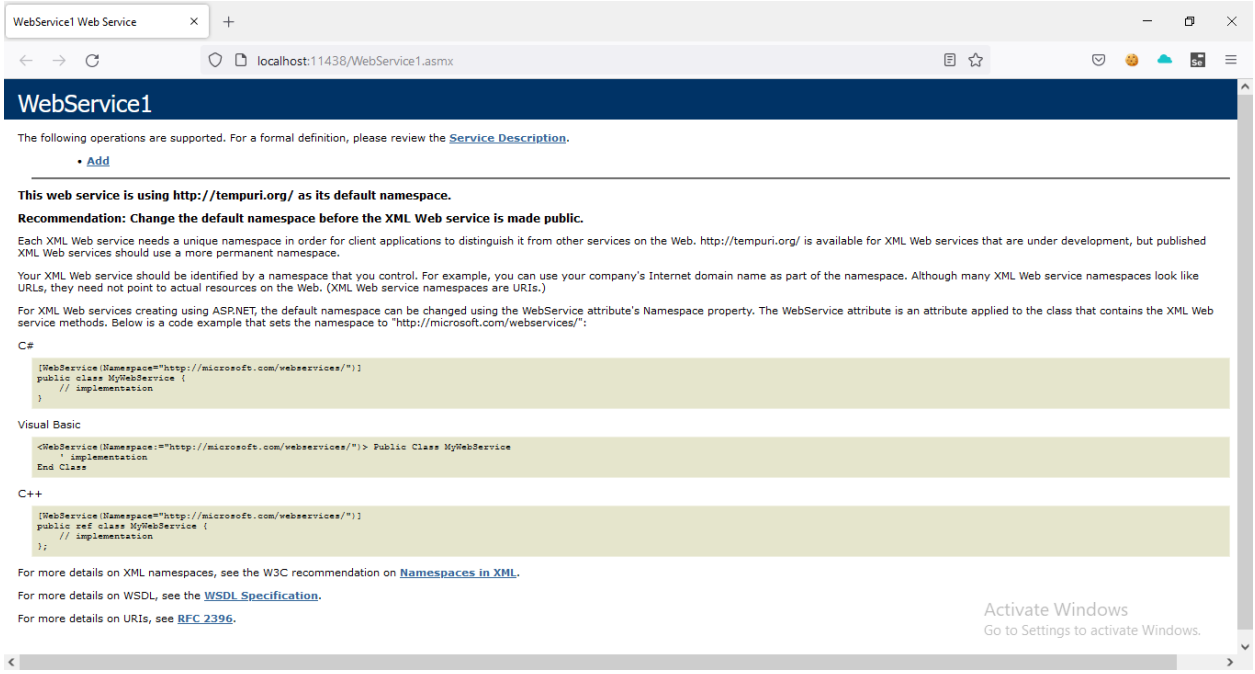
STEP 2: Add – New Item – Web Service(ASMX) and add the code to create server. Run IIS Express and copy the WSDL URL for later use



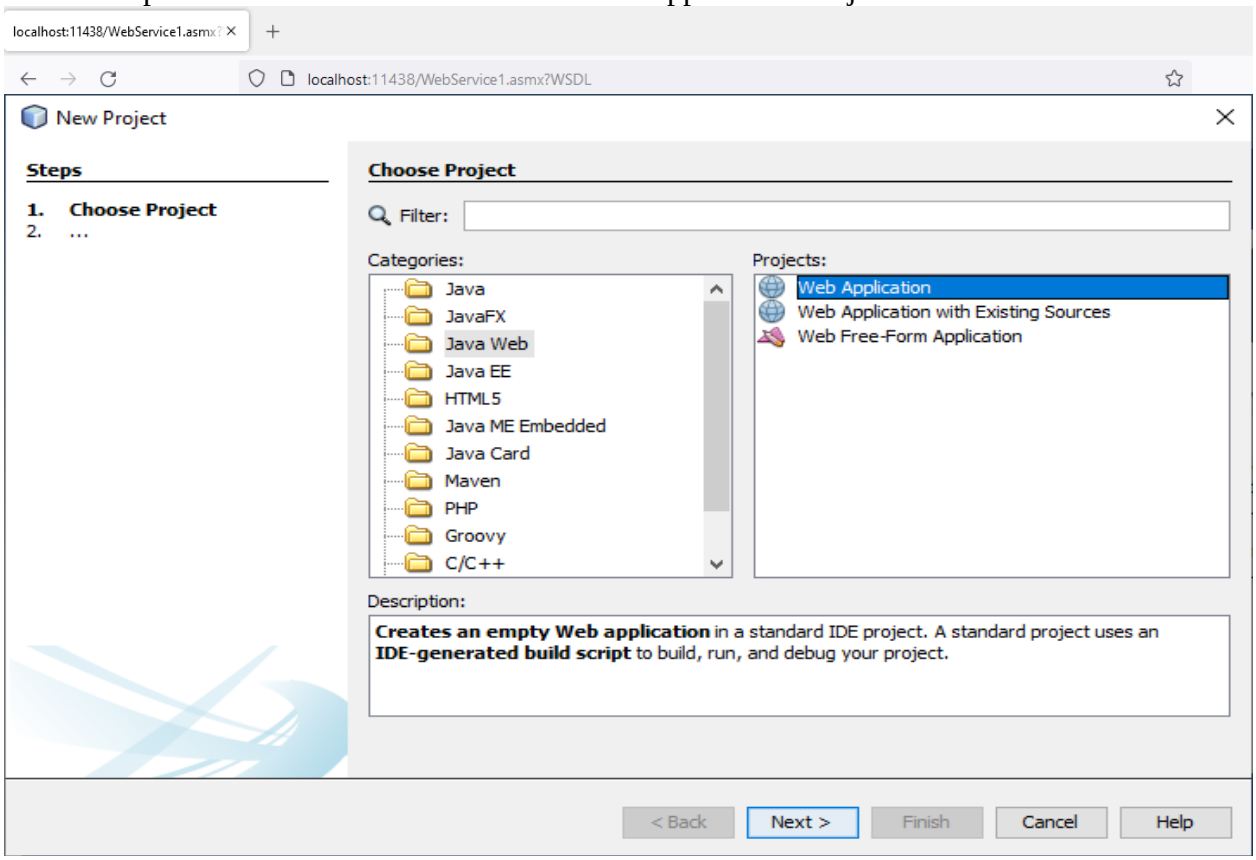


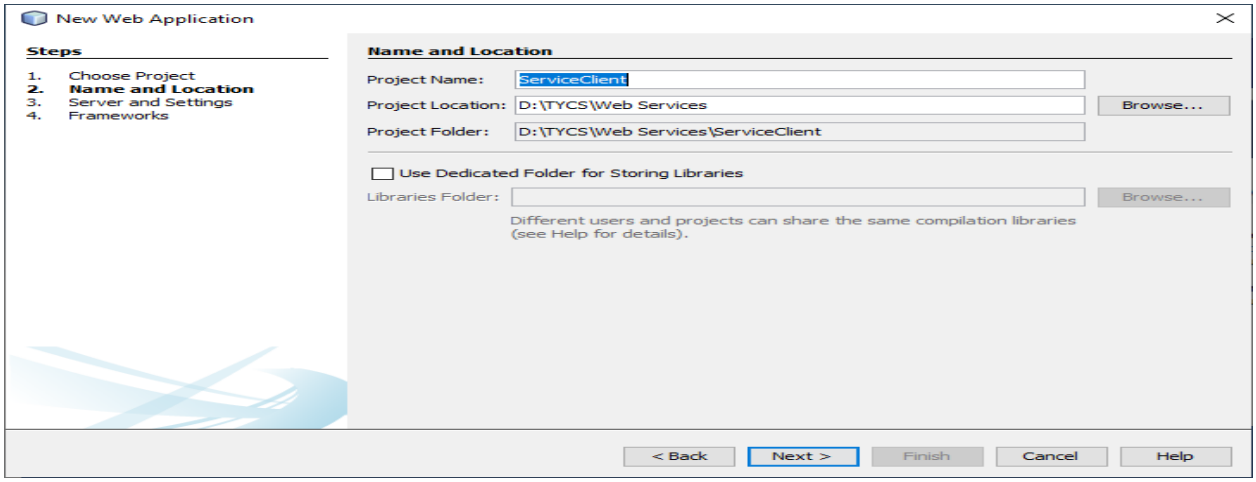

```
using System;
 using System.Collections.Generic;
using System.Web;
using System.Web.Services;
namespace Practical_2_WS_ {
 [WebService(Namespace = "http://tempuri.org/")] [WebServiceBinding(ConformsTo =
WsiProfiles.BasicProfile1_1)]
```

```
[System.ComponentModel.ToolboxItem(false)]
 public class WebService1 : System.Web.Services.WebService {
[WebMethod] public string HelloWorld() {
 return "Hello World"; }
 }
 }
```

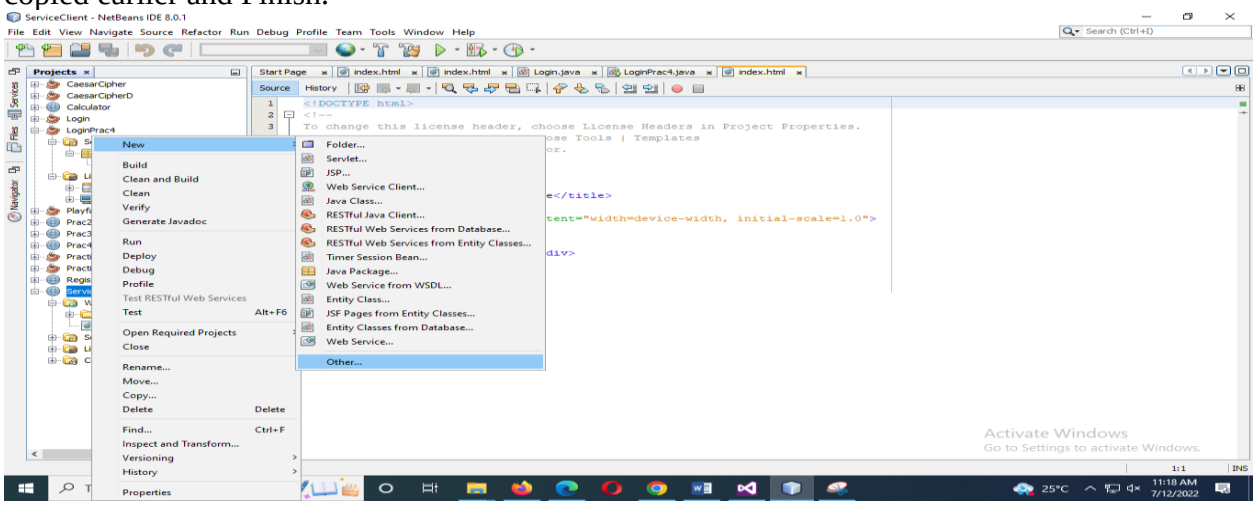


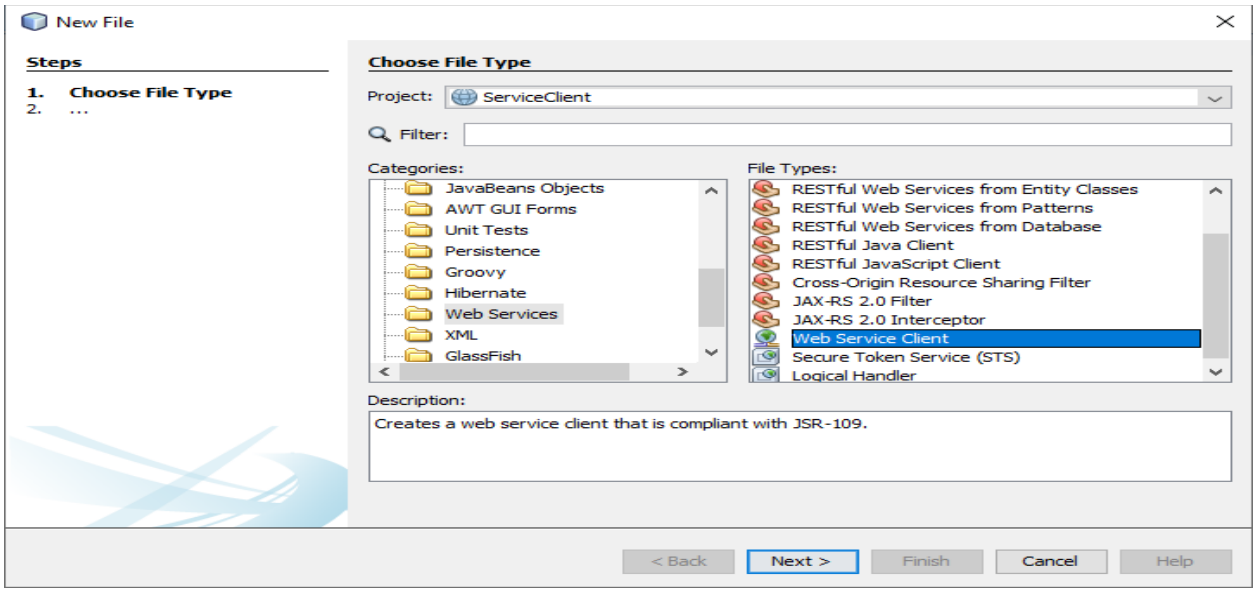
STEP 3: Open Netbeans IDE. Create a New Web Application Project. Name it

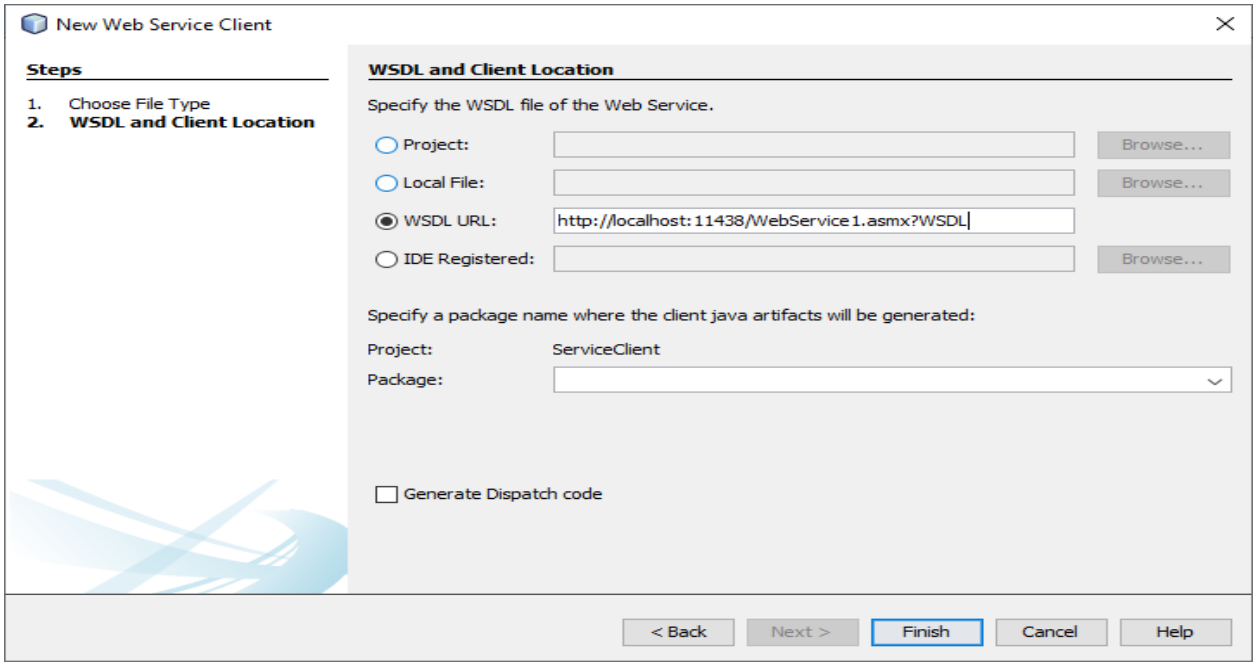


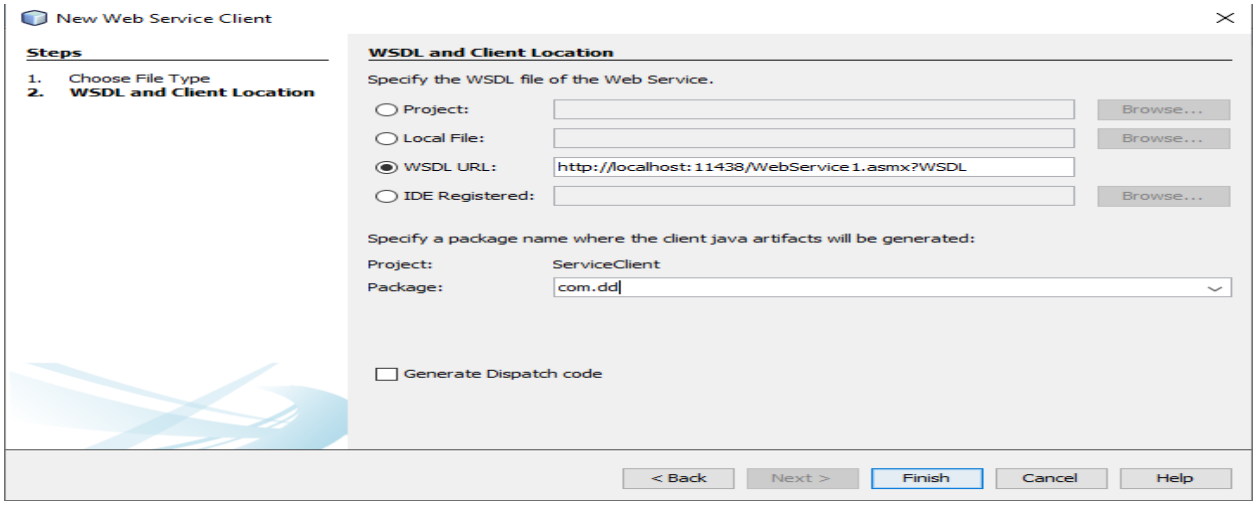


STEP 4: Add – New – Web Service Client. Keep everything default. Add the WSDL Url copied earlier and Finish.

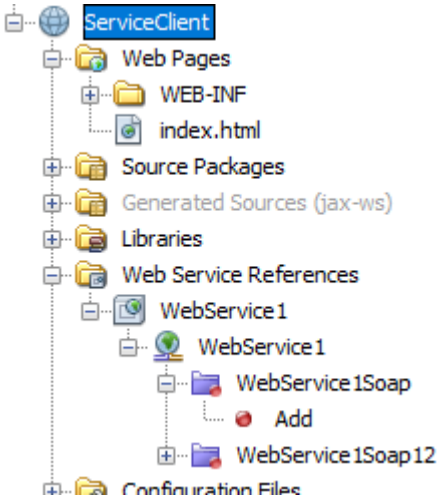


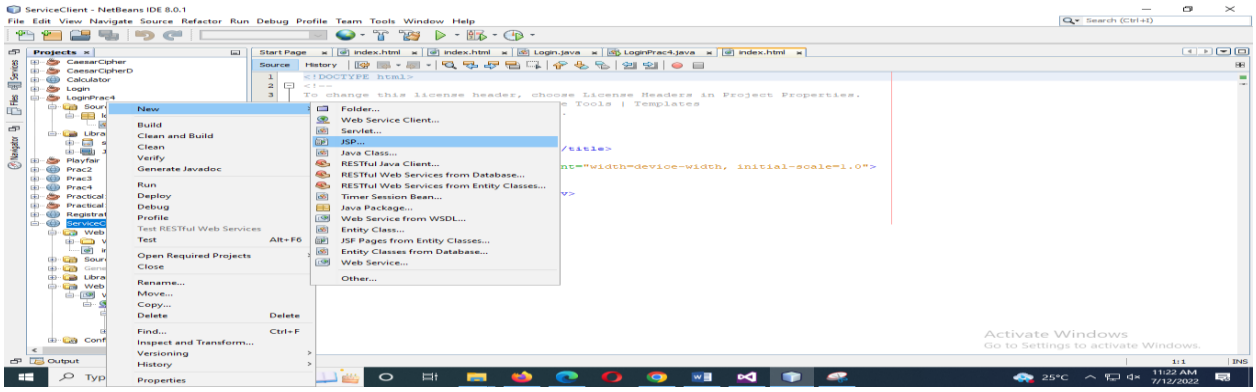


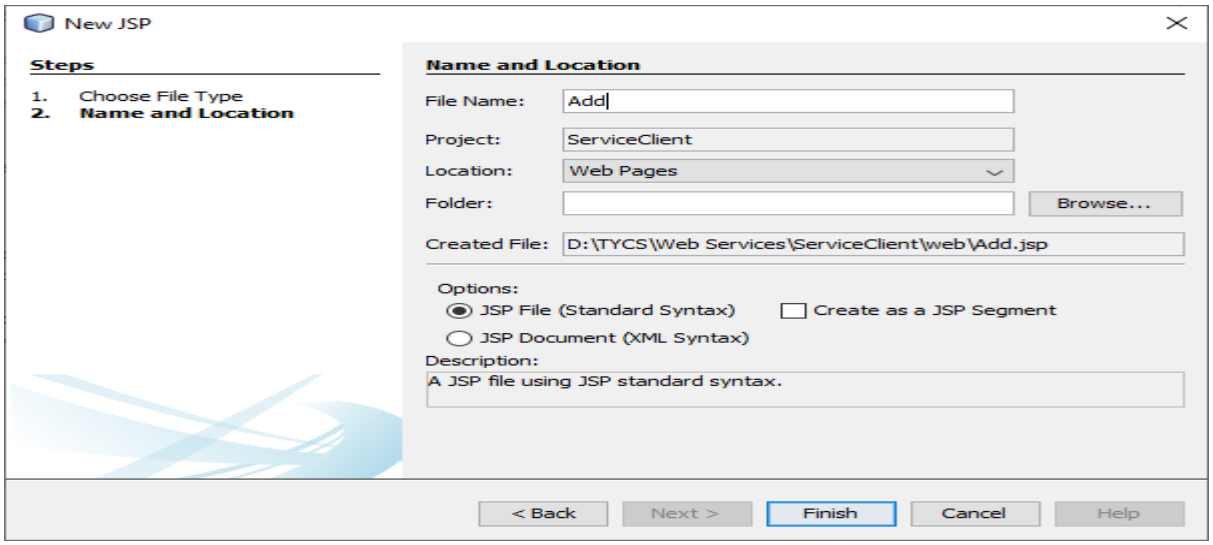




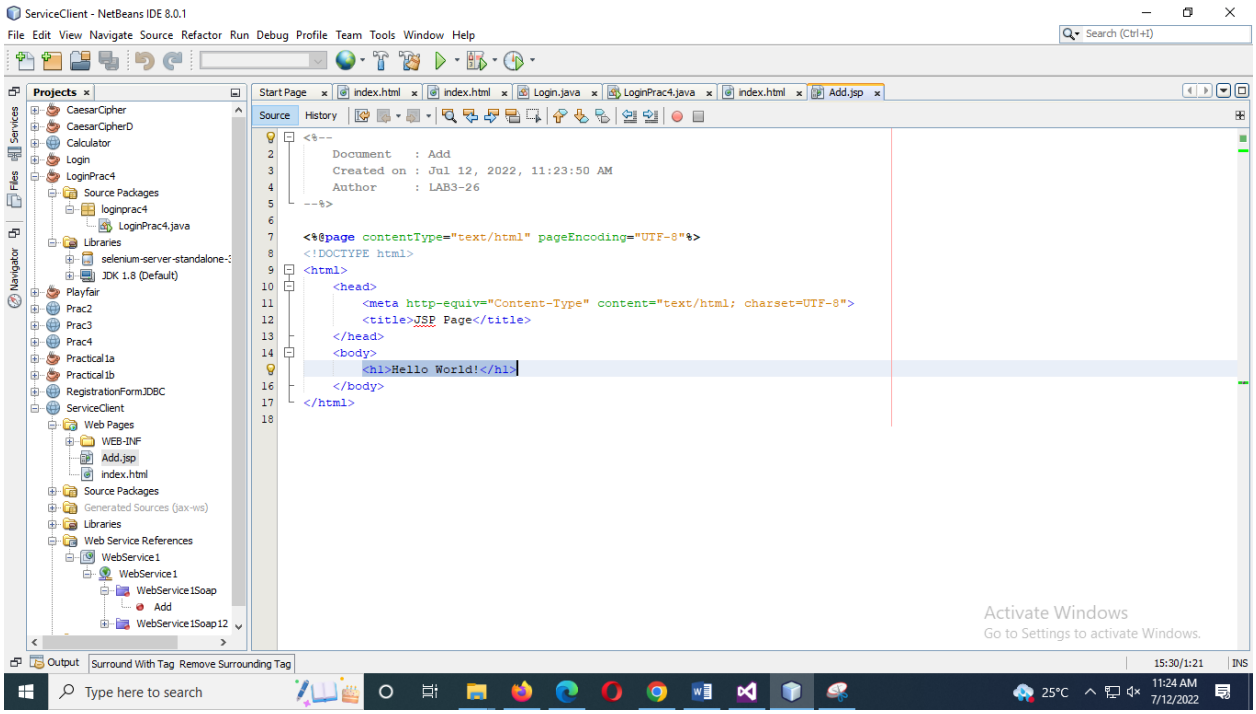
STEP 5: Add – New – JSP Page. Name it and keep everything default

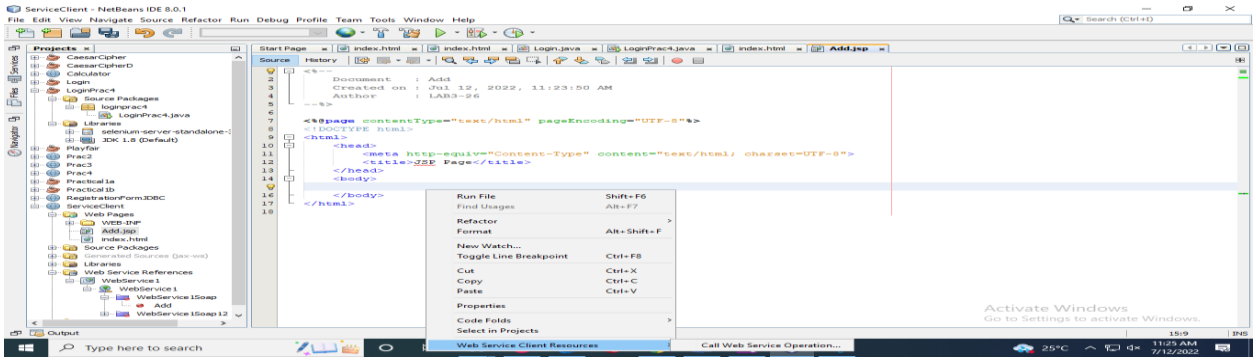


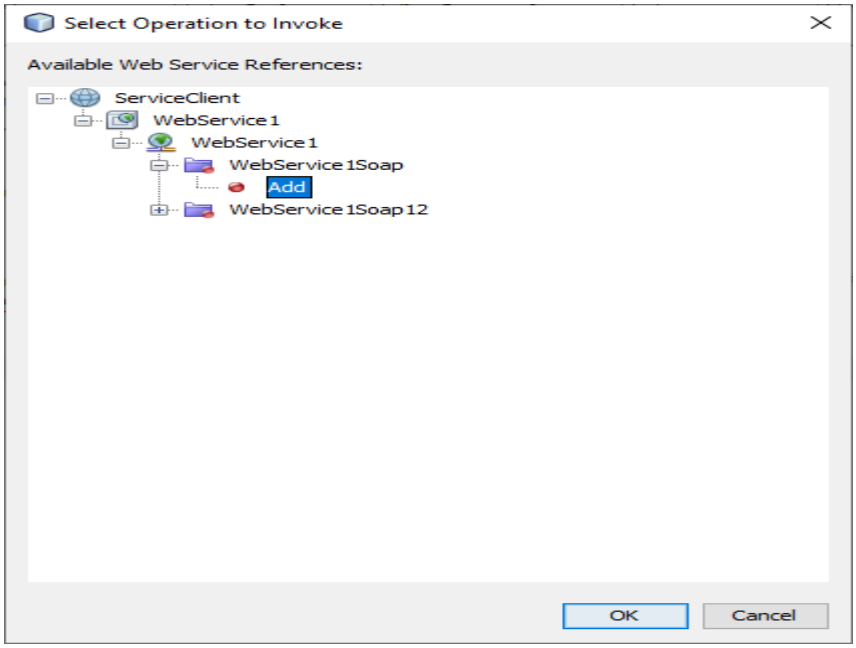


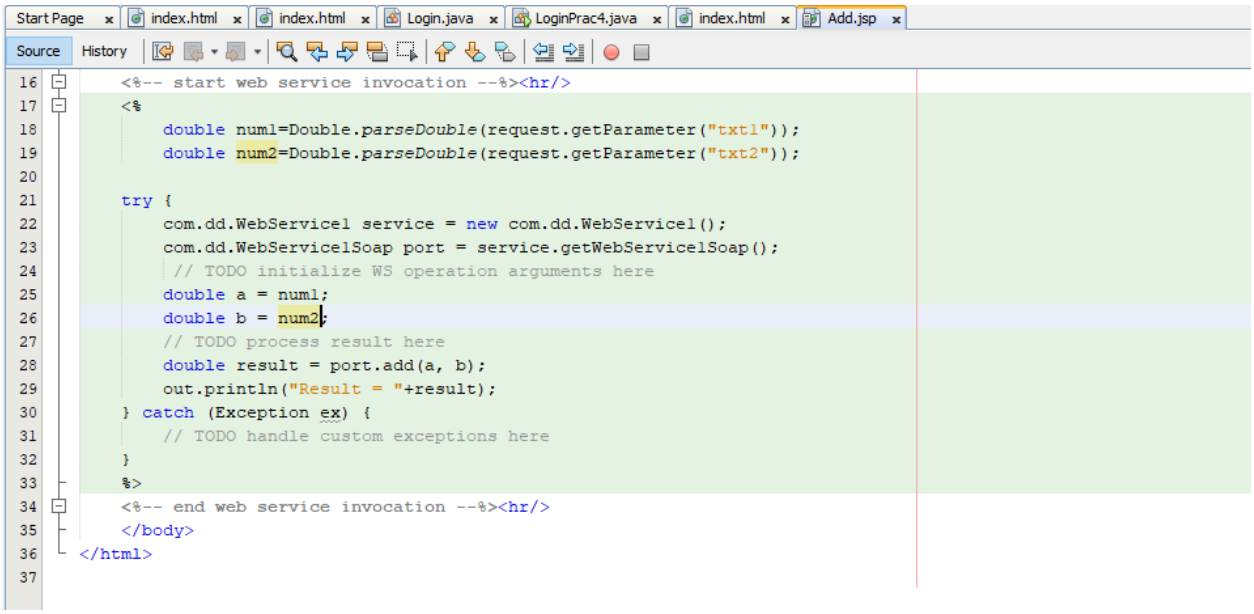


STEP 6: Right Click – Web Service Client Resources – Call Web Service Operation. Add the methods. Clean and Build and Run

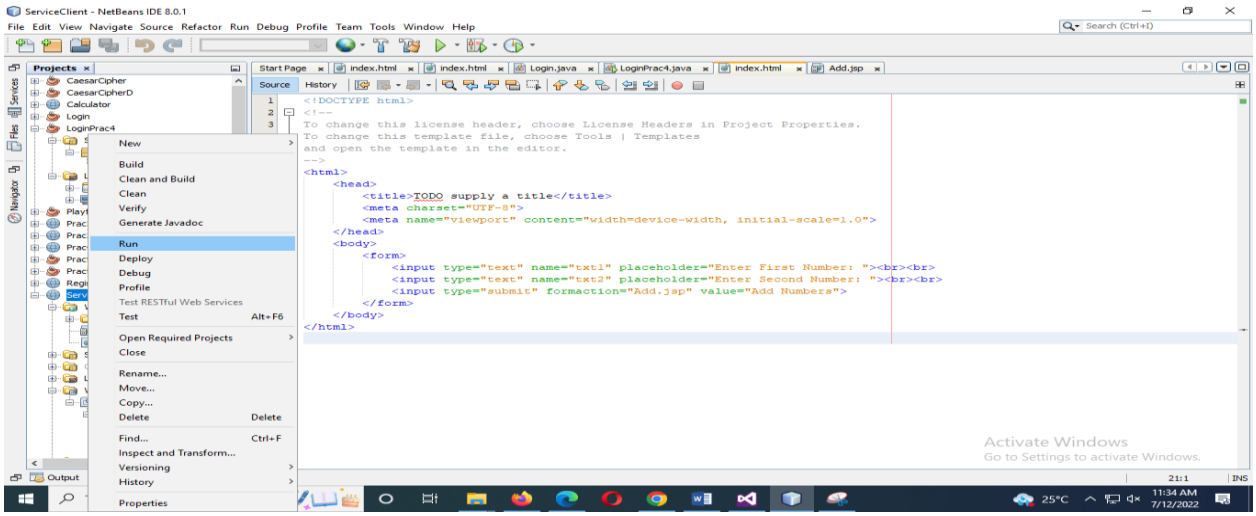



ServiceClient - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Start Page
index.html
Login.java
LoginPrac4.java
Add.jsp
Source
History
Run File
Find Usages
Refactor
Format
New Watch...
Toggle Line Breakpoint
Cut
Copy
Paste
Properties
Code Folds
Select in Projects
Web Service Client Resources
Call Web Service Operation...
Activate Windows
Go to Settings to activate Windows.
Output
Type here to search

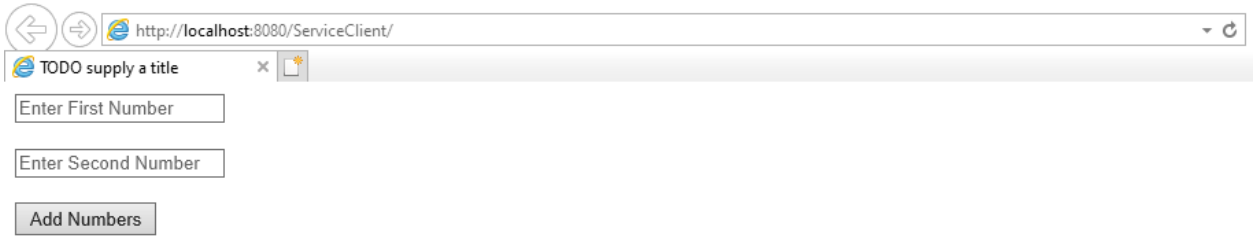

Select Operation to Invoke
Available Web Service References:
ServiceClient
WebService1
WebService1
WebService1Soap
Add
WebService1Soap12
OK
Cancel

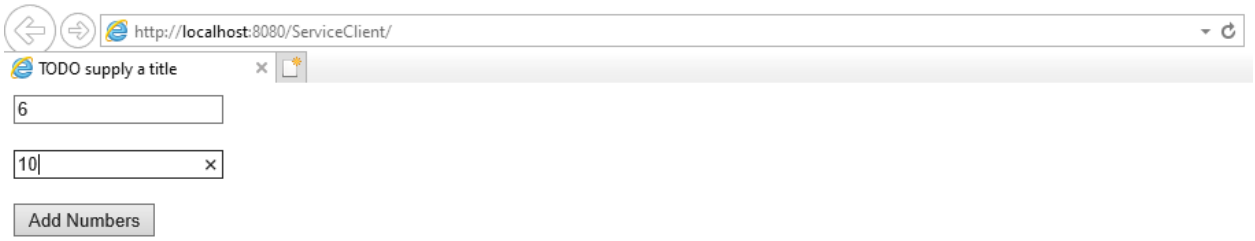

Start Page
index.html
index.html
Login.java
LoginPrac4.java
index.html
Add.jsp
Source
History
16 <%-- start web service invocation --%><hr/>
17 <%
18 double num1=Double.parseDouble(request.getParameter("txt1"));
19 double num2=Double.parseDouble(request.getParameter("txt2"));
20
21 try {
22 com.dd.WebService1 service = new com.dd.WebService1();
23 com.dd.WebService1Soap port = service.getWebService1Soap();
24 // TODO initialize WS operation arguments here
25 double a = num1;
26 double b = num2;
27 // TODO process result here
28 double result = port.add(a, b);
29 out.println("Result = "+result);
30 } catch (Exception ex) {
31 // TODO handle custom exceptions here
32 }
33 %>
34 <%-- end web service invocation --%><hr/>
35 </body>
36 </html>
37

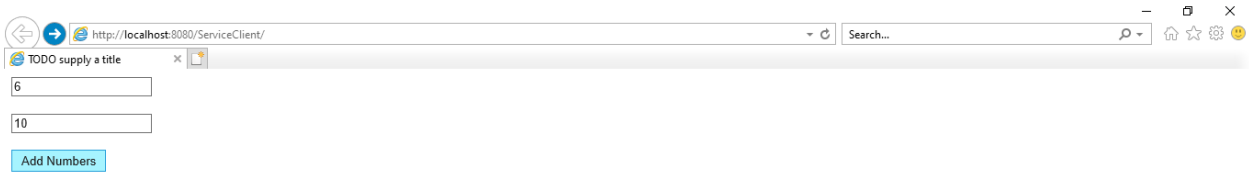
```html
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form>
            <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>
    </body>
</html>
```

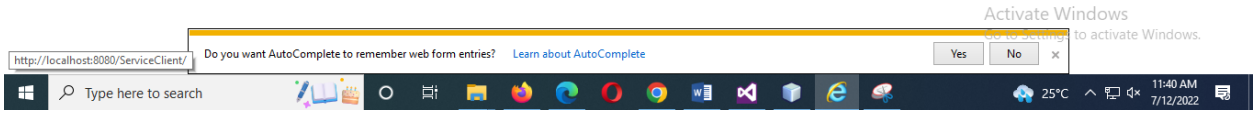


OUTPUT




http://localhost:8080/ServiceClient/
TODO supply a title
6
10
Add Numbers


http://localhost:8080/ServiceClient/
Search...
TODO supply a title
6
10
Add Numbers

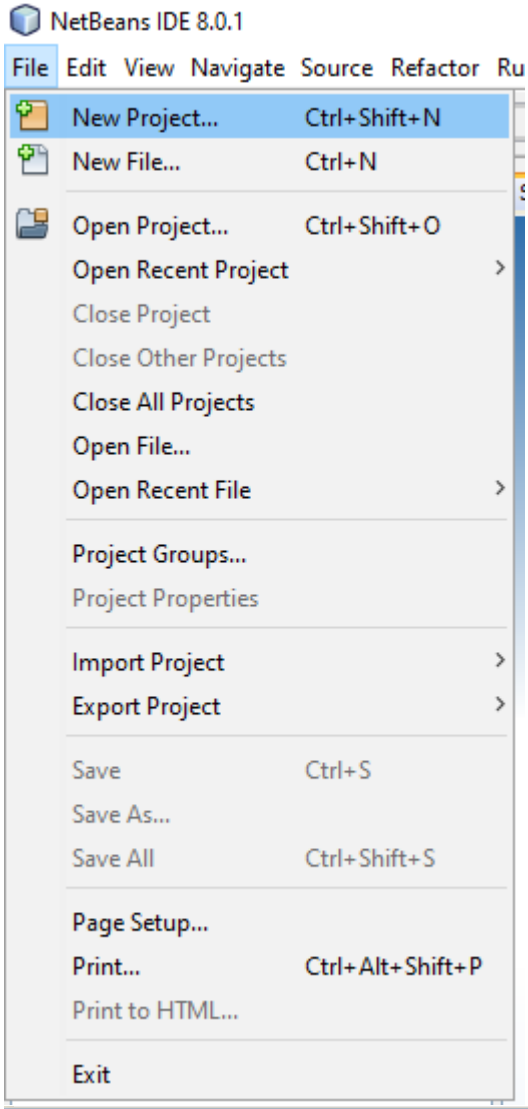

Activate Windows
http://localhost:8080/ServiceClient/
Do you want AutoComplete to remember web form entries?
Learn about AutoComplete
Yes
No
Type here to search
25°C
11:40 AM
7/12/2022

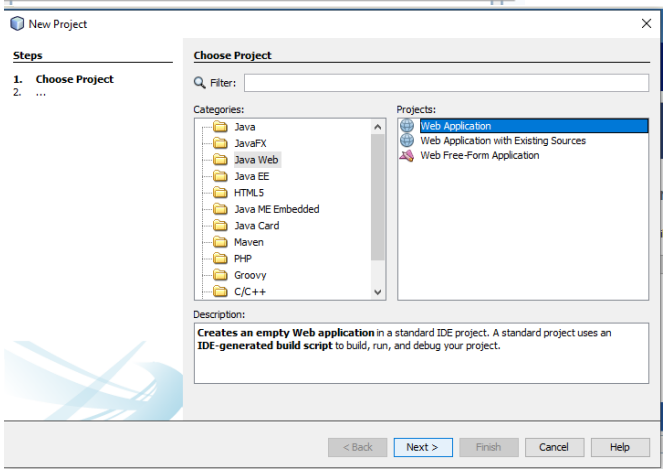
http://localhost:8080/ServiceClient/Add.jsp?txt1=6&txt2=10
JSP Page
Result = 16.0

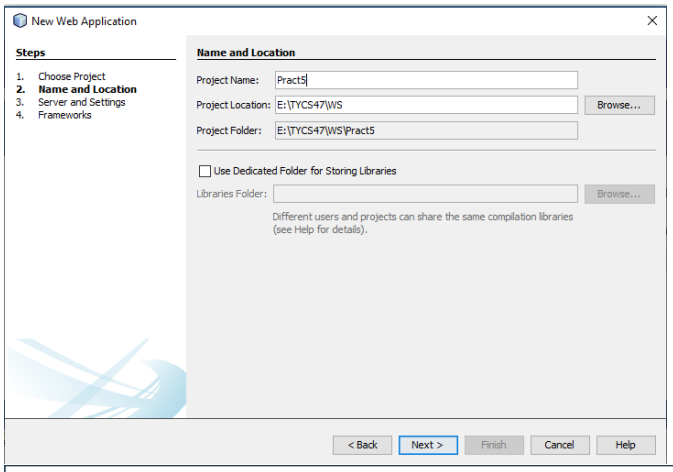
Practical 4

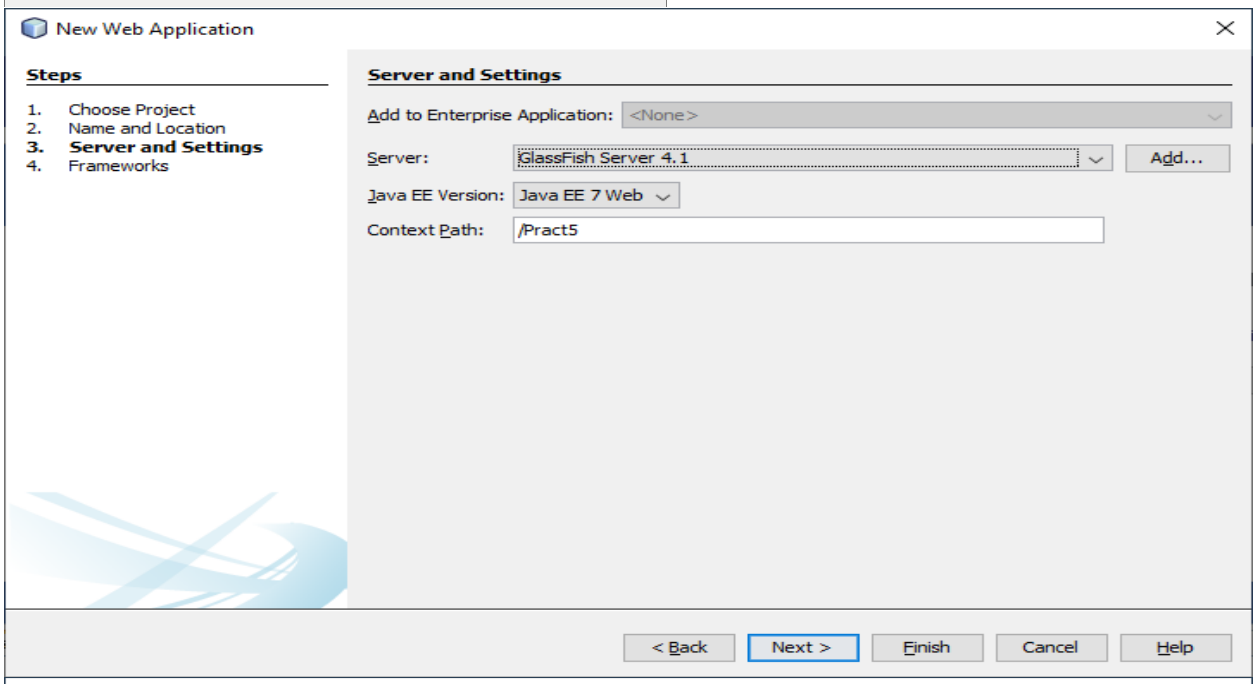
Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.

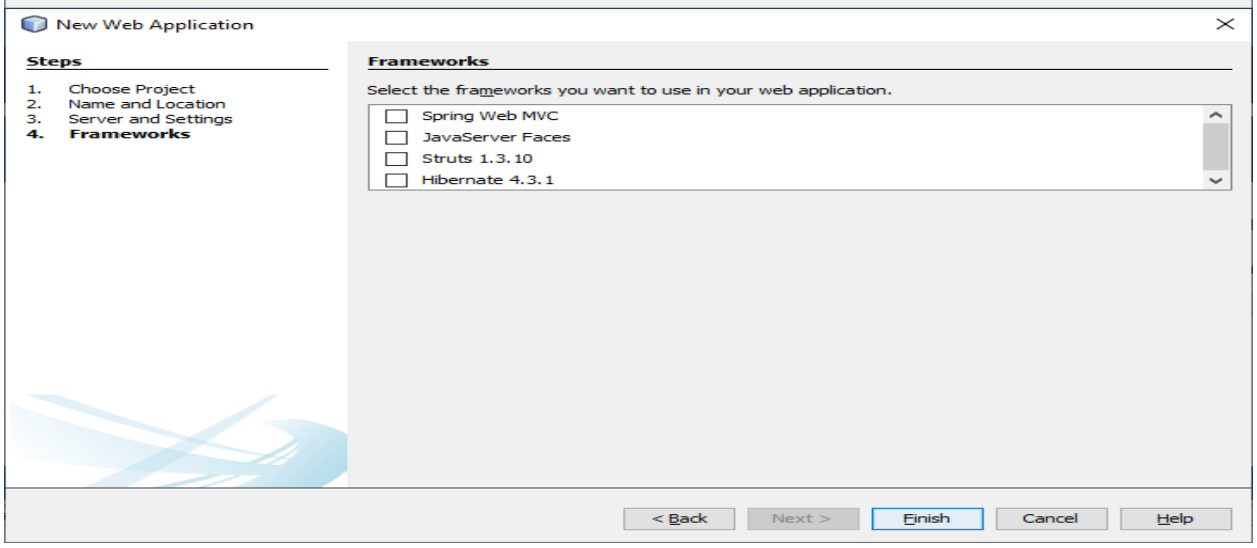
Step 1: Create a new Web Application project name it as Server-P6. Within the project create new java package and name it as tycs.

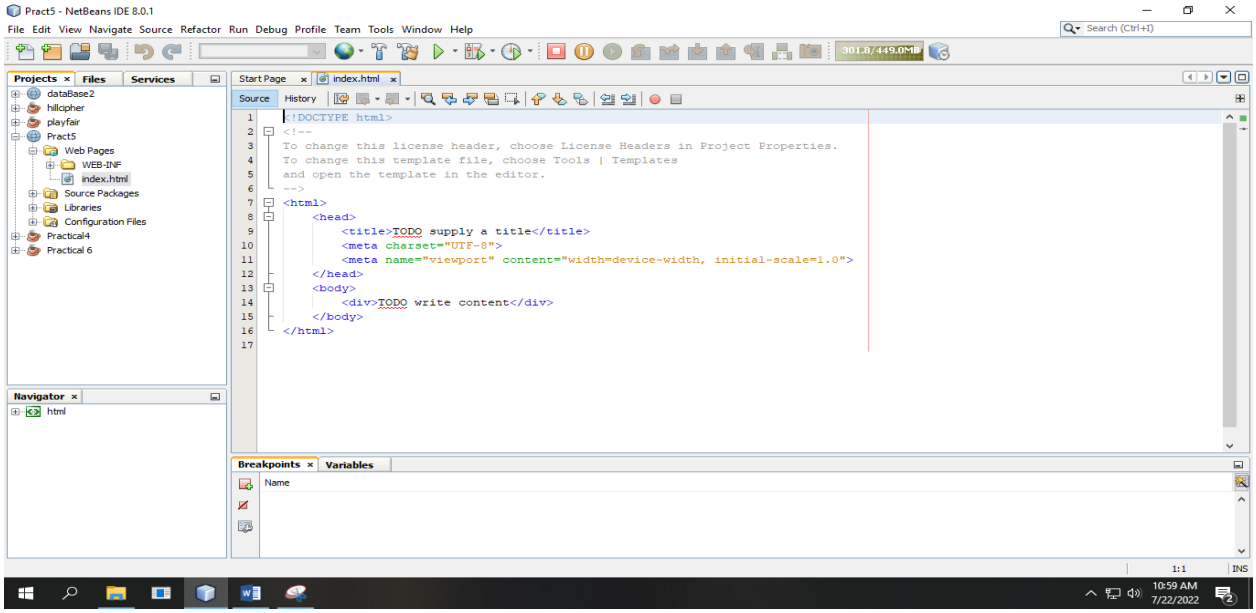




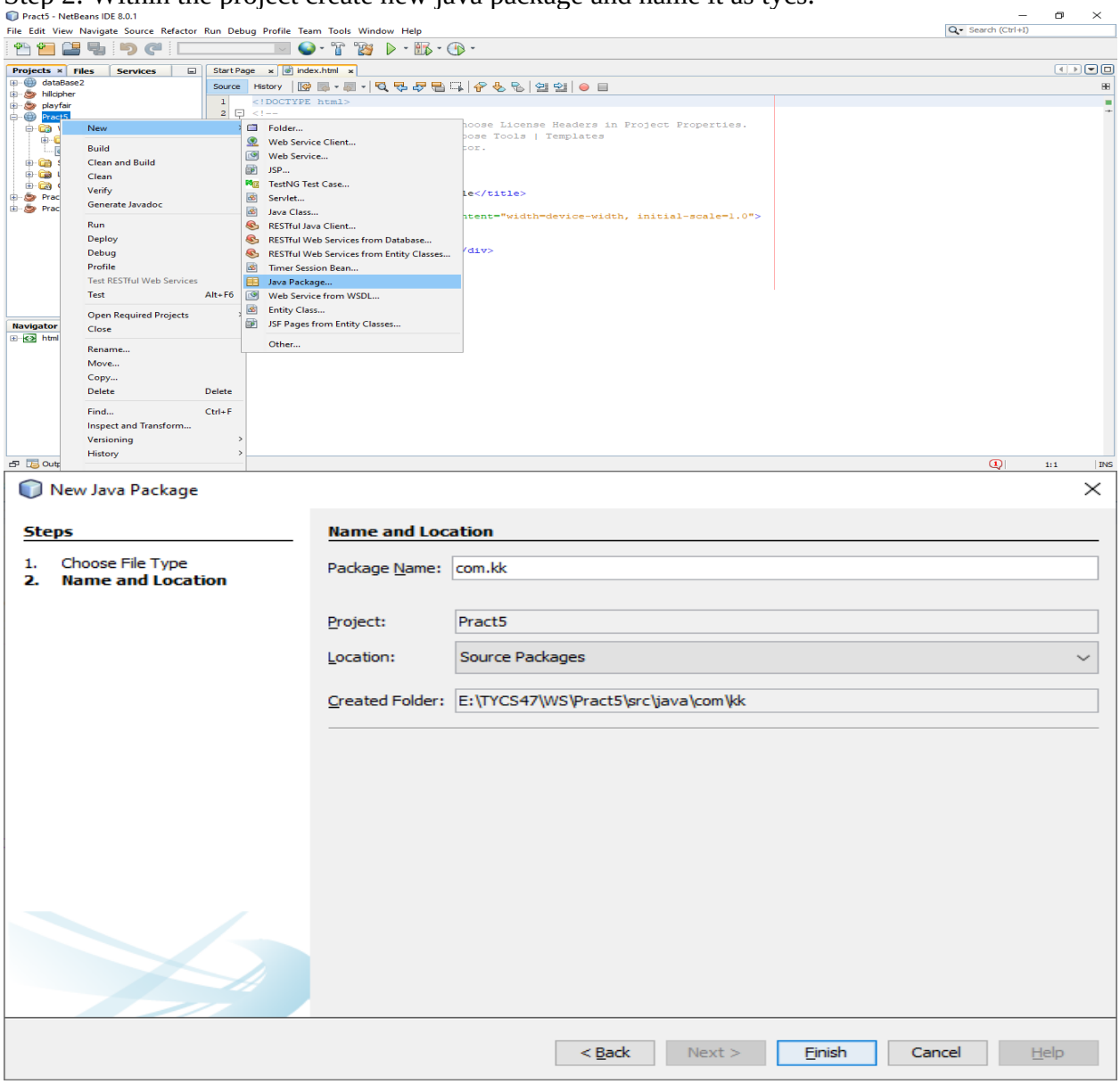

New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: Pract5
Project Location: E:\TYCS47\WS
Browse...
Project Folder: E:\TYCS47\WS\Pract5
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Browse...
Different users and projects can share the same compilation libraries (see Help for details).
< Back
Next >
Finish
Cancel
Help

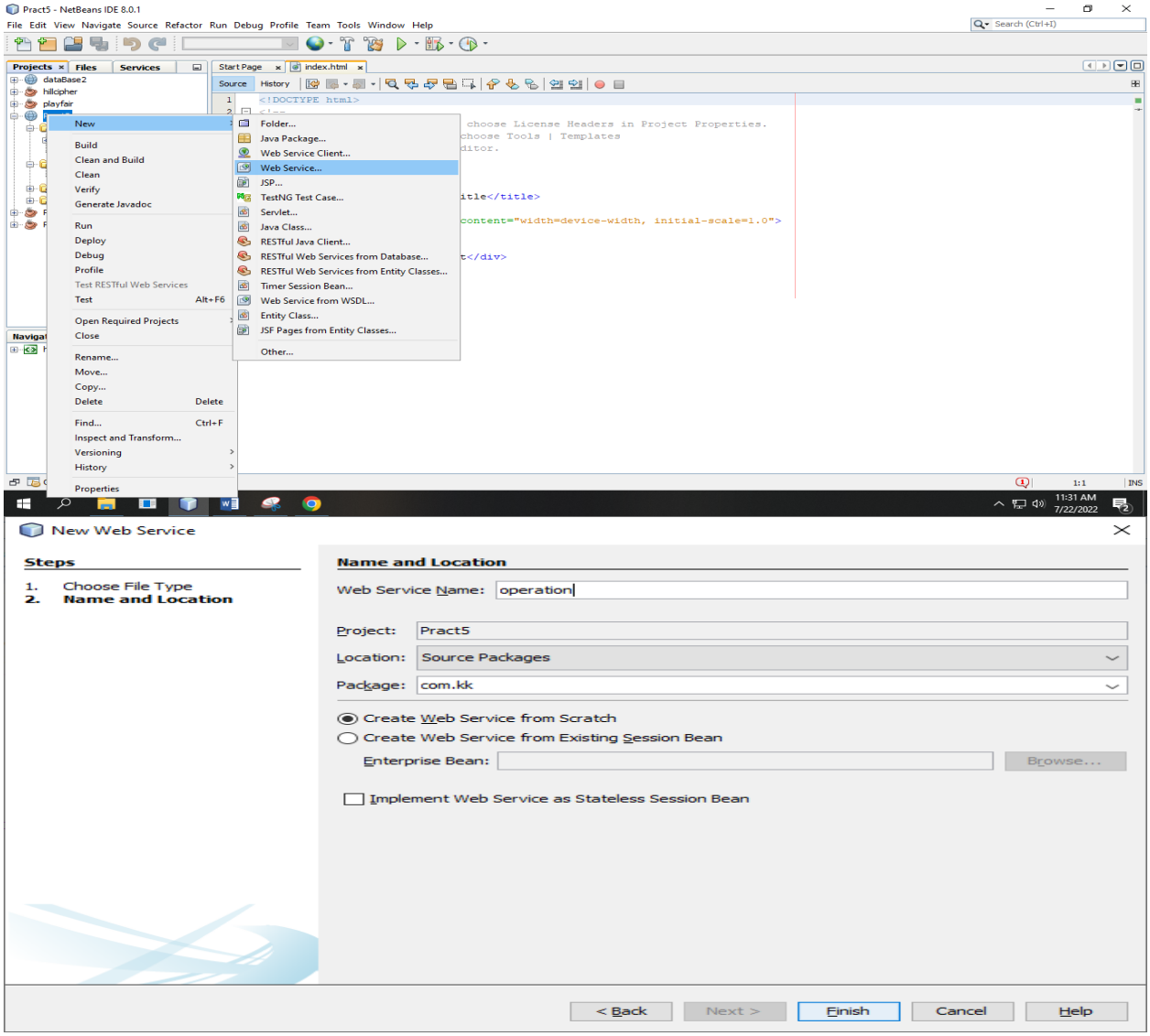

New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Server and Settings
Add to Enterprise Application: <None>
Server: GlassFish Server 4.1
Add...
Java EE Version: Java EE 7 Web
Context Path: /Pract5
< Back
Next >
Finish
Cancel
Help


New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Frameworks
Select the frameworks you want to use in your web application.
Spring Web MVC
JavaServer Faces
Struts 1.3.10
Hibernate 4.3.1
< Back
Next >
Finish
Cancel
Help

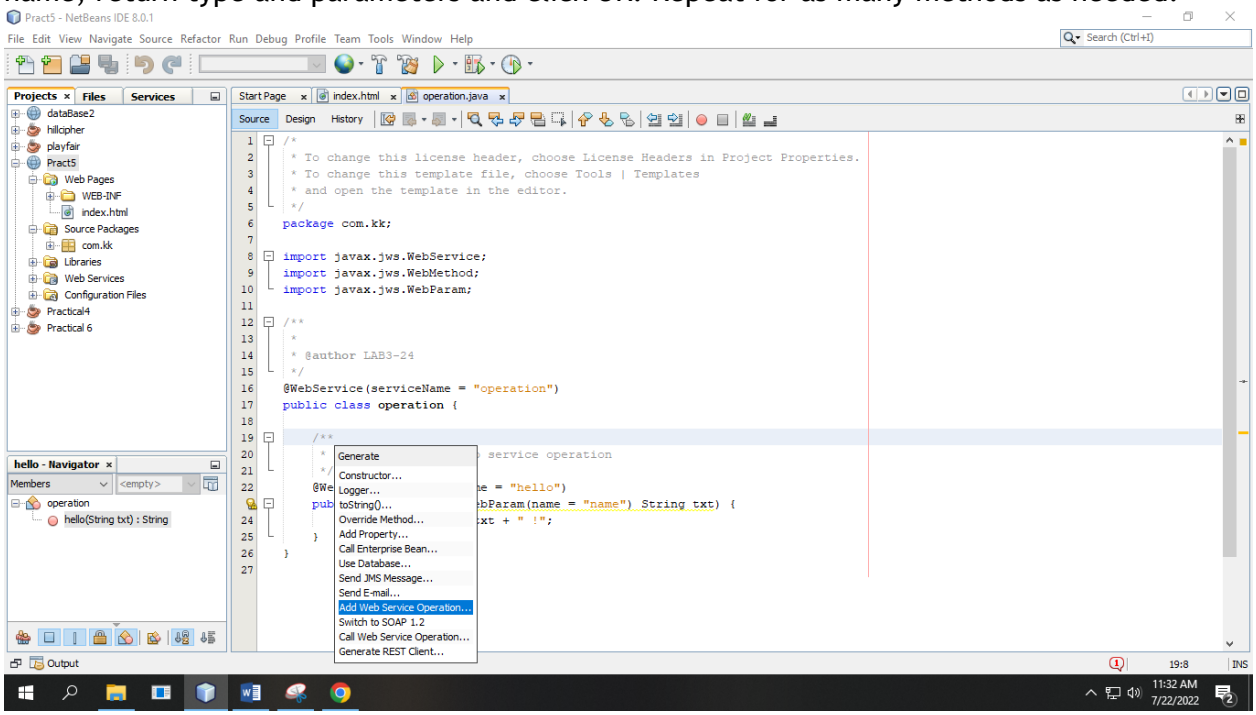


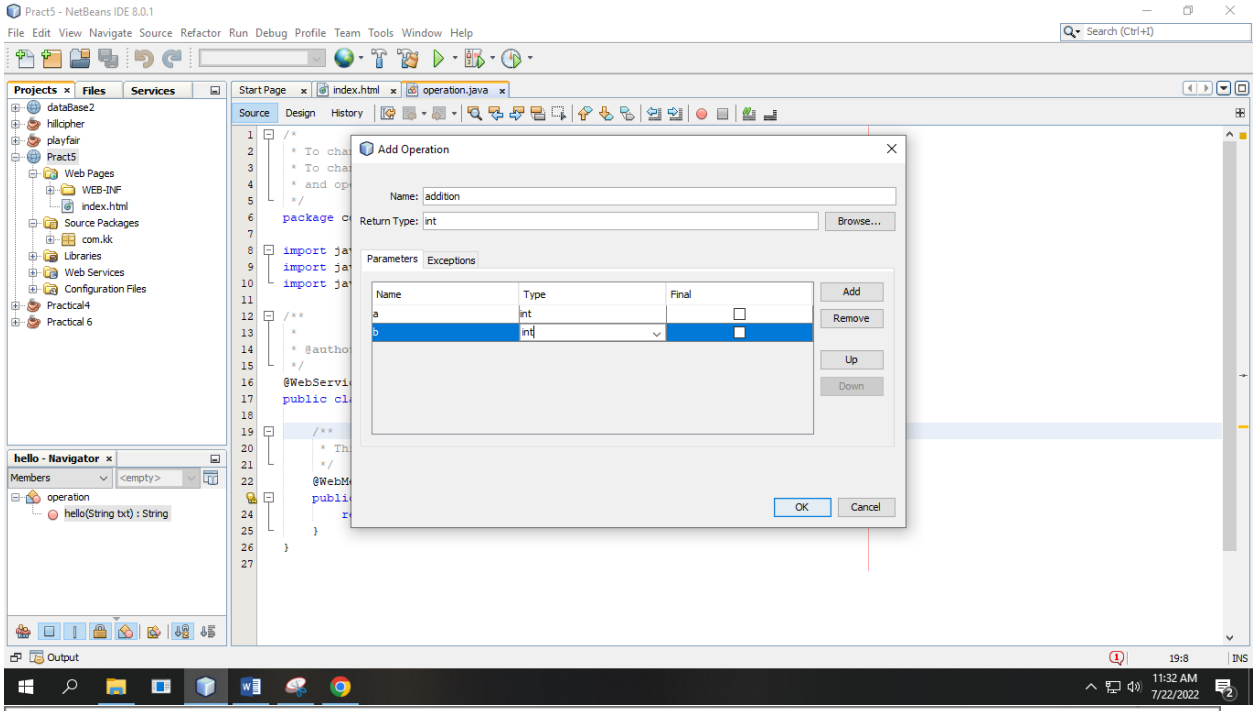
Step 2: Within the project create new java package and name it as tycs.

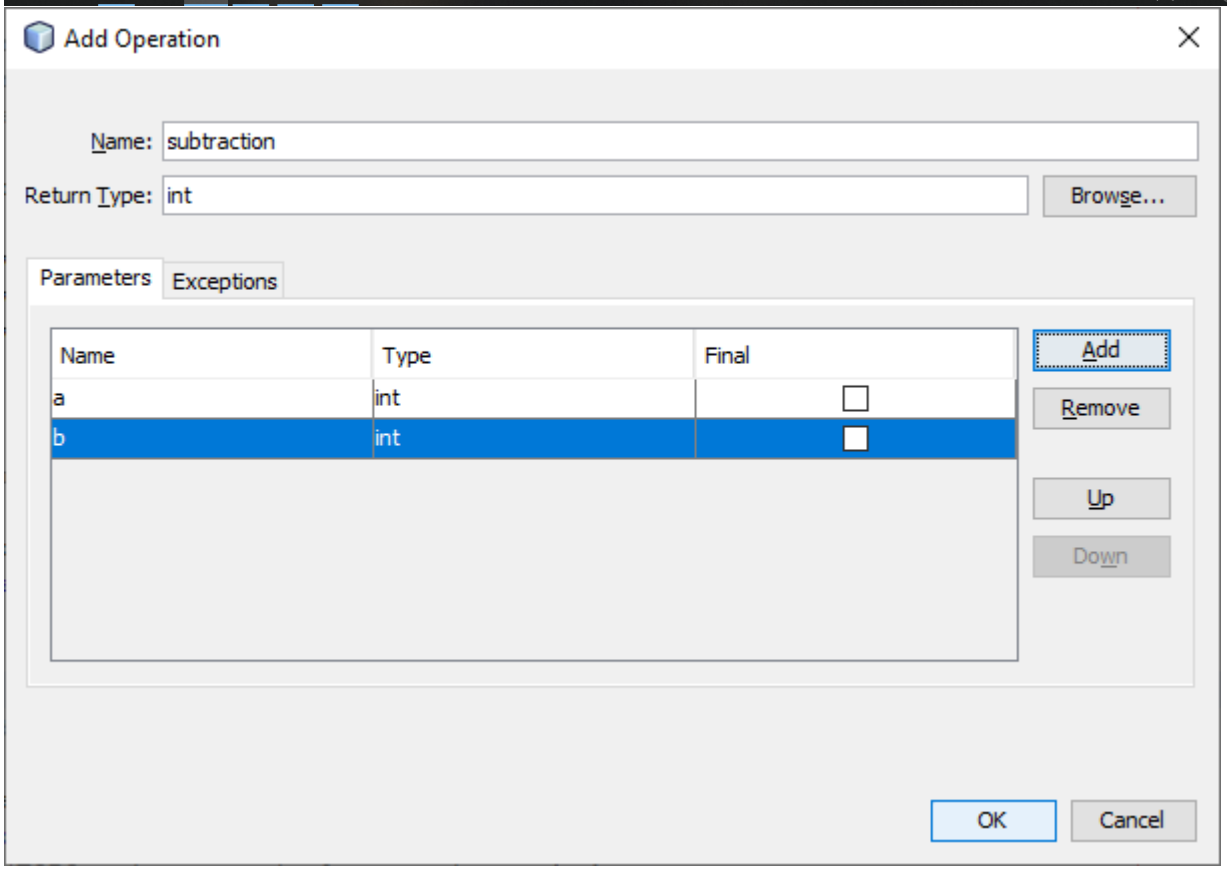


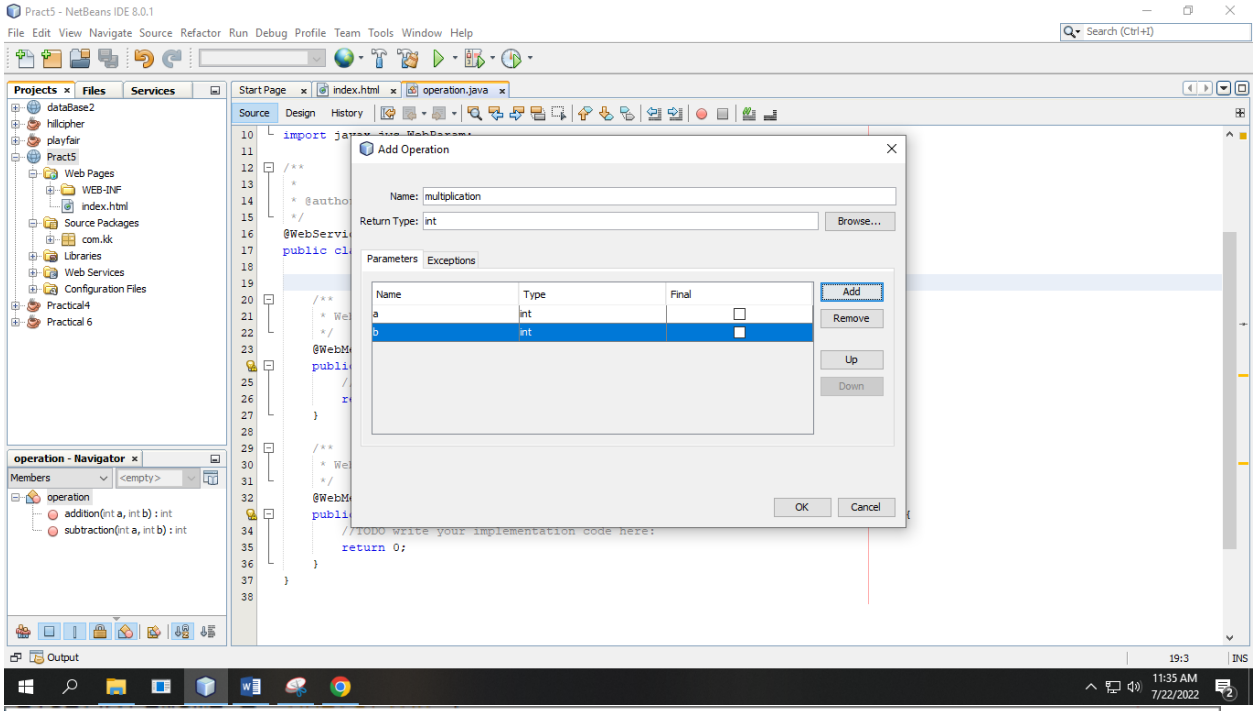


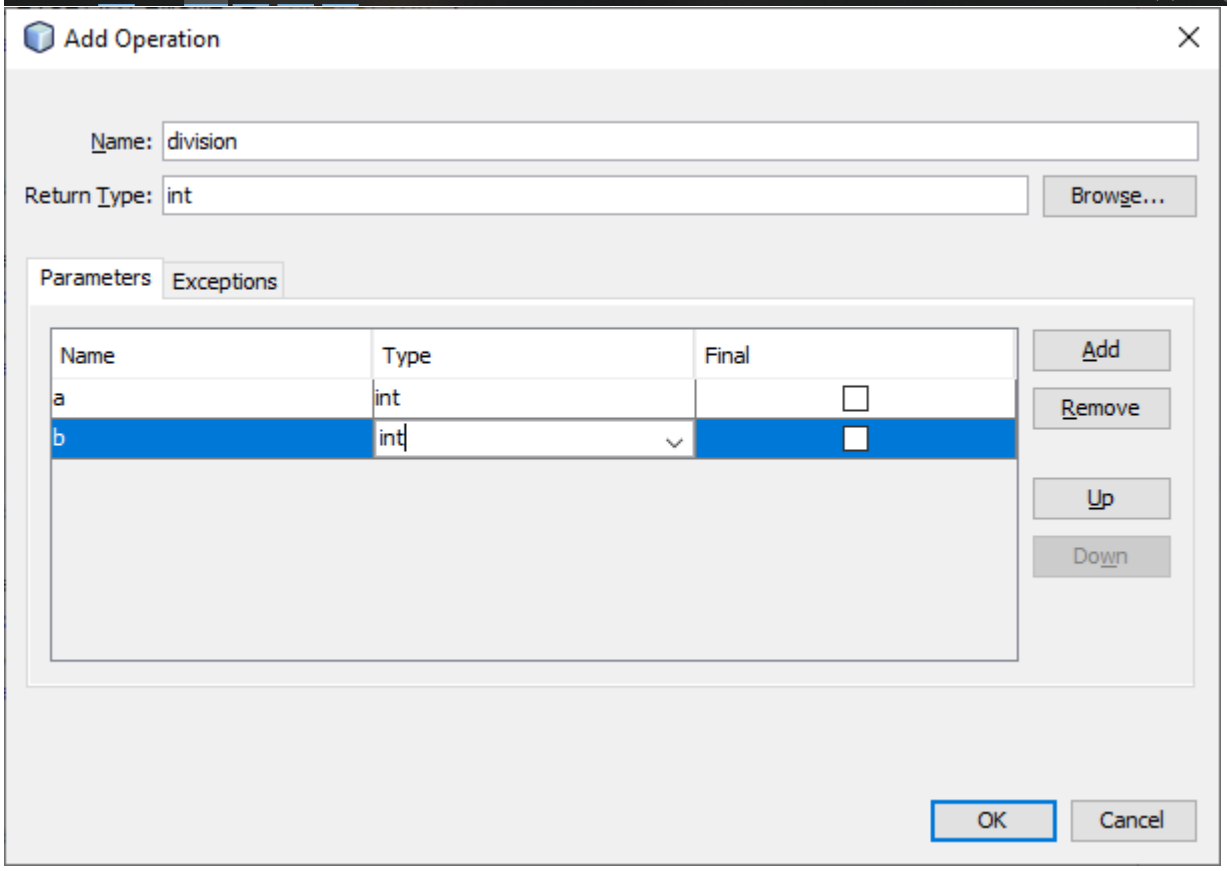
STEP 3: Right Click on code – Insert Code – Add Web Service Operation . Add the method name, return type and parameters and click OK. Repeat for as many methods as needed.

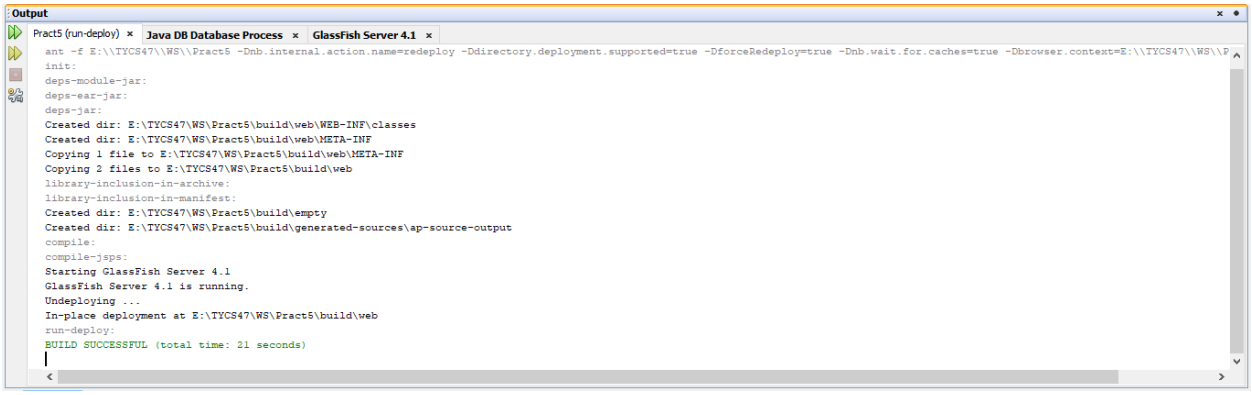



Pract5 - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
dataBase2
hillcipher
playfair
Pract5
Web Pages
WEB-INF
index.html
Source Packages
com.kk
Libraries
Web Services
Configuration Files
Practical4
Practical 6
hello - Navigator
operation
hello(String txt) : String
Start Page
index.html
operation.java
Source
Design
History
Add Operation
Name: addition
Return Type: int
Browse...
Parameters
Exceptions
Name
Type
Final
a
int
b
int
Add
Remove
Up
Down
OK
Cancel
Output

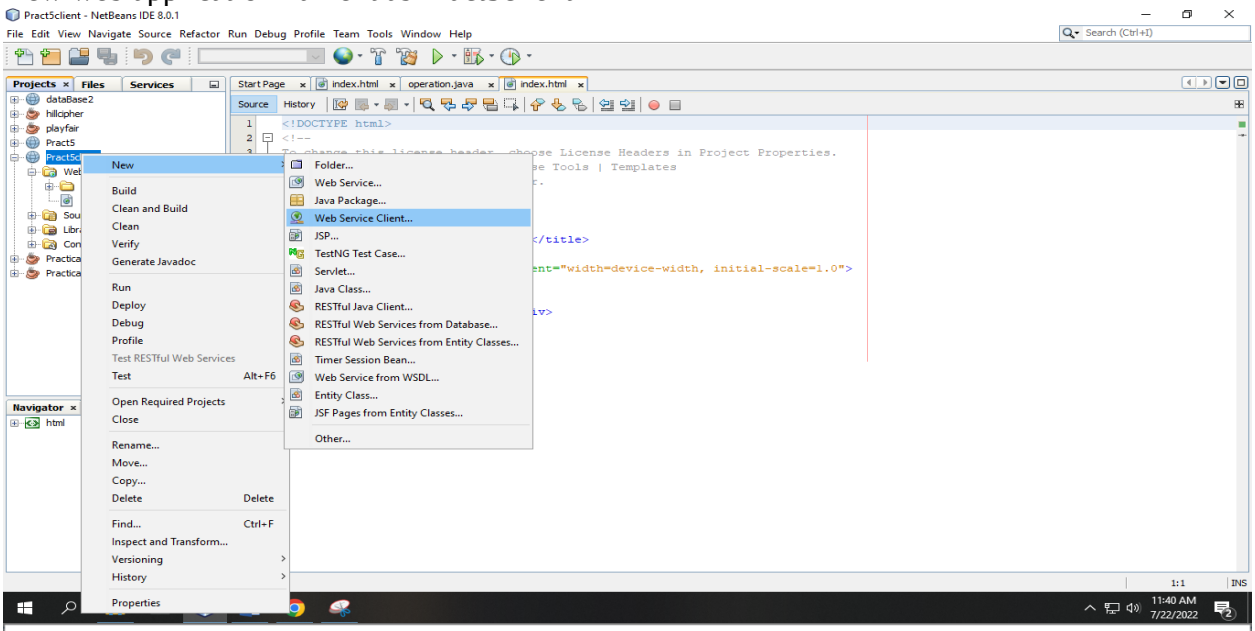

Add Operation
Name: subtraction
Return Type: int
Browse...
Parameters
Exceptions
Name
Type
Final
a
int
b
int
Add
Remove
Up
Down
OK
Cancel

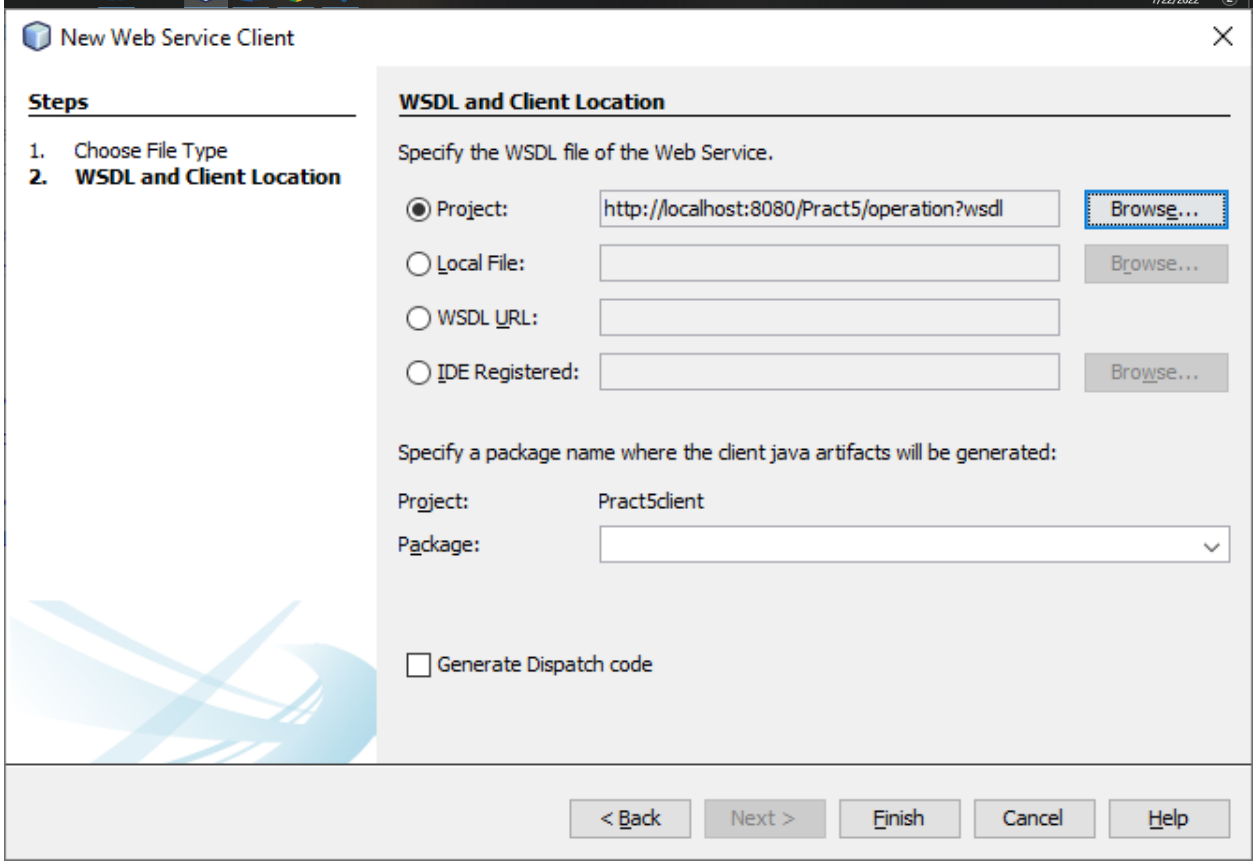

Pract5 - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
dataBase2
hillcipher
playfair
Pract5
Web Pages
WEB-INF
index.html
Source Packages
com.kk
Libraries
Web Services
Configuration Files
Practical4
Practical 6
operation - Navigator
Members
operation
addition(int a, int b) : int
subtraction(int a, int b) : int
Start Page
index.html
operation.java
Source
Design
History
import
/**
@WebServi
public cl
return 0;
Add Operation
Name: multiplication
Return Type: int
Browse...
Parameters
Exceptions
Name
Type
Final
a
int
b
int
Add
Remove
Up
Down
OK
Cancel
Output
19:3
INS
11:35 AM
7/22/2022

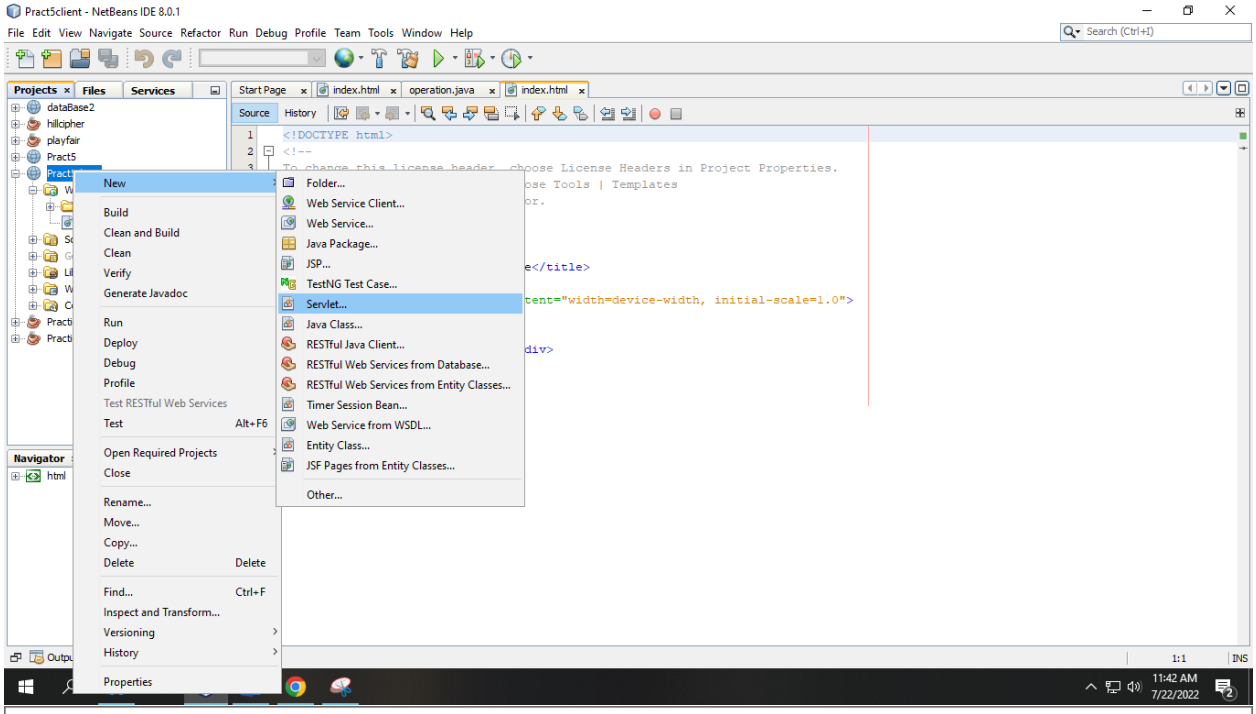

Add Operation
Name: division
Return Type: int
Browse...
Parameters
Exceptions
Name
Type
Final
a
int
b
int
Add
Remove
Up
Down
OK
Cancel

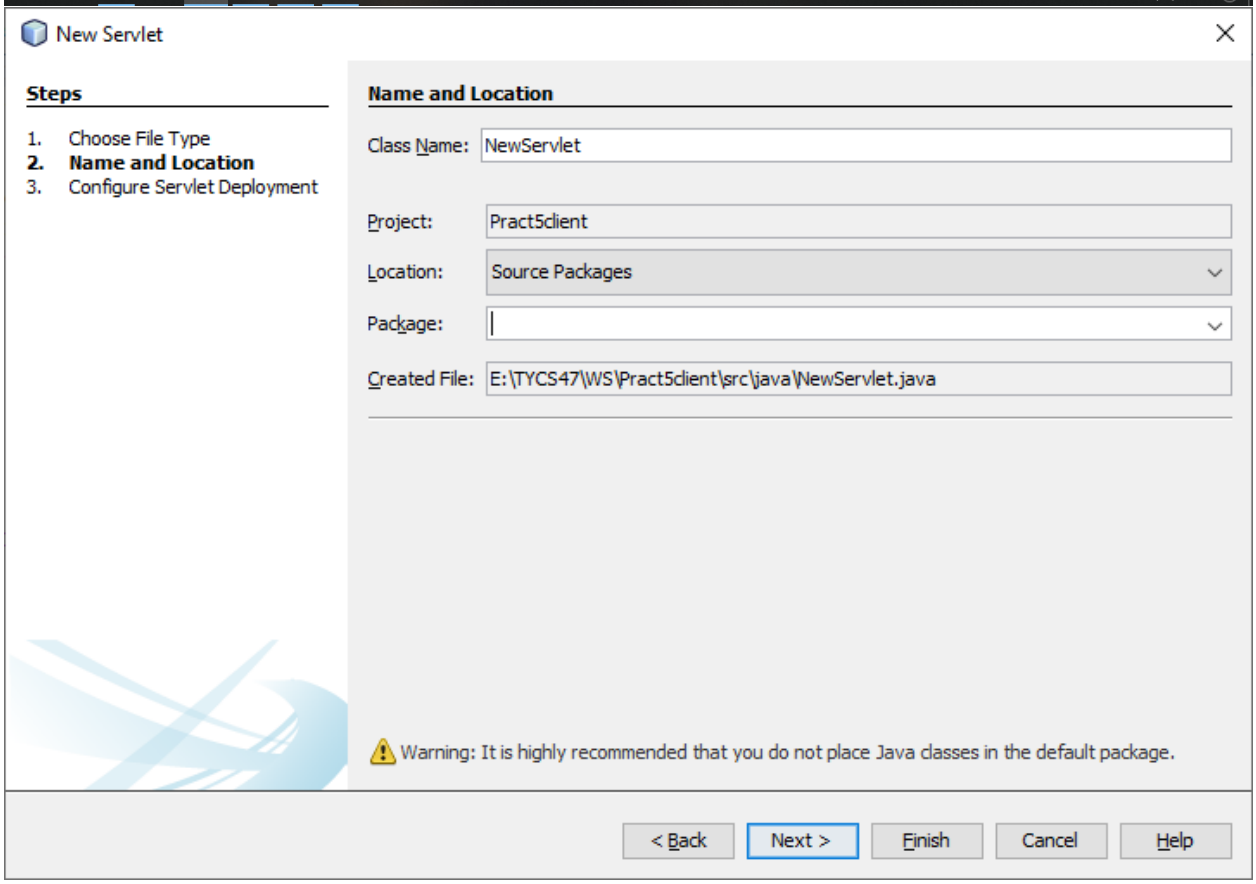


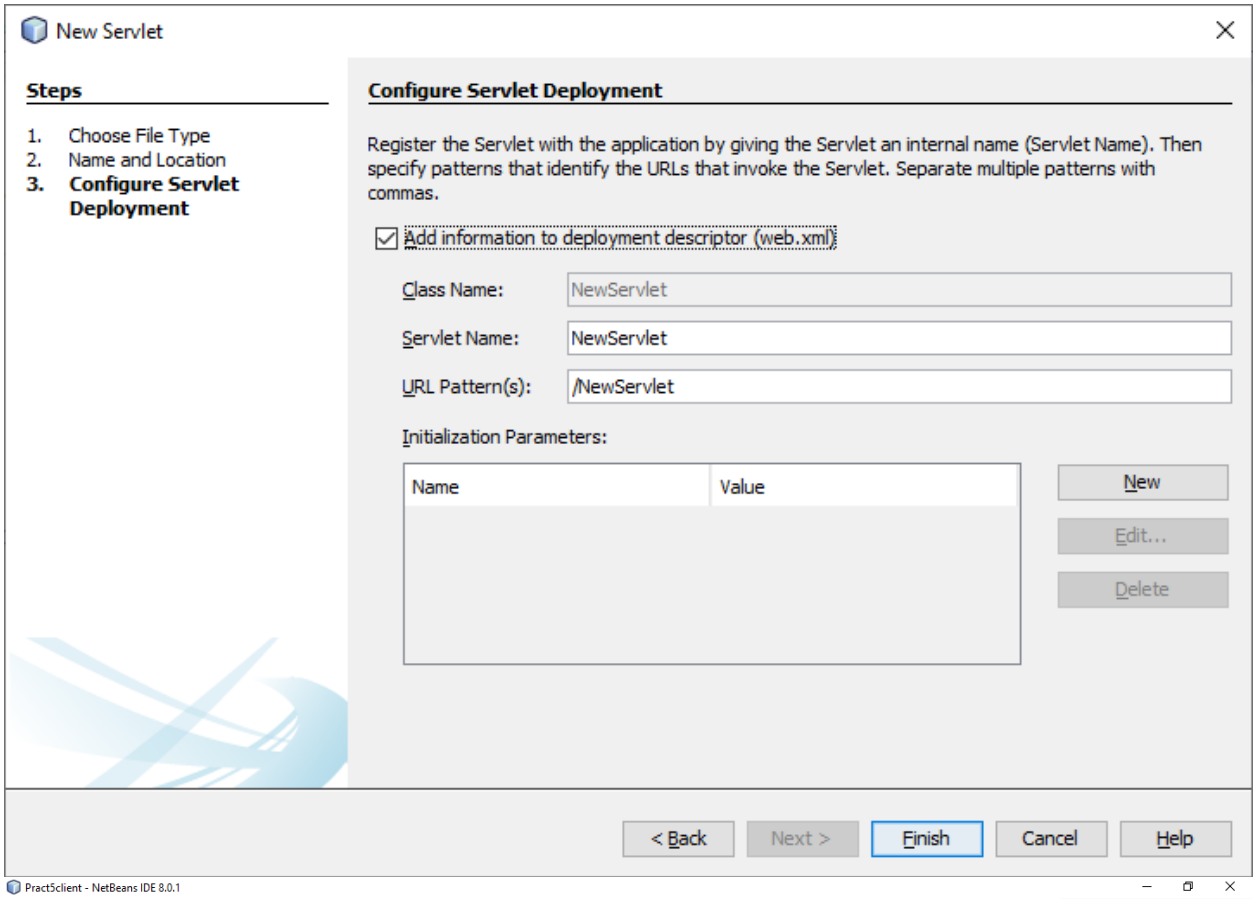
New web application name it as Pract5client

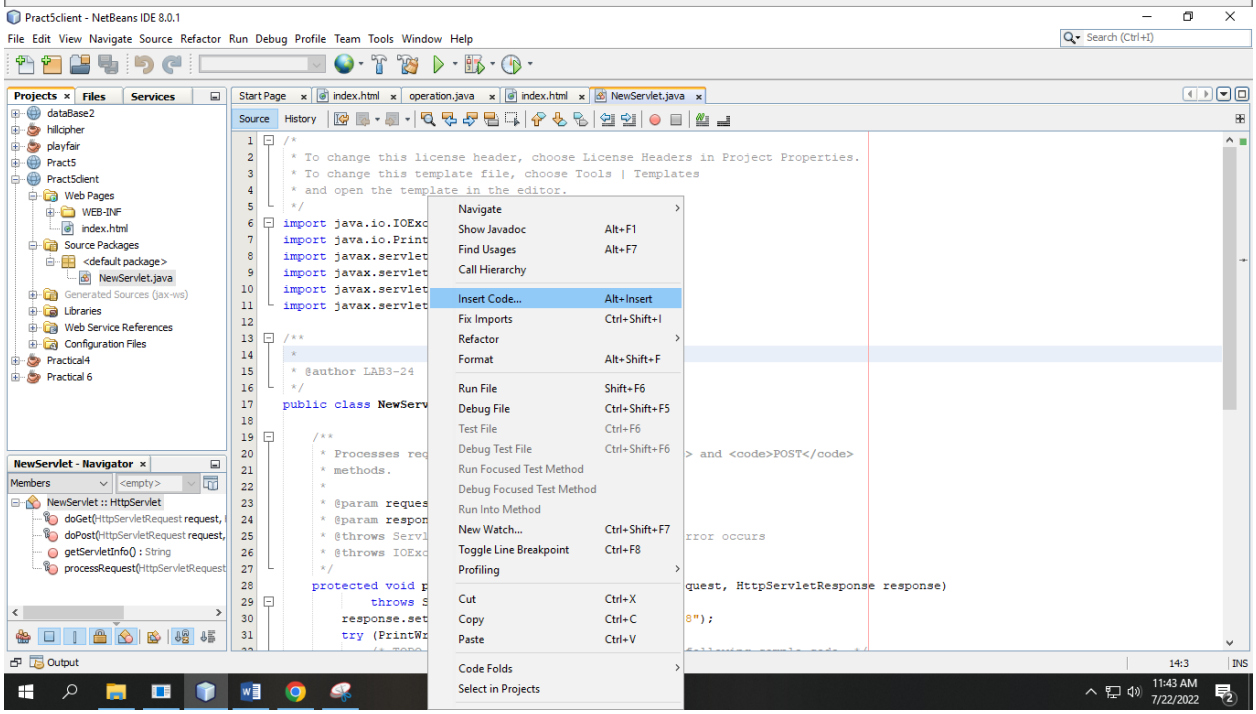


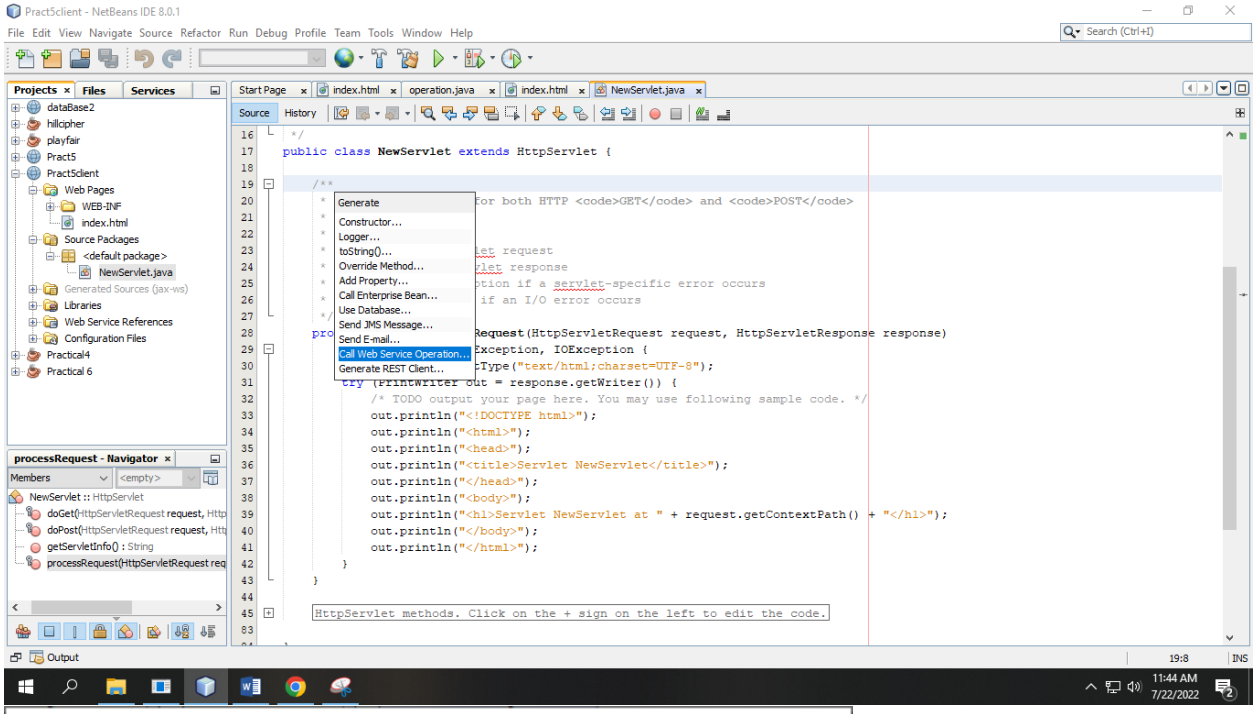


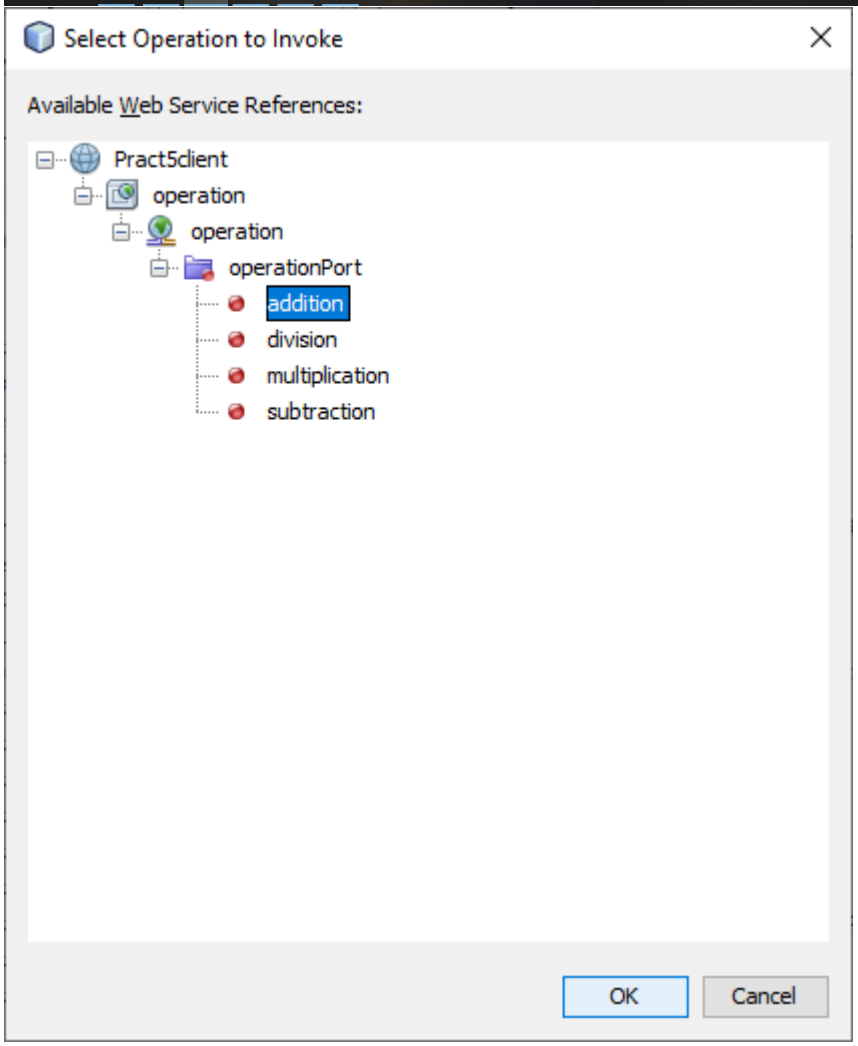

Pract5client - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
dataBase2
hillcipher
playfair
Pract5
Start Page
index.html
operation.java
index.html
Source
History
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
Web Service Client...
Web Service...
Java Package...
JSP...
TestNG Test Case...
Servlet...
Java Class...
RESTful Java Client...
RESTful Web Services from Database...
RESTful Web Services from Entity Classes...
Timer Session Bean...
Web Service from WSDL...
Entity Class...
JSF Pages from Entity Classes...
Other...
Navigator
html
11:42 AM
7/22/2022

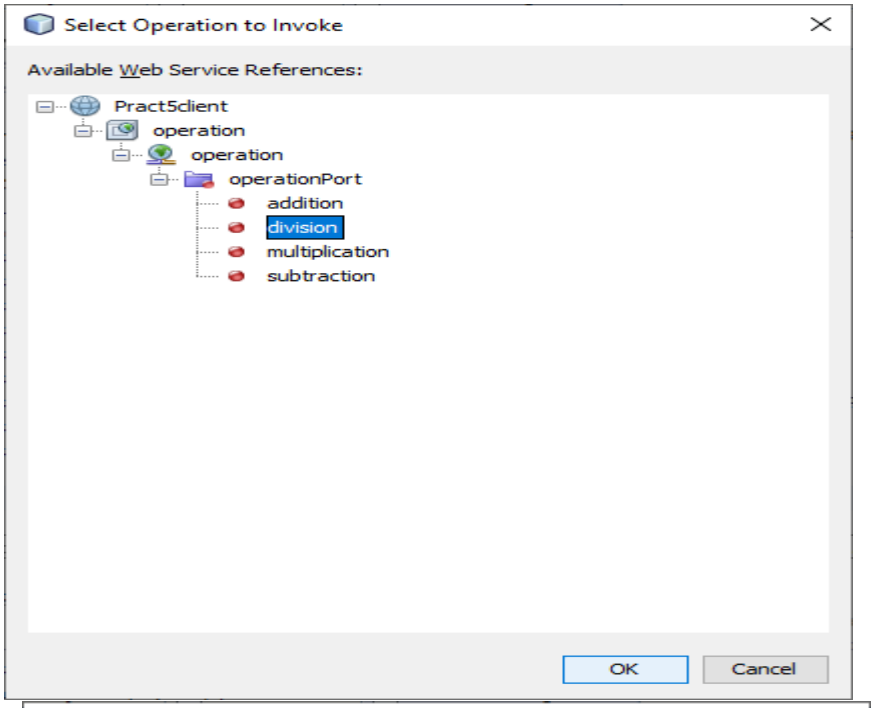

New Servlet
Steps
1. Choose File Type
2. Name and Location
3. Configure Servlet Deployment
Name and Location
Class Name: NewServlet
Project: Pract5client
Location: Source Packages
Package:
Created File: E:\TYCS47\WS\Pract5client\src\java\NewServlet.java
Warning: It is highly recommended that you do not place Java classes in the default package.
< Back
Next >
Finish
Cancel
Help

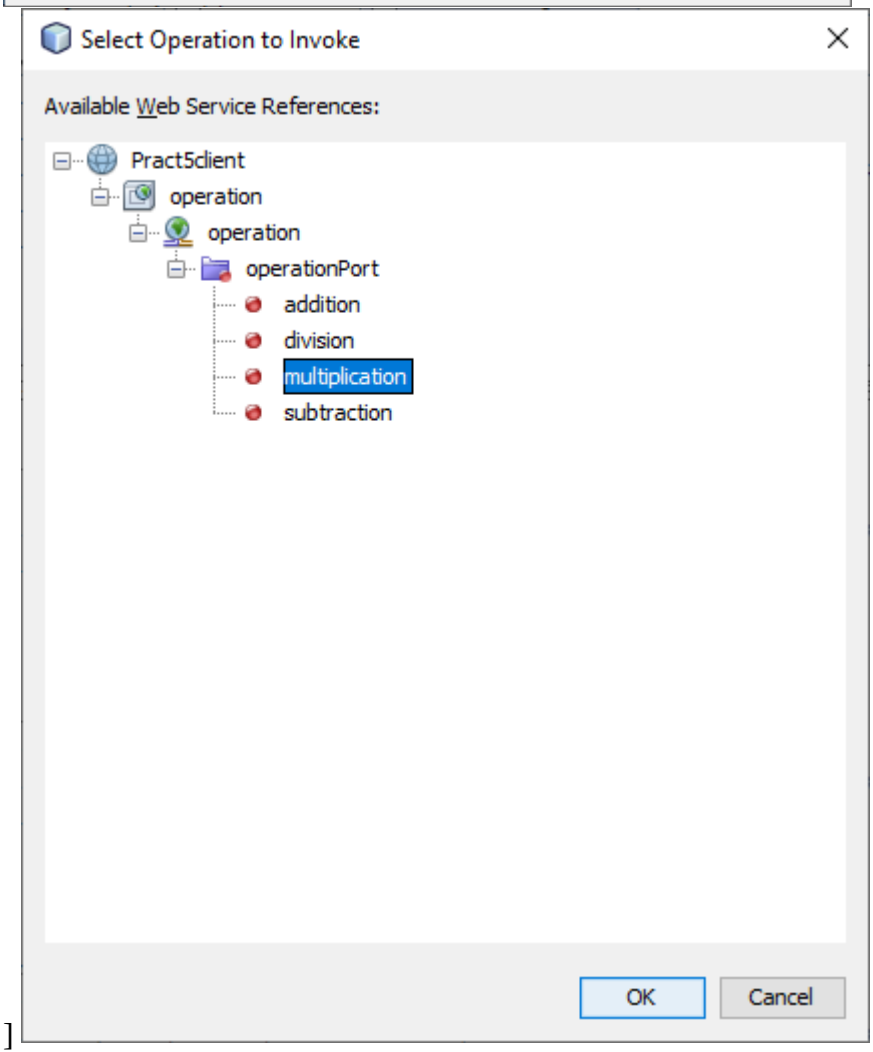

New Servlet
Steps
1. Choose File Type
2. Name and Location
3. Configure Servlet Deployment
Configure Servlet Deployment
Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.
Add information to deployment descriptor (web.xml)
Class Name: NewServlet
Servlet Name: NewServlet
URL Pattern(s): /NewServlet
Initialization Parameters:
Name
Value
New
Edit...
Delete
< Back
Next >
Finish
Cancel
Help


Pract5client - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
dataBase2
hillcipher
playfair
Pract5
Pract5client
Web Pages
WEB-INF
index.html
Source Packages
<default package>
NewServlet.java
Generated Sources (jax-ws)
Libraries
Web Service References
Configuration Files
Practical4
Practical 6
NewServlet - Navigator
NewServlet :: HttpServlet
Start Page
index.html
operation.java
NewServlet.java
Source
History
Navigate
Show Javadoc Alt+F1
Find Usages Alt+F7
Call Hierarchy
Insert Code... Alt+Insert
Fix Imports Ctrl+Shift+I
Refactor
Format Alt+Shift+F
Run File Shift+F6
Debug File Ctrl+Shift+F5
Test File Ctrl+F6
Debug Test File Ctrl+Shift+F6
Run Focused Test Method
Debug Focused Test Method
Run Into Method
New Watch... Ctrl+Shift+F7
Toggle Line Breakpoint Ctrl+F8
Profiling
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Code Folds
Select in Projects
Output
14:3
INS
11:43 AM
7/22/2022

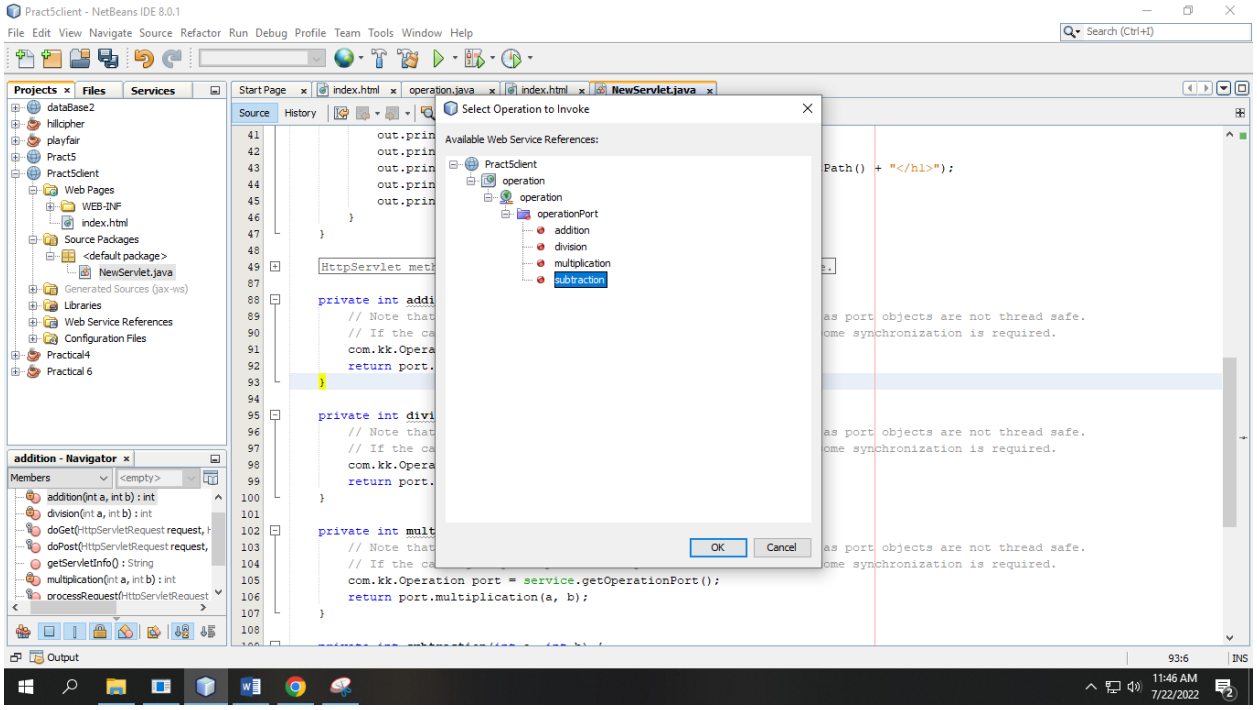

Pract5client - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
dataBase2
hillcipher
playfair
Pract5
Pract5client
Web Pages
WEB-INF
index.html
Source Packages
<default package>
NewServlet.java
Generated Sources (jax-ws)
Libraries
Web Service References
Configuration Files
Practical4
Practical 6
processRequest - Navigator
NewServlet :: HttpServlet
doGet(HttpServletRequest request, Http
doPost(HttpServletRequest request, Htt
getServletInfo() : String
processRequest(HttpServletRequest req
Start Page
index.html
operation.java
index.html
NewServlet.java
Source History
public class NewServlet extends HttpServlet {
Generate
Constructor...
Logger...
toString()...
Override Method...
Add Property...
Call Enterprise Bean...
Use Database...
Send JMS Message...
Send E-mail...
Call Web Service Operation...
Generate REST Client...
HttpServlet methods. Click on the + sign on the left to edit the code.
Output
19:8
INS
11:44 AM
7/22/2022


Select Operation to Invoke
Available Web Service References:
Pract5client
operation
operation
operationPort
addition
division
multiplication
subtraction
OK
Cancel


Select Operation to Invoke
Available Web Service References:
Pract5client
operation
operation
operationPort
addition
division
multiplication
subtraction
OK
Cancel


Select Operation to Invoke
Available Web Service References:
Pract5client
operation
operation
operationPort
addition
division
multiplication
subtraction
OK
Cancel

]



Source Code

Index.html

```
<!DOCTYPE html>
<!--
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.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action ="NewServlet" method ="post">
            <table>
                <tr><td>Enter First Number: </td><td><input type ="text" name="a"></td></tr>
                <tr><td>Enter Second Number: </td><td><input type ="text"
name="b"></td></tr>
                <tr><td></td><td><input type ="submit" value="Calculate"/></td></tr>
            </table>
        </form>
    </body>
</html>
```

operation.java

```
package com.kk;

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

```
@WebService(serviceName = "operation")
public class operation {
    @WebMethod(operationName = "addition")
    public int addition(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
        return a+b;
    }

    @WebMethod(operationName = "subtraction")
    public int subtraction(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
        return a-b;
    }

    @WebMethod(operationName = "multiplication")
    public int multiplication(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
        return a*b;
    }

    @WebMethod(operationName = "division")
    public int division(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
        return a/b;
    }
}
```

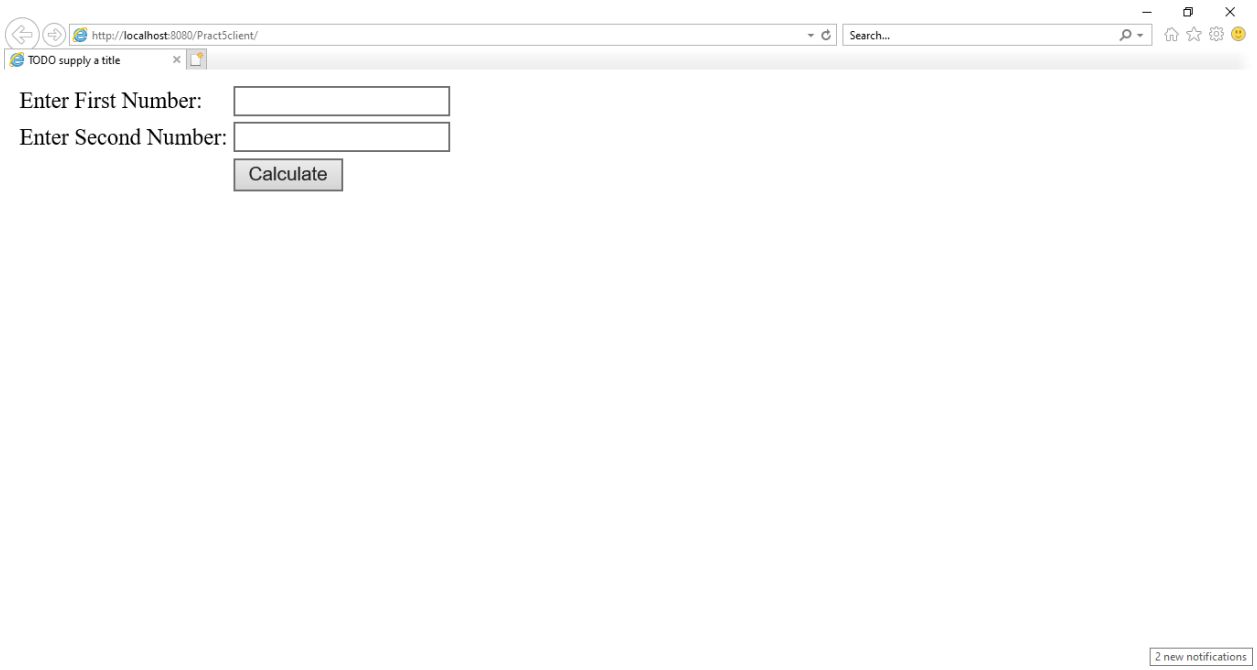
NewServlet.java

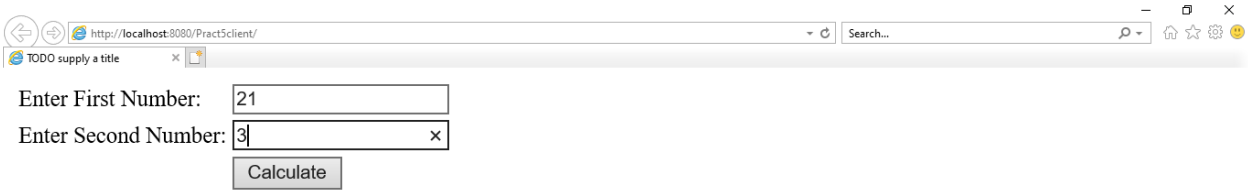
```
import com.kk.Operation_Service;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.ws.WebServiceRef;

public class NewServlet extends HttpServlet {
    @WebServiceRef(wsdlLocation = "WEB-
INF/wsdl/localhost_8080/Pract5/operation.wsdl")
    private Operation_Service service;

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet NewServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1> ADDITION IS
"+addition(Integer.parseInt(request.getParameter("a")),
                    Integer.parseInt(request.getParameter("b")))+"</h1>");
```

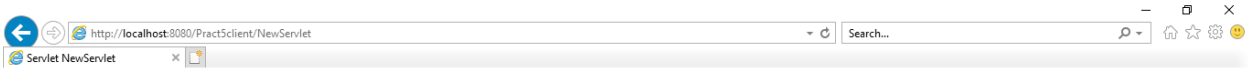
```
            out.println("<h1> SUBTRACTION IS
"+subtraction(Integer.parseInt(request.getParameter("a")),
                    Integer.parseInt(request.getParameter("b")))+"</h1>");
            out.println("<h1> MULTIPLICATION IS
"+multiplication(Integer.parseInt(request.getParameter("a")),
                    Integer.parseInt(request.getParameter("b")))+"</h1>");
            out.println("<h1> DIVISION IS
"+division(Integer.parseInt(request.getParameter("a")),
                    Integer.parseInt(request.getParameter("b")))+"</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
@Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

    private int addition(int a, int b) {
        // Note that the injected javax.xml.ws.Service reference as well as port objects are not
thread safe.
        // If the calling of port operations may lead to race condition some synchronization is
required.
        com.kk.Operation port = service.getOperationPort();
        return port.addition(a, b);
    }

    private int division(int a, int b) {
        // Note that the injected javax.xml.ws.Service reference as well as port objects are not
thread safe.
        // If the calling of port operations may lead to race condition some synchronization is
required.
        com.kk.Operation port = service.getOperationPort();
        return port.division(a, b);
    }

    private int multiplication(int a, int b) {
        // Note that the injected javax.xml.ws.Service reference as well as port objects are not
thread safe.
        // If the calling of port operations may lead to race condition some synchronization is
```

```
required.
        com.kk.Operation port = service.getOperationPort();
        return port.multiplication(a, b);
    }

    private int subtraction(int a, int b) {
        // Note that the injected javax.xml.ws.Service reference as well as port objects are not
thread safe.
        // If the calling of port operations may lead to race condition some synchronization is
required.
        com.kk.Operation port = service.getOperationPort();
        return port.subtraction(a, b);
    }

}
```

Output:







ADDITION IS 24

SUBTRACTION IS 18

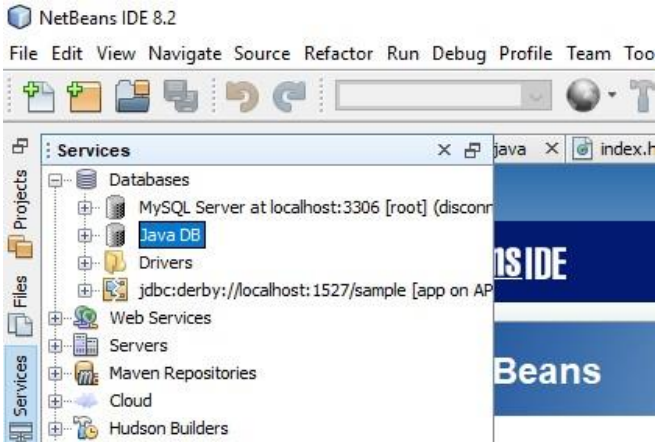
MULTIPLICATION IS 63

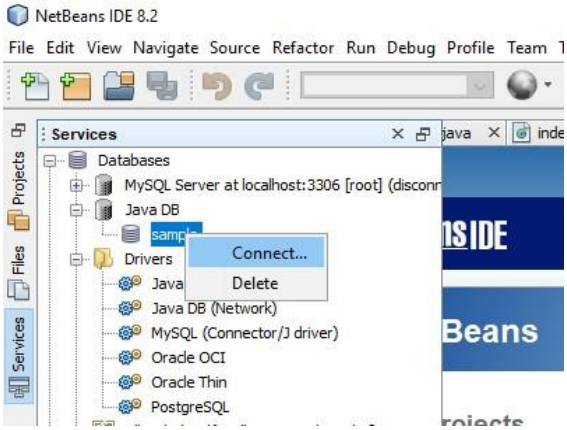
DIVISION IS 7

Practical 5

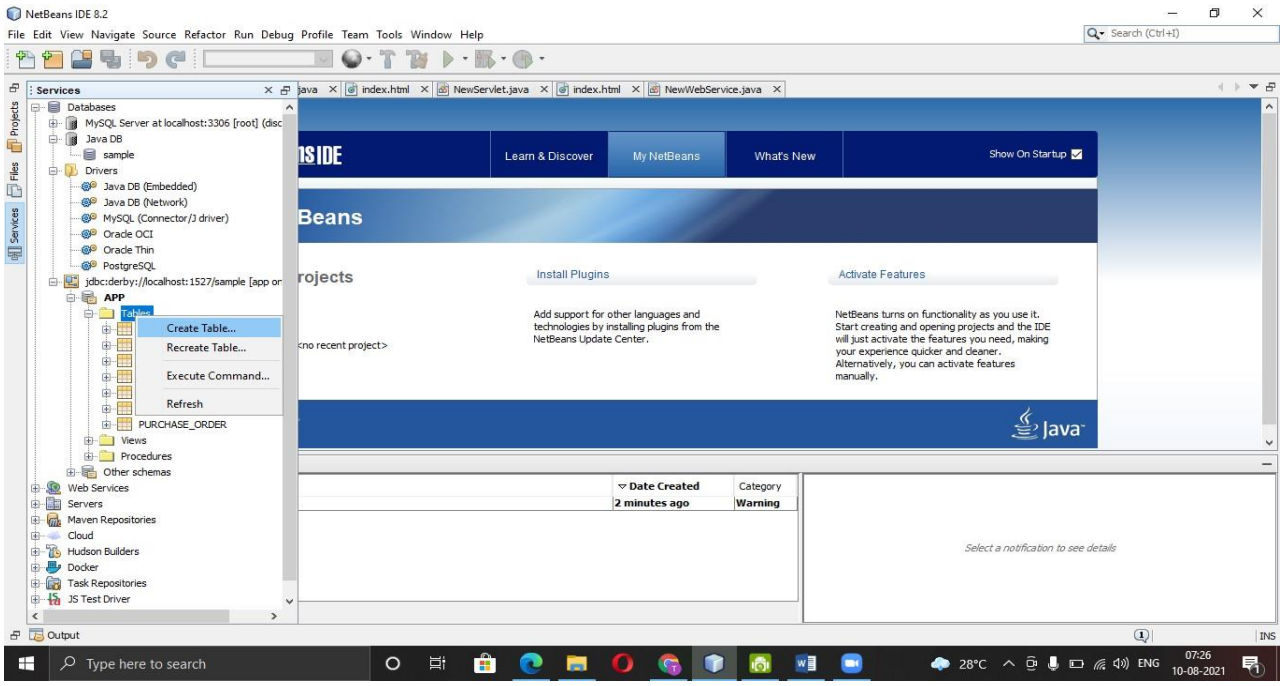
AIM : Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.

STEP 1: Go to Services. Select Databases – JavaDB. Click on it. Start the Server and connect the Sample database

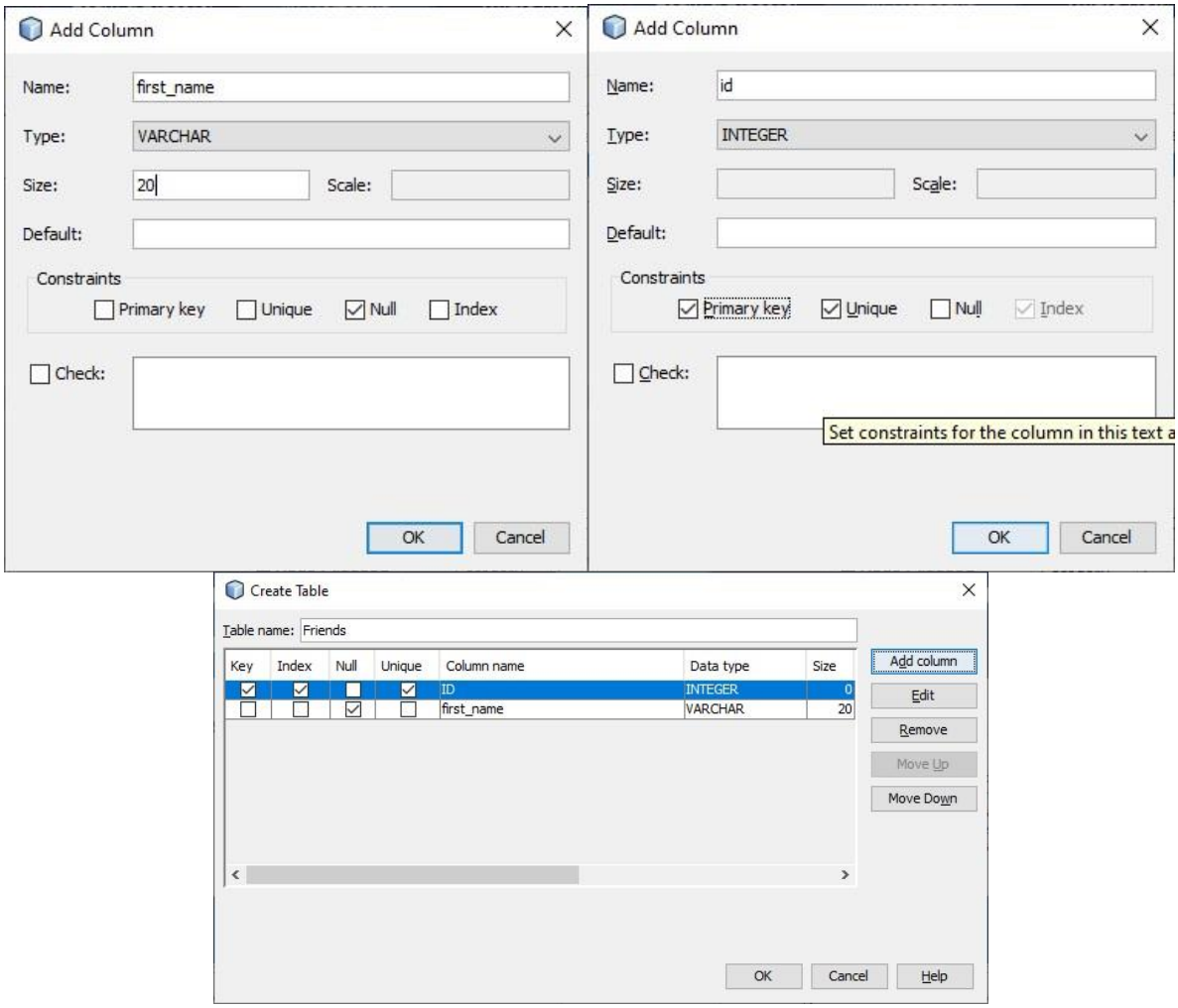




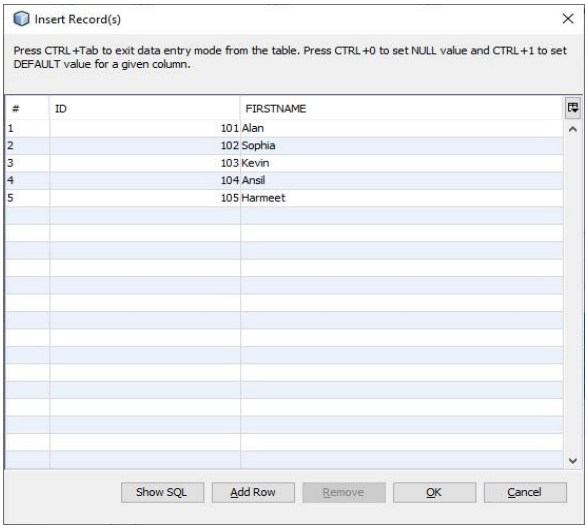
STEP 2: Expand the url below. Select APP – Tables –Right Click to Create Table. Name the table as Friends



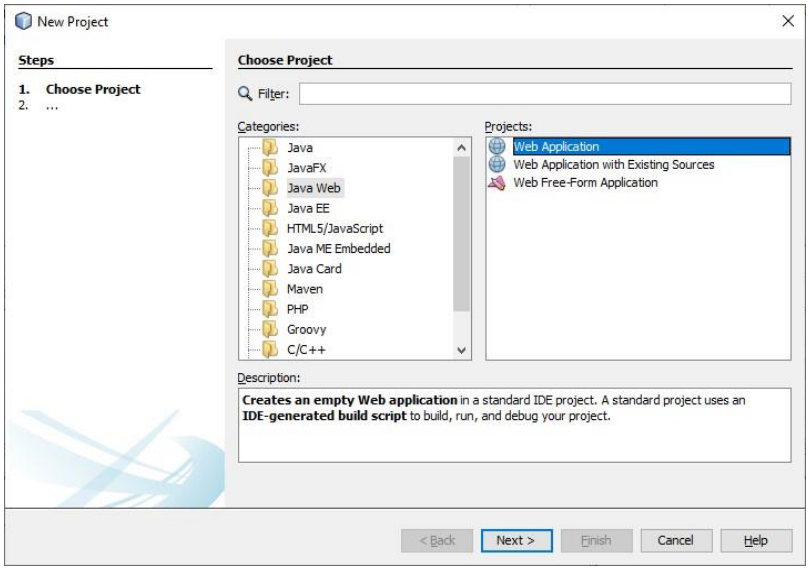
STEP 3: Now click on Add Column. Enter details as in below images. Click OK.

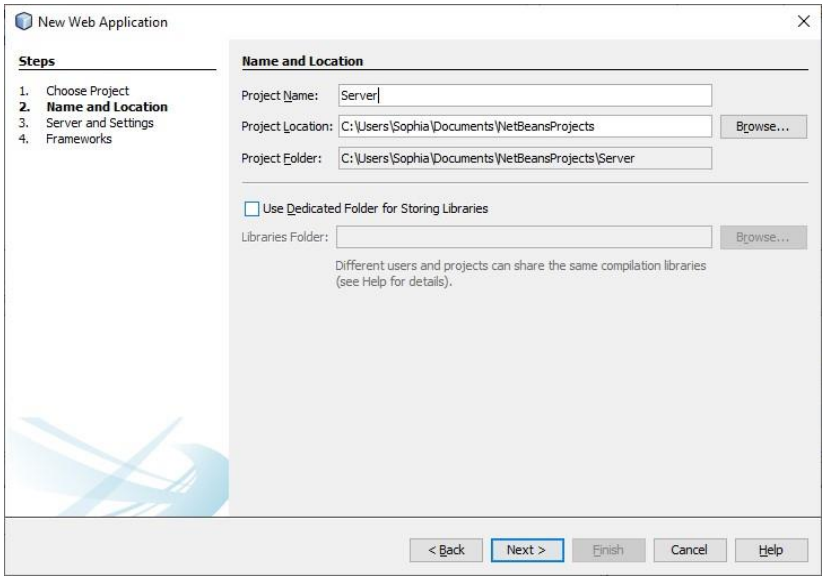


STEP 4: Right click on FRIENDS – View Data then insert records. Insert a record then click on Add Row button to insert more record. Next click OK.

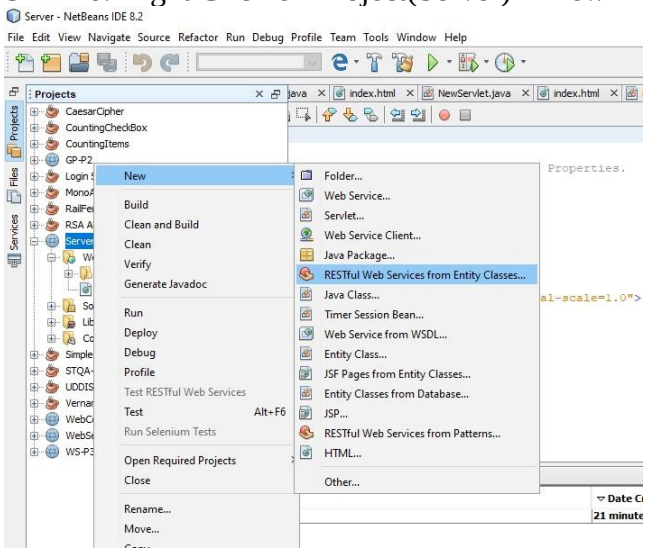


STEP 5: Now create a New Project – Java Web – Web Application with name Server. After that click on Next and then Finish button.

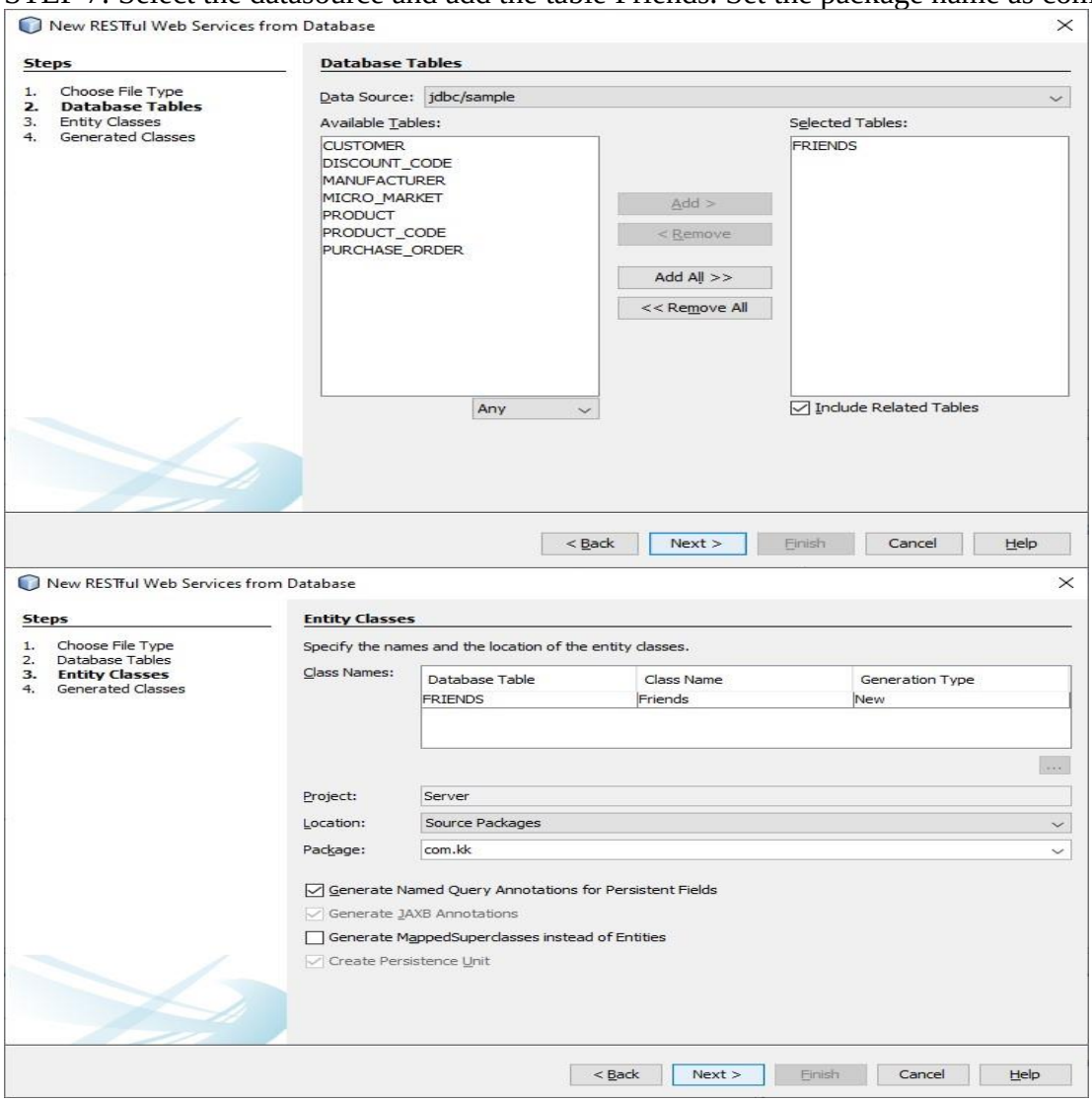




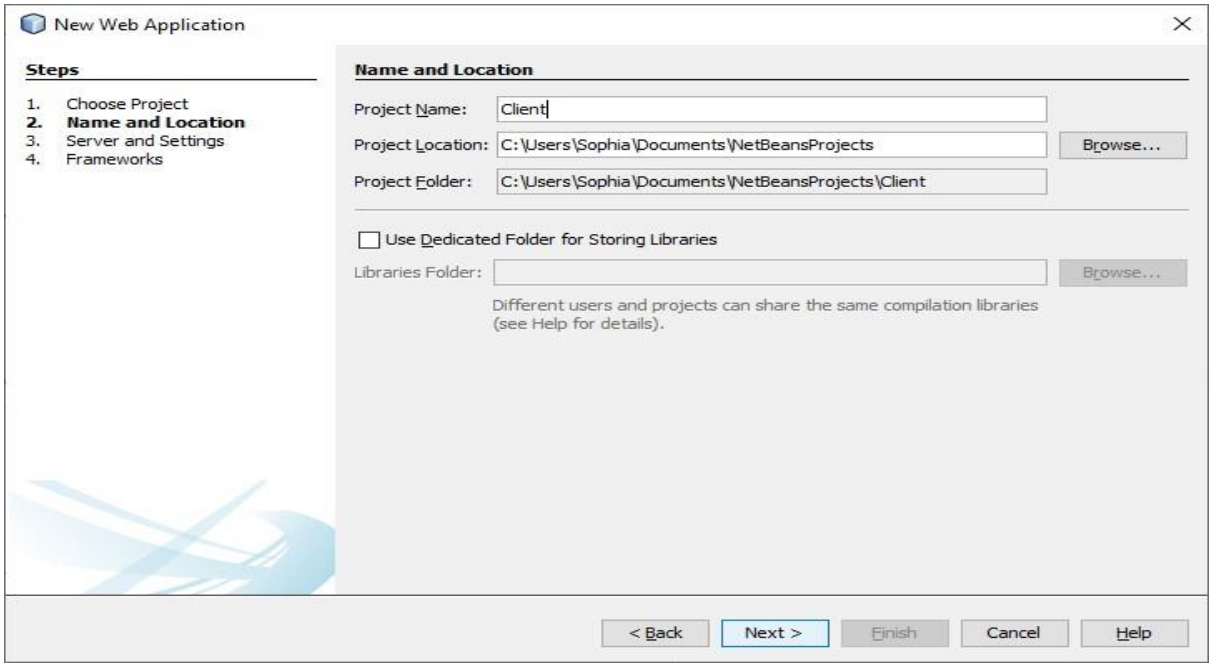
STEP 6: Right Click on Project(Server) - New – RESTful Web Services from Database.



STEP 7: Select the datasource and add the table Friends. Set the package name as com.kk



STEP 8: Now create a New Project – Java Web – Web Application with name Client. After that click on Next and then Finish button.

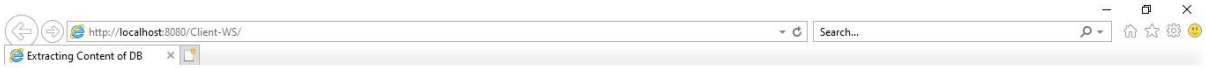


STEP 9: To the index.html page of the Client project add the following code. Once the code is added Deploy the Server project. Next Clean and Build – Run the Client project to see the output

index.html <!DOCTYPE html>

```
<!--
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.
-->
<html>
    <head>
        <title>Extracting Content of DB</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>  table {
 font-family: arial, sans-serif;

 }
 td, th {
 border: 2px solid #000000;  text-align: center;  padding: 10px;
 }
 </style>  <script>
 var request = new XMLHttpRequest();
request.open('GET','http://localhost:8080/Server-WS/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>
</html>
```

OUTPUT:

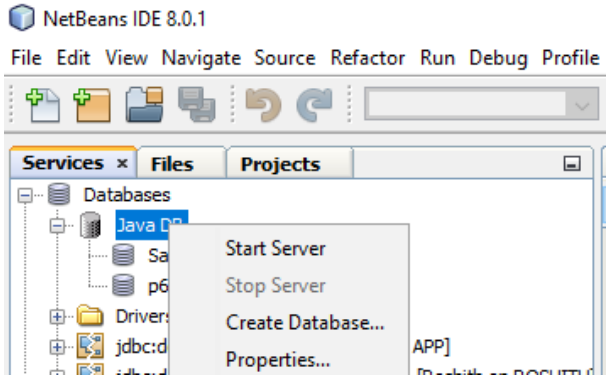


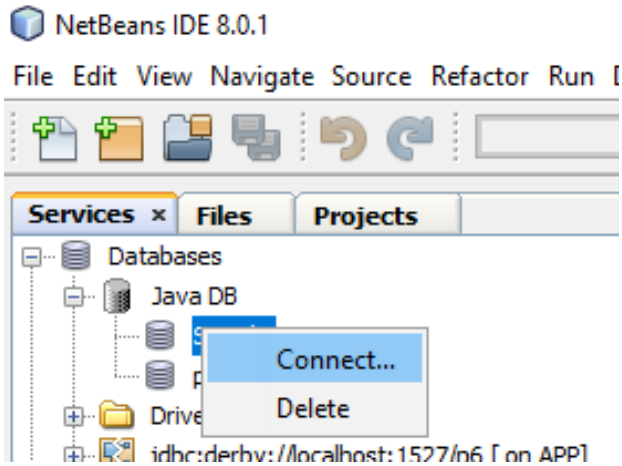
ID	NAME
101	Alan
102	Sophia
103	Kevin
104	Ansil
105	Harmeet

Practical 6

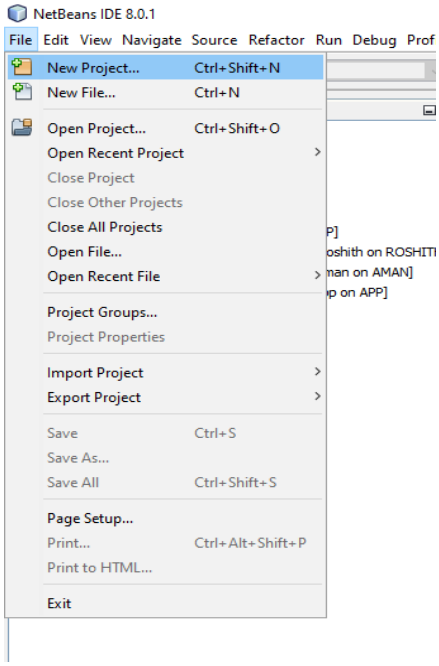
Define a RESTful web service that accepts the details to be stored in a database and performs CRUD operation.

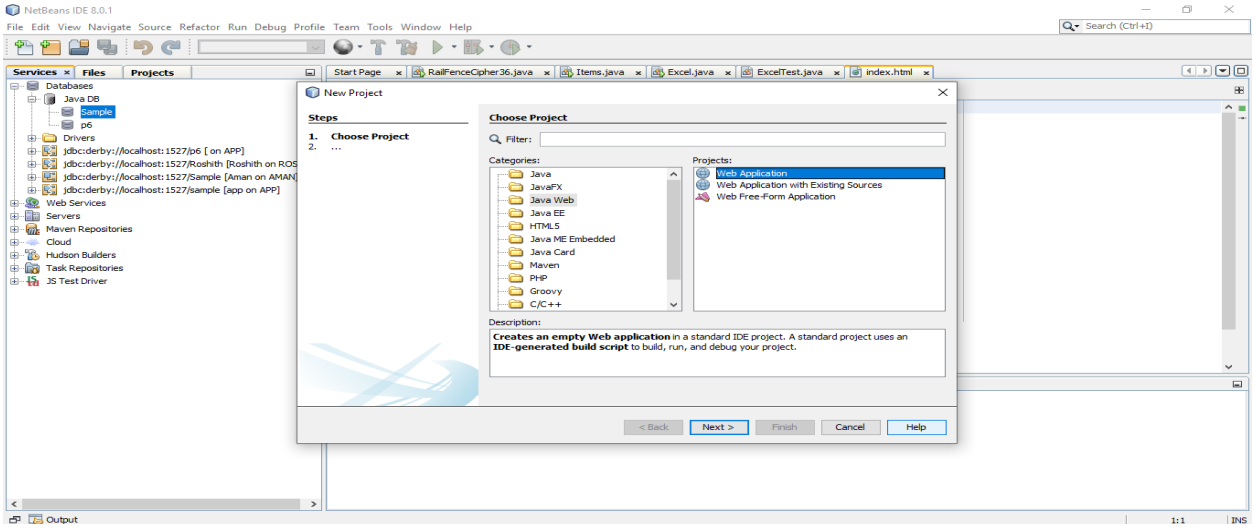
STEP1: Click on Services and expand Database and click on Java DB and expand it and click on Connect.

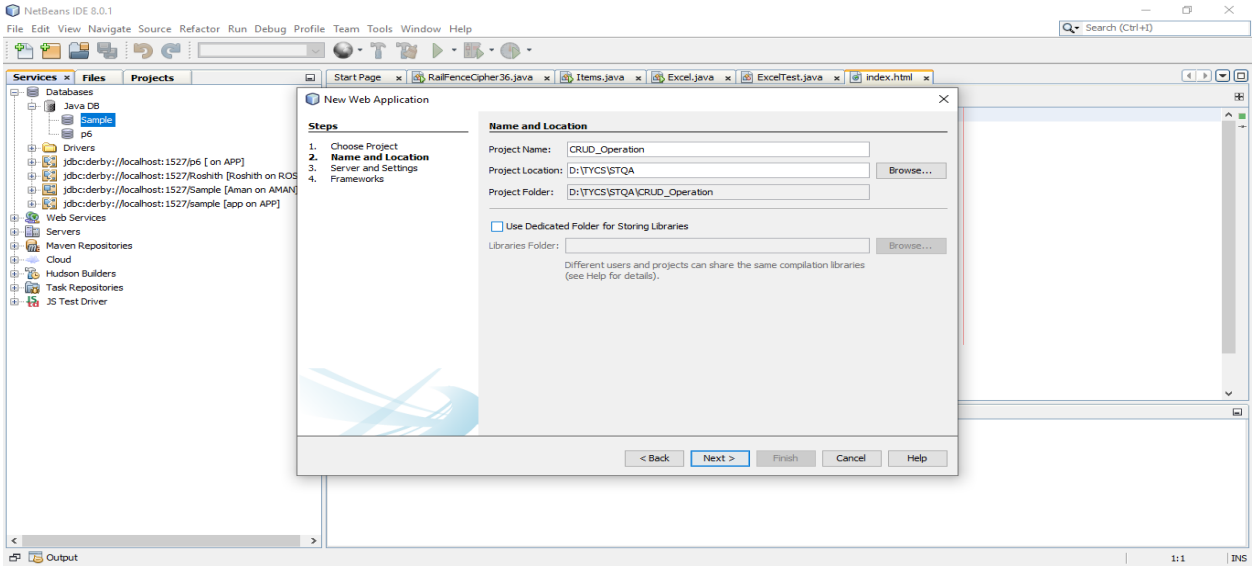




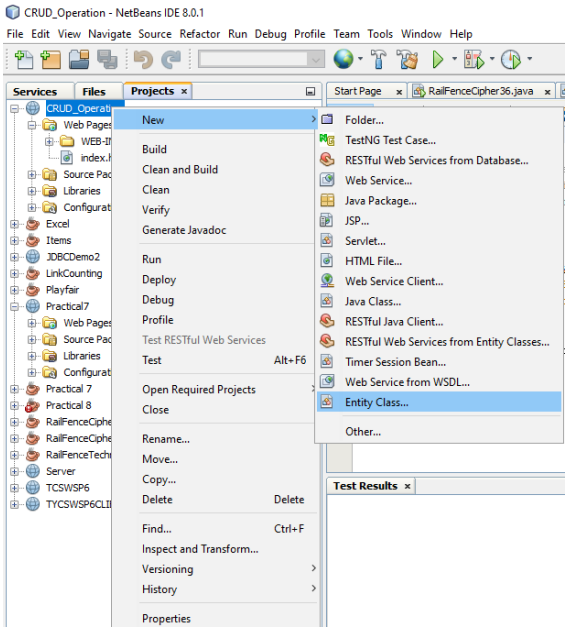
STEP 2: Create a new Java Web Application Project name it click Next and Finish

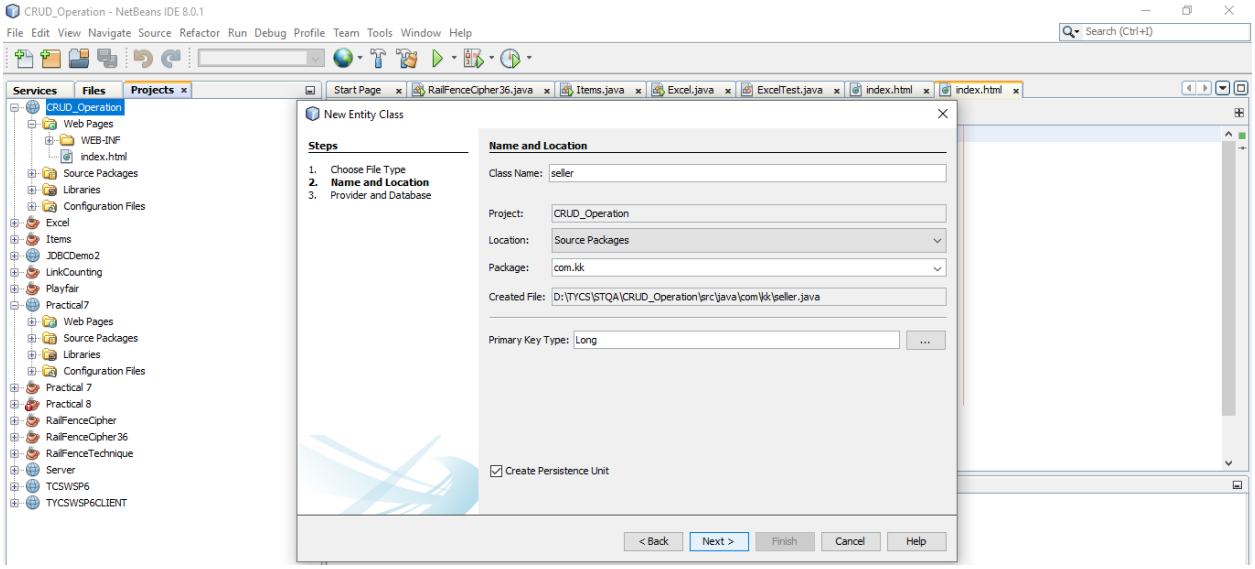


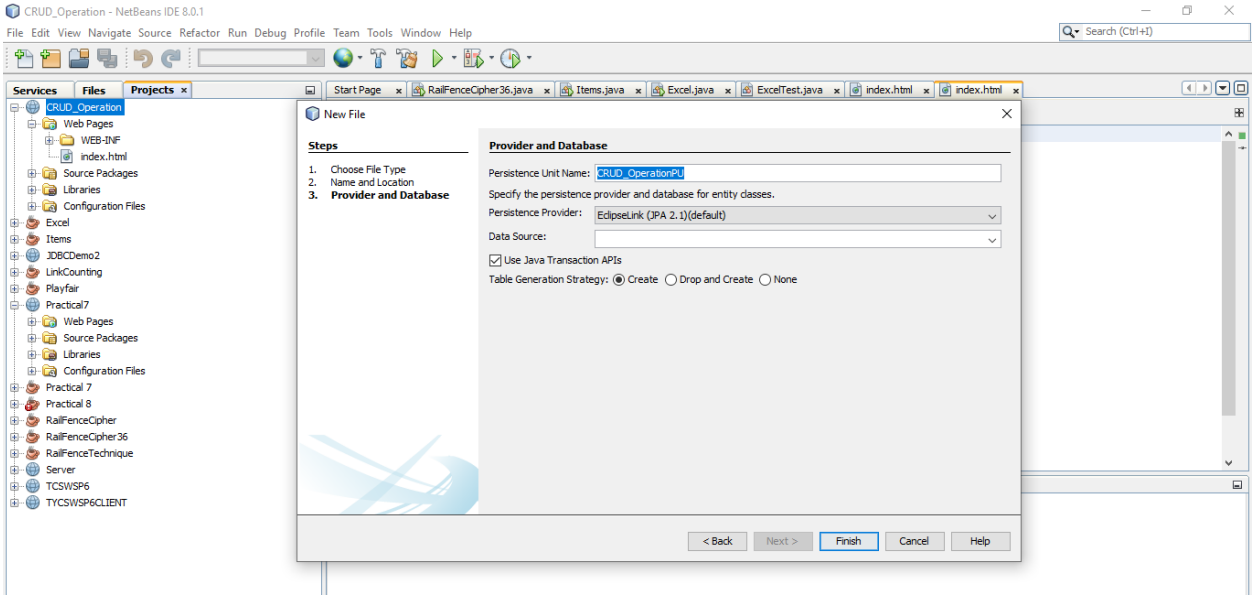




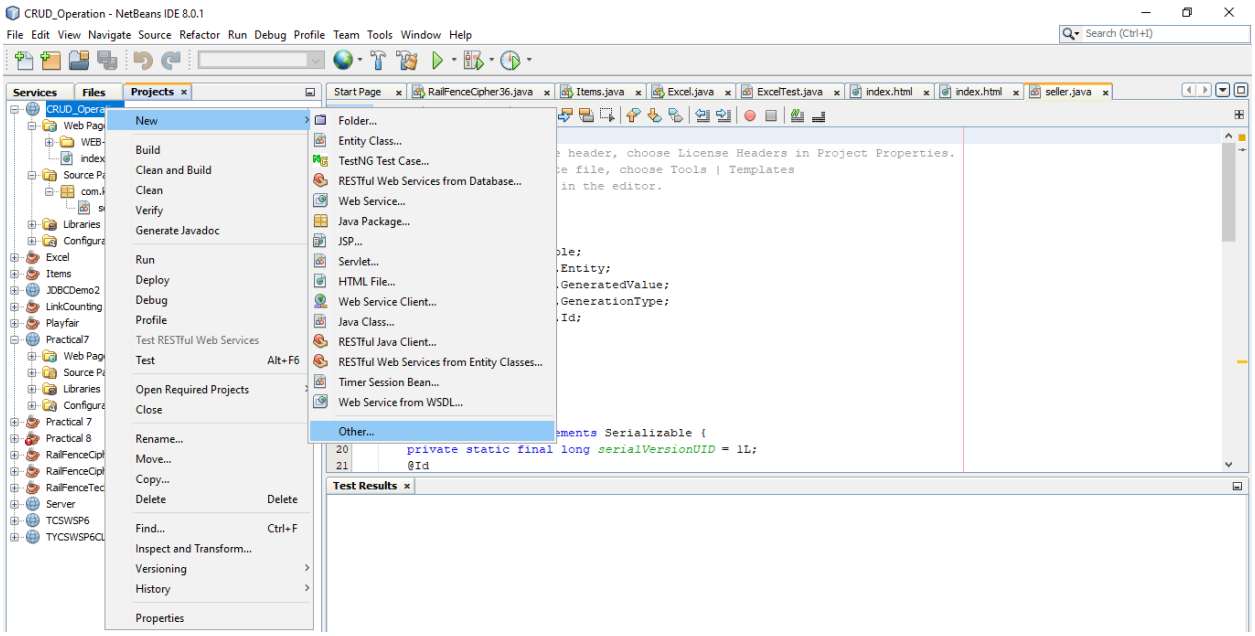
STEP3:Right Click on Project -> New -> Entity Class

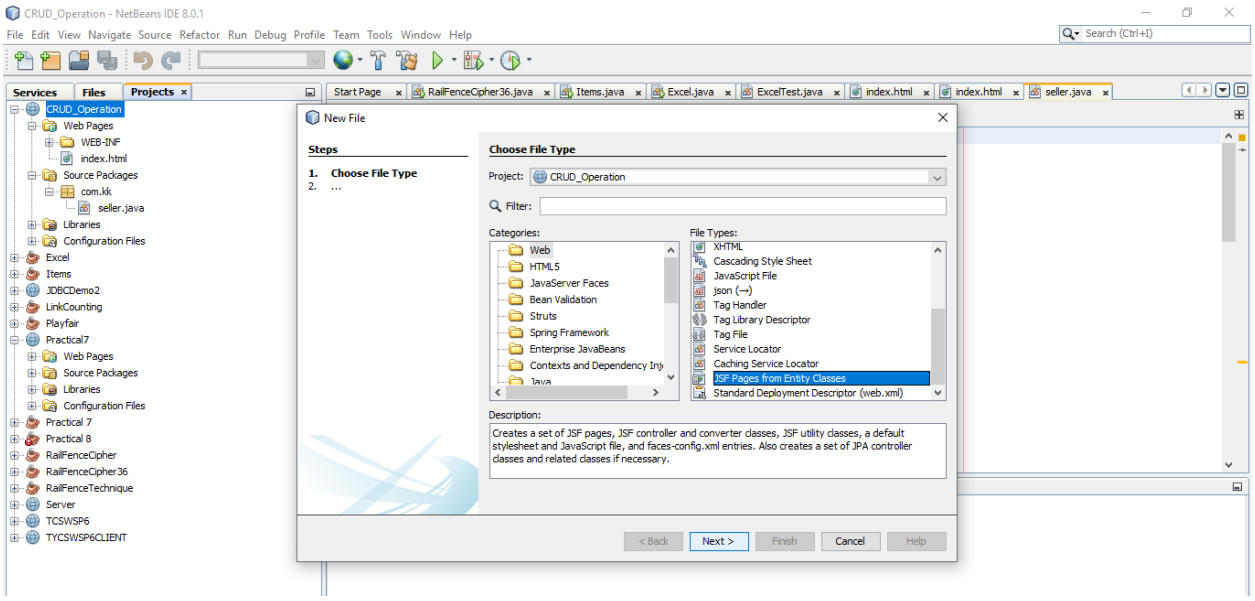




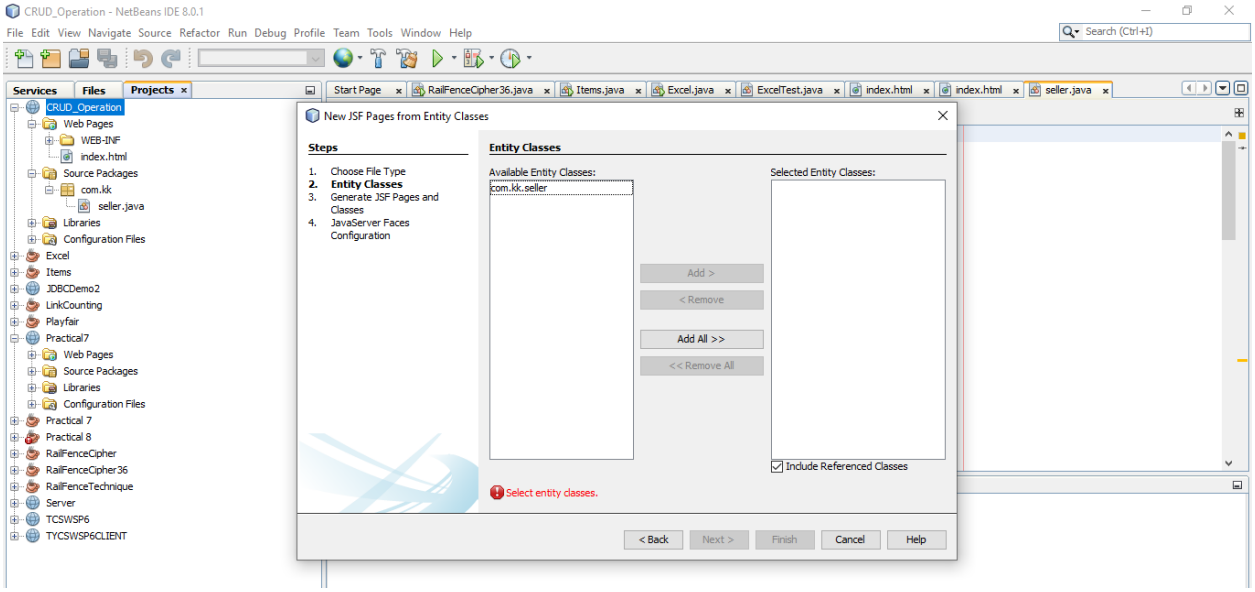


STEP4: Click on Project->New-> Others.Select JSP Page Entity Class.

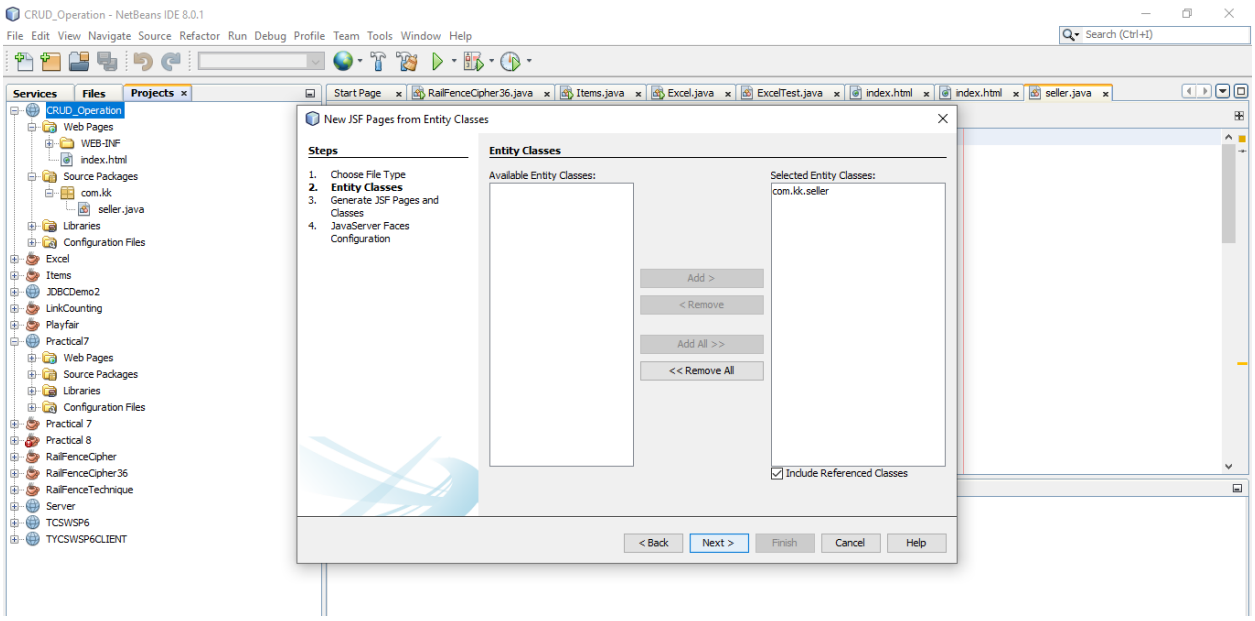




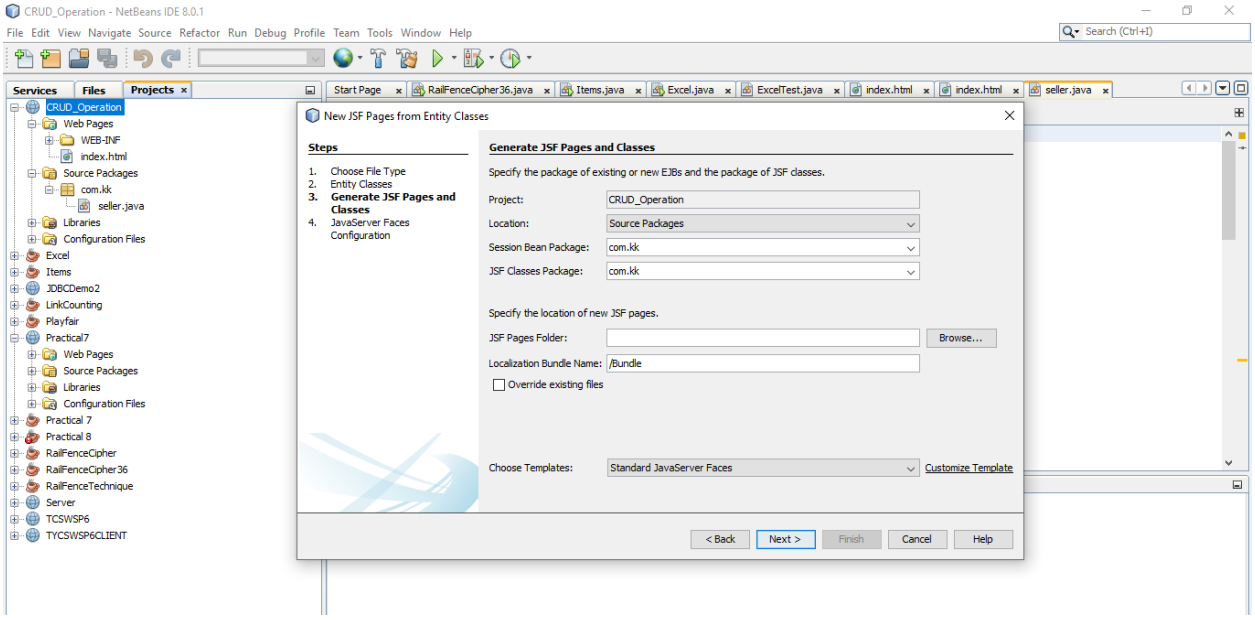
STEP5: Select class and click on add all.



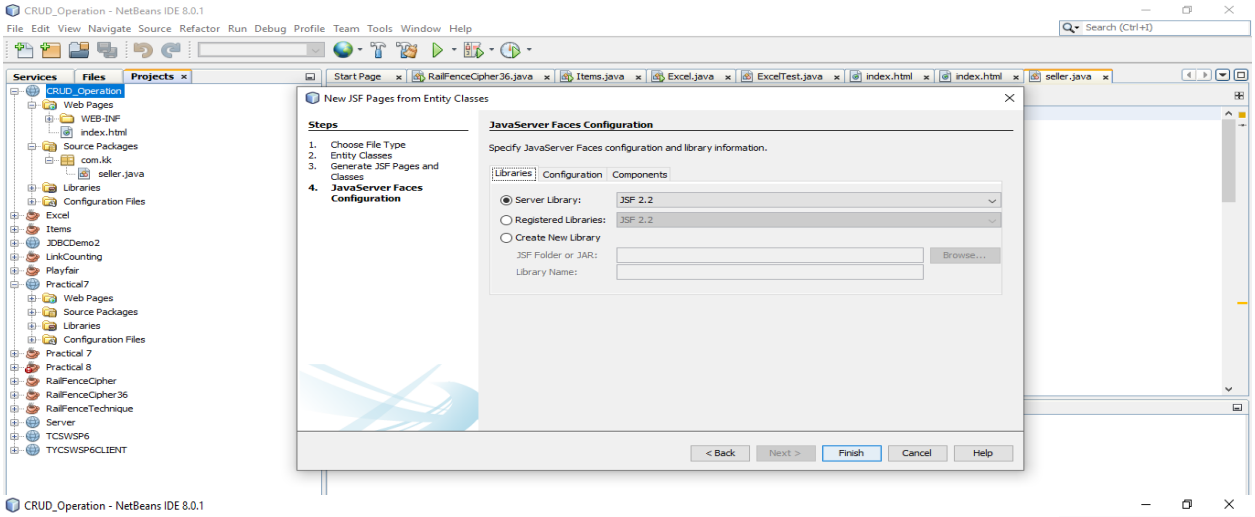
STEP6: Click on next

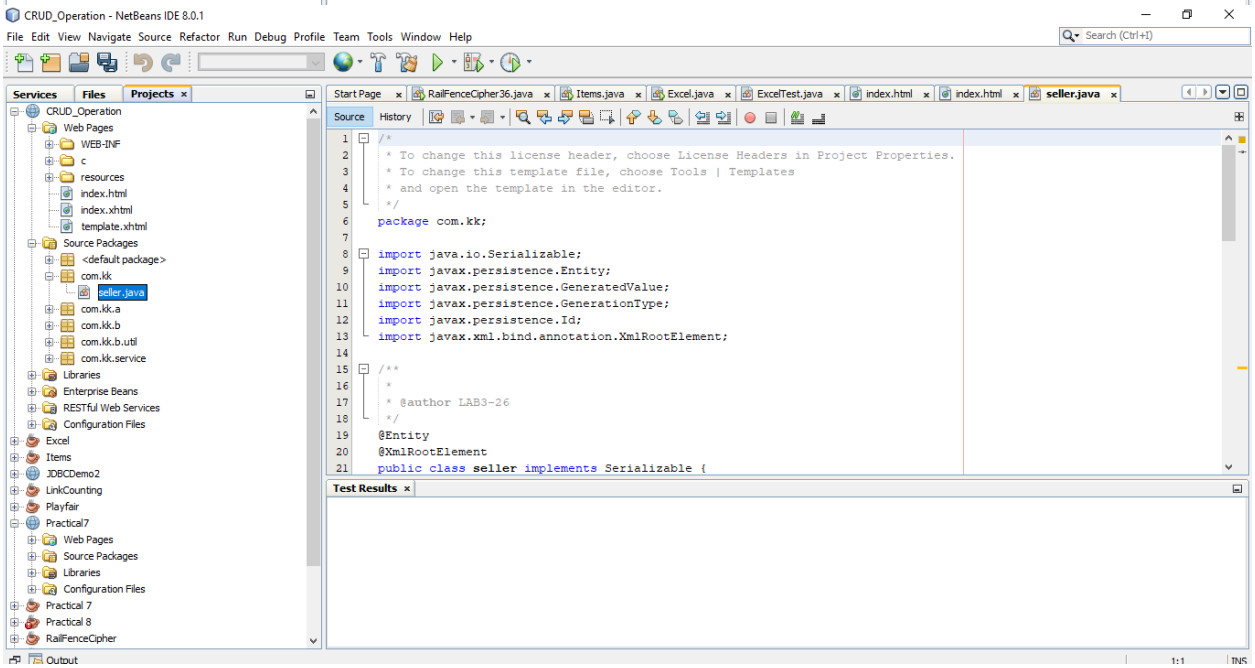


STEP7:Session Bean Package as com.kk and JSF classes pacakage as com.kk and click on next

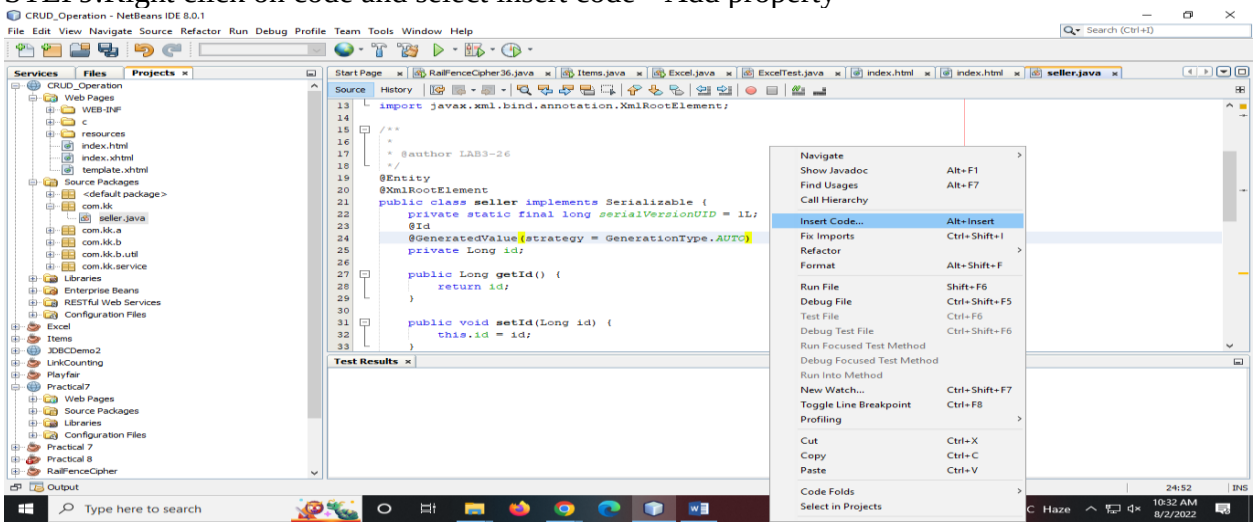


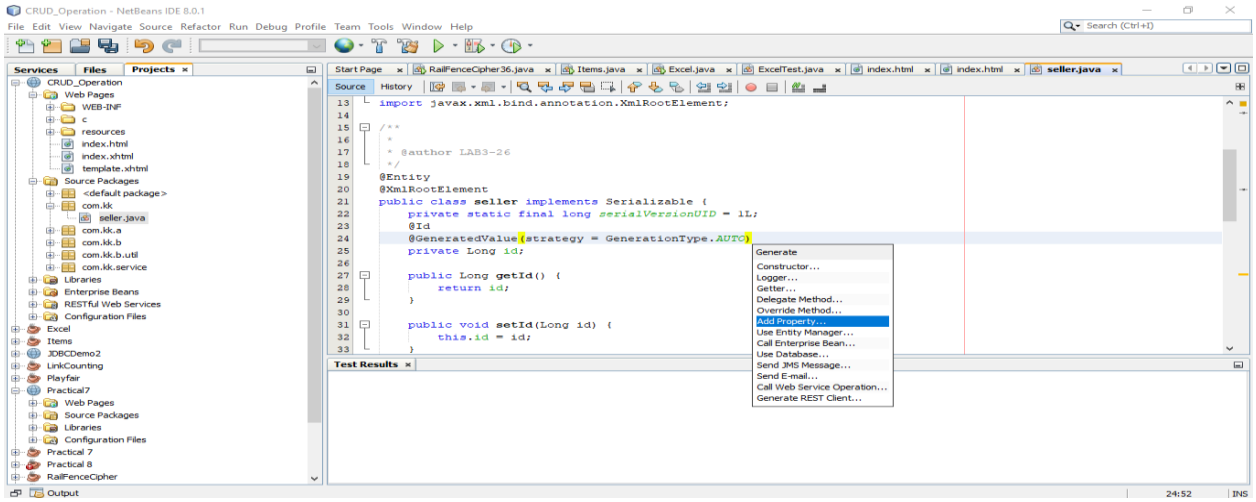
STEP8:Click on Finish



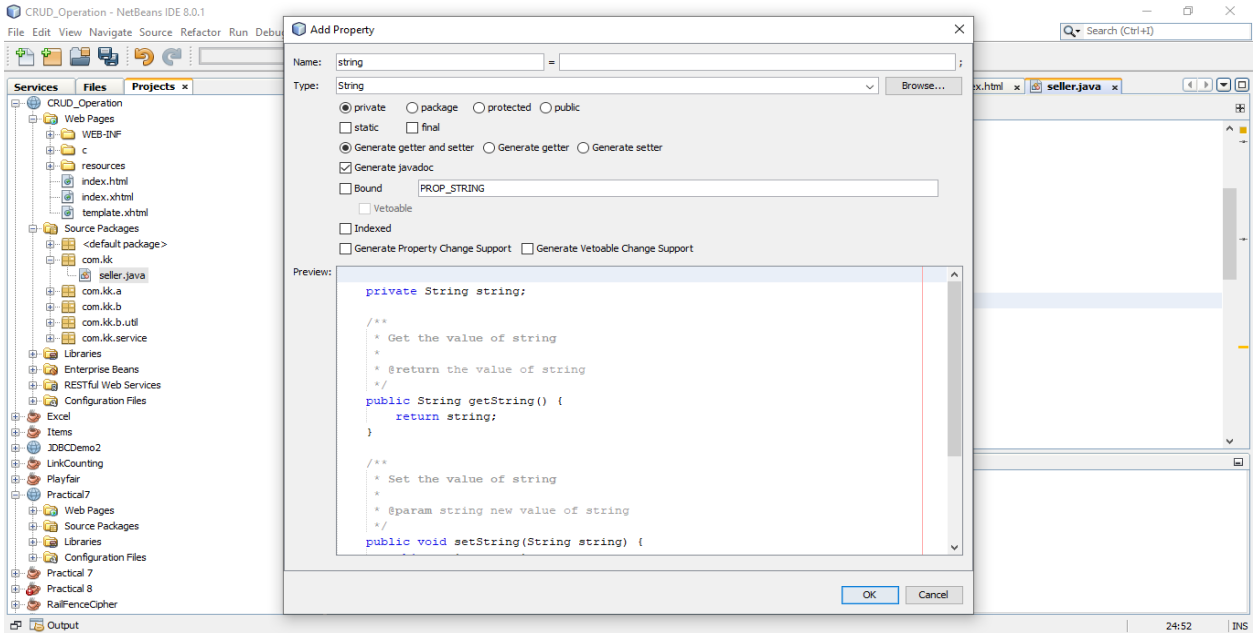


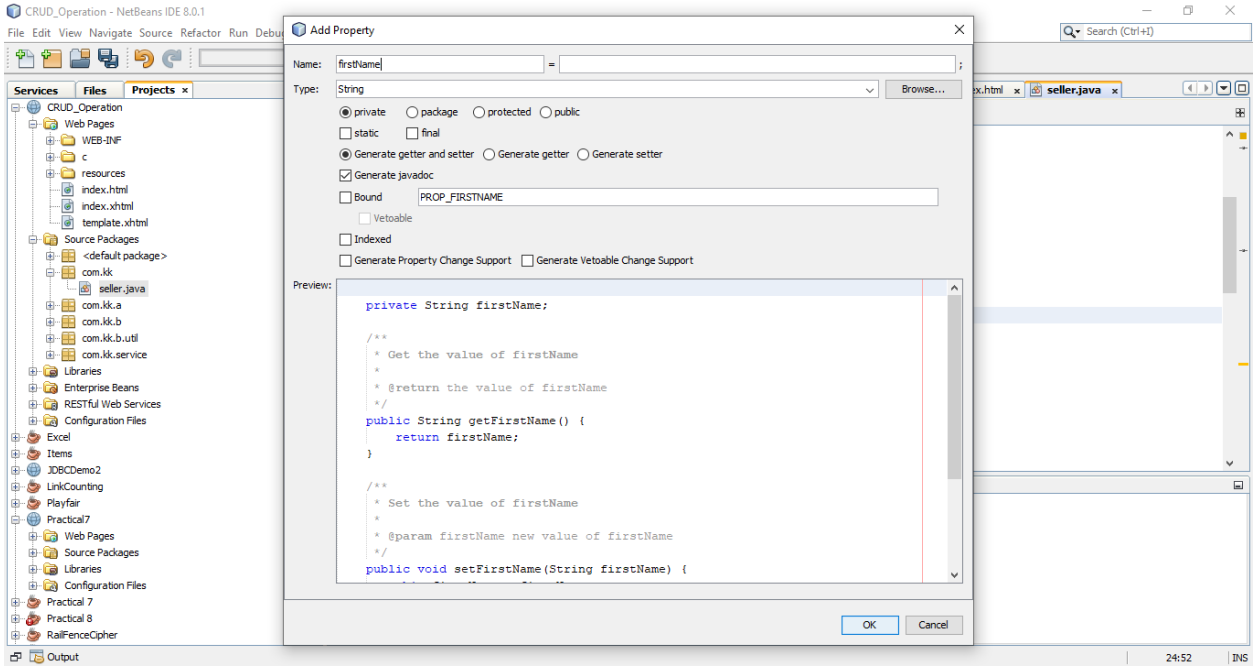
STEP9:Right click on code and select insert code->Add property



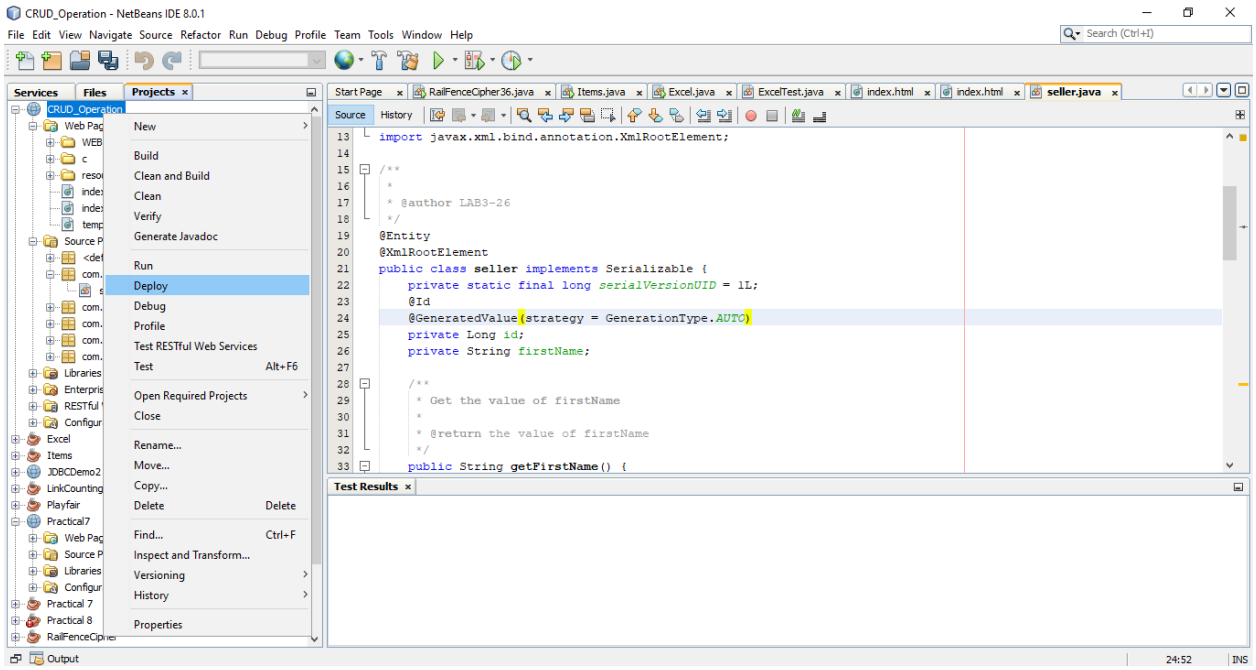


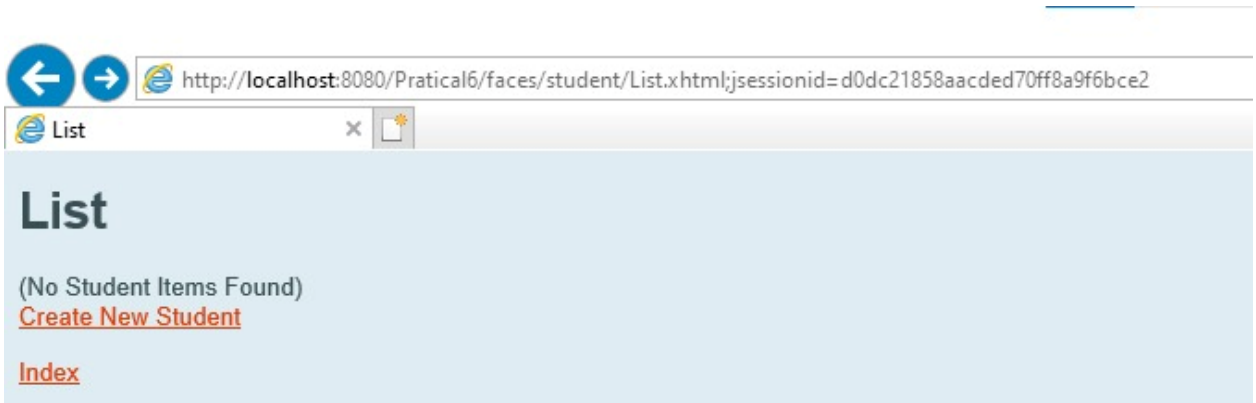
Step10:Change name to FirstName and type as string.

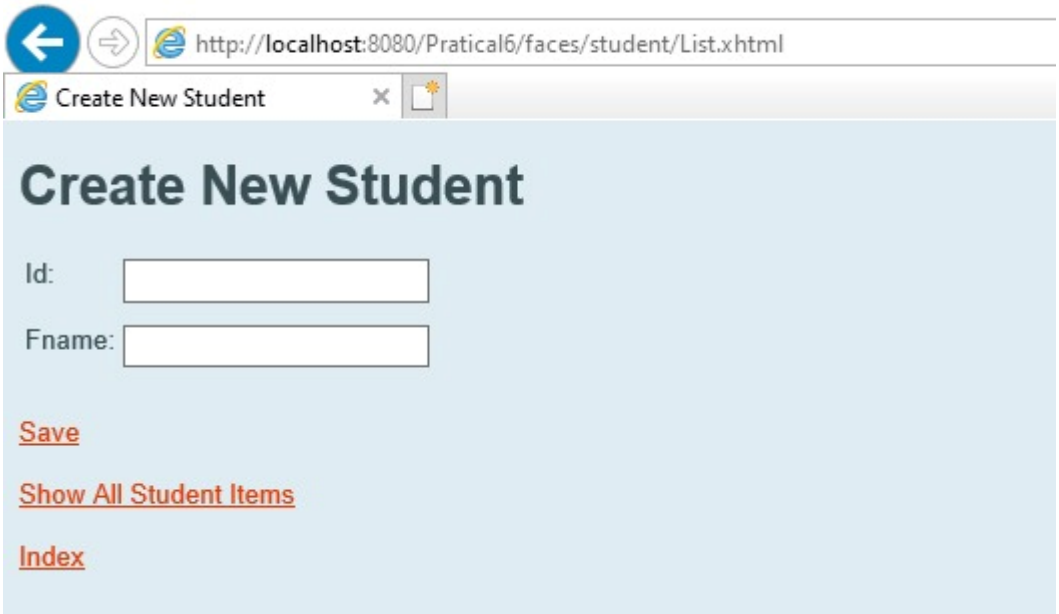


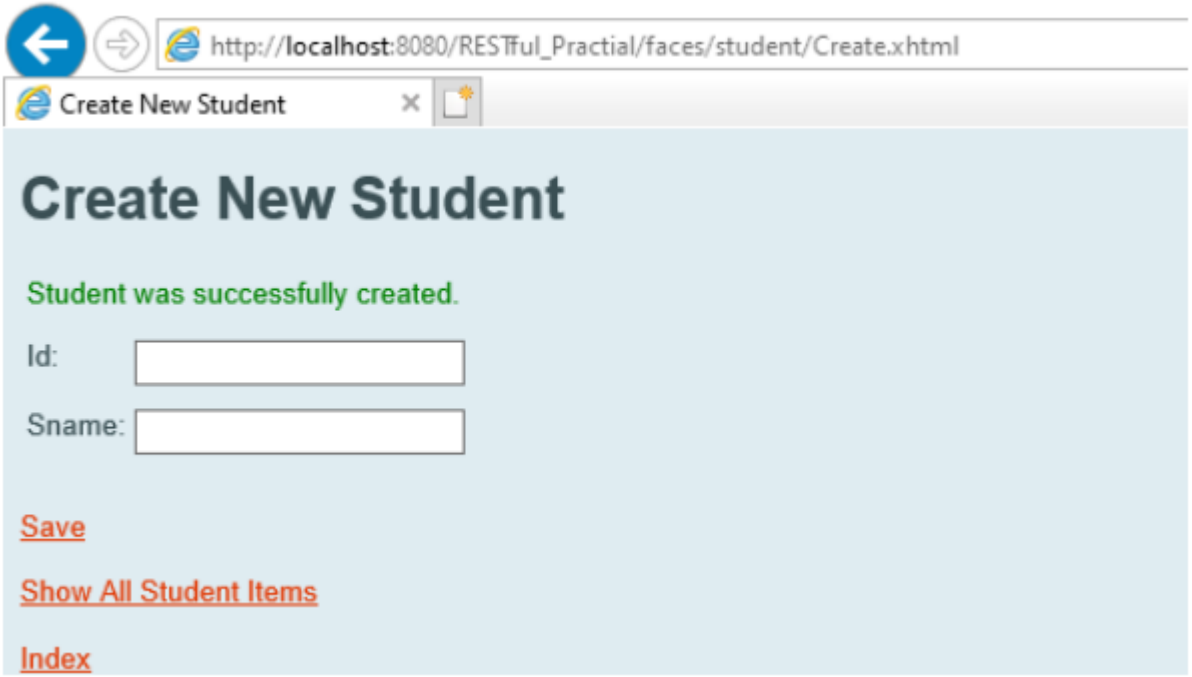


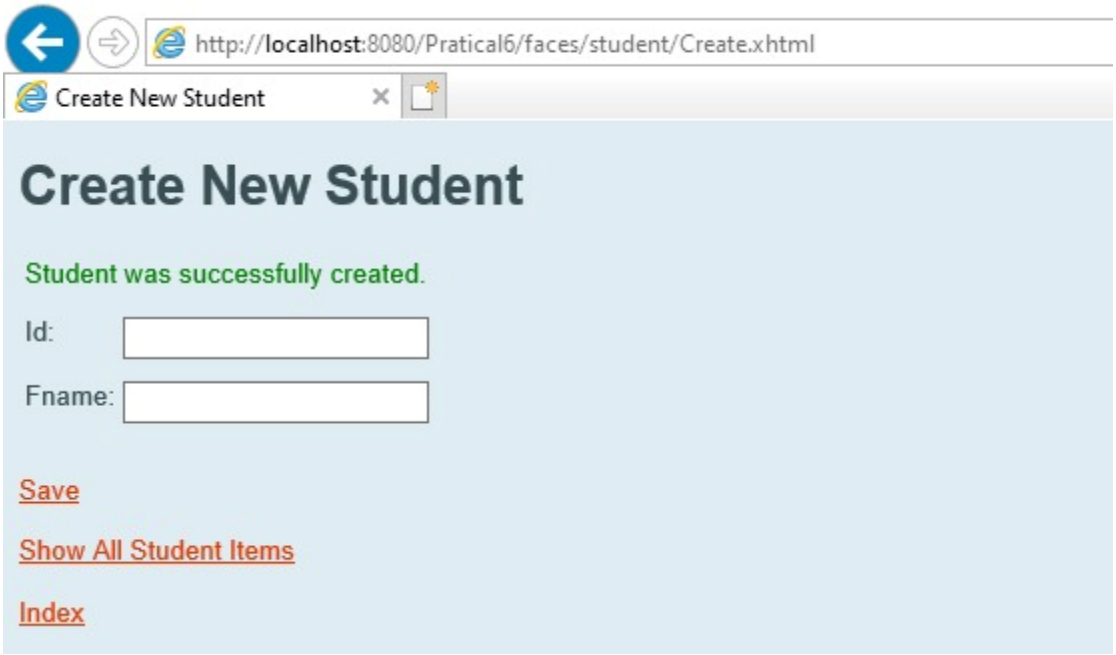
STEP11:Click on Project and select deploy.

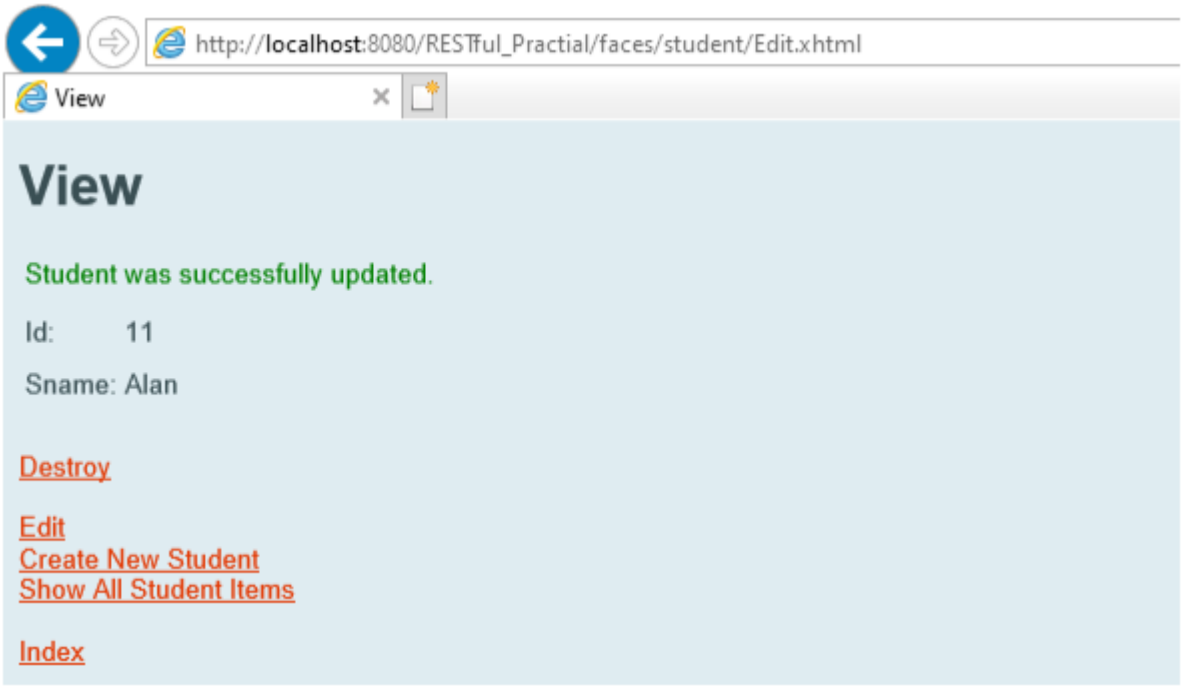





http://localhost:8080/Pratical6/faces/student/List.xhtml
Create New Student
Create New Student
Id:
Fname:
Save
Show All Student Items
Index


http://localhost:8080/RESTful_Practial/faces/student/Create.xhtml
Create New Student
Create New Student
Student was successfully created.
Id:
Sname:
Save
Show All Student Items
Index

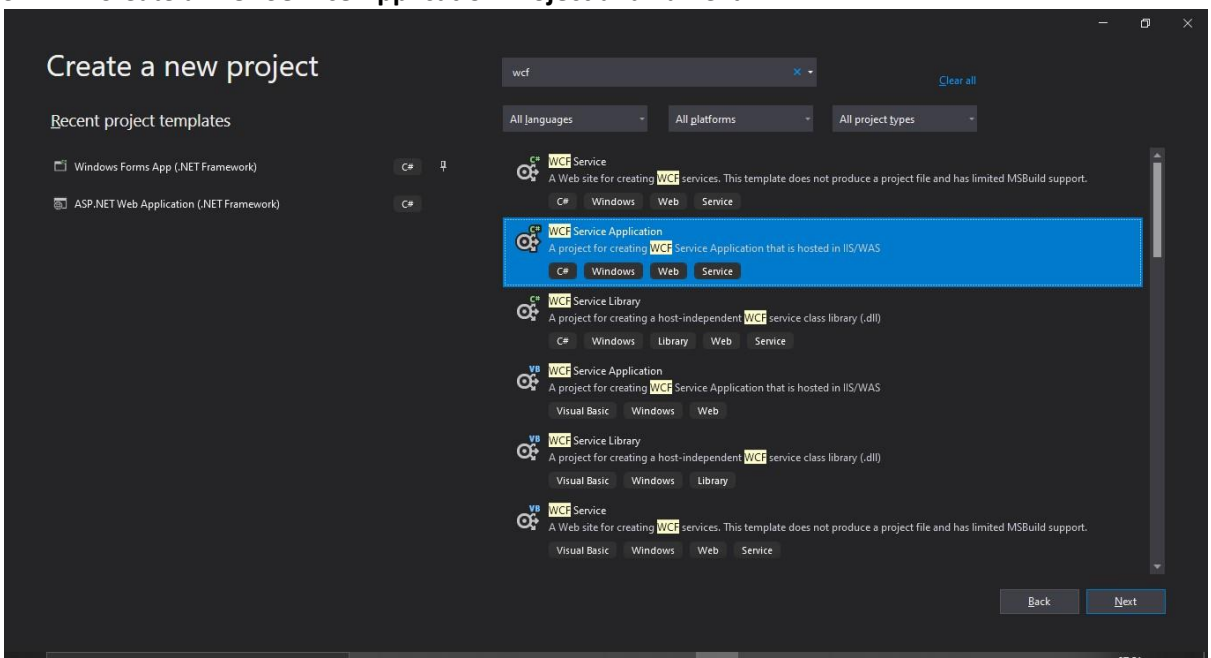

http://localhost:8080/Pratical6/faces/student/Create.xhtml
Create New Student
Create New Student
Student was successfully created.
Id:
Fname:
Save
Show All Student Items
Index

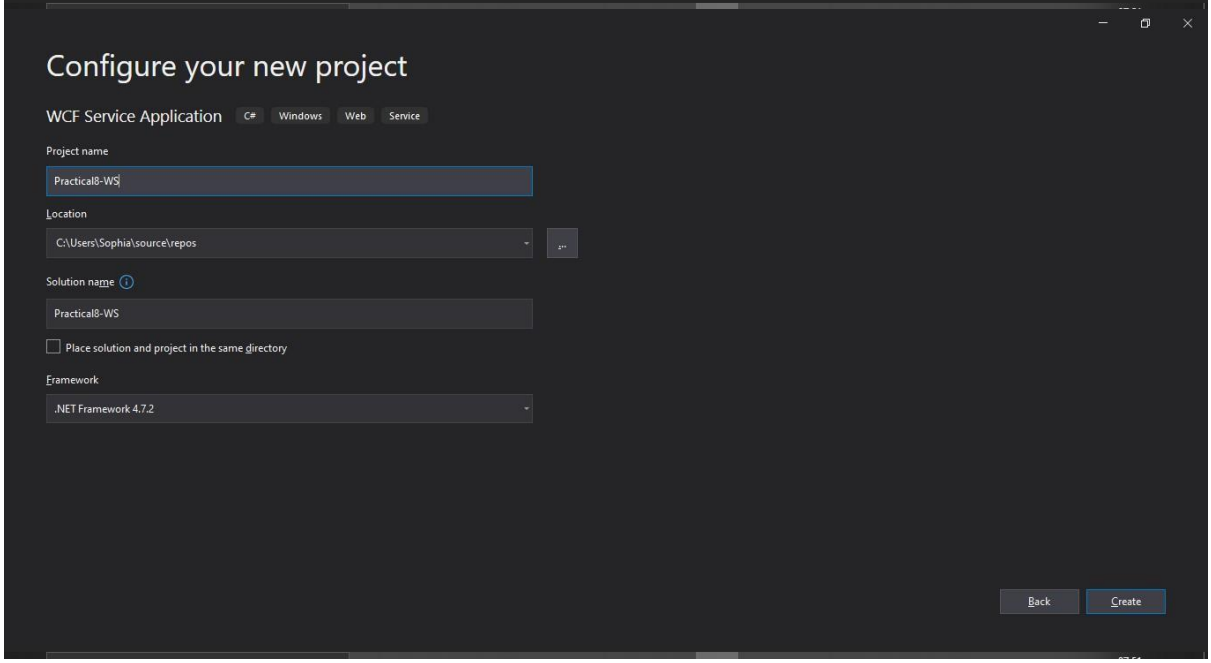

http://localhost:8080/RESTful_Practial/faces/student/Edit.xhtml
View
View
Student was successfully updated.
Id: 11
Sname: Alan
Destroy
Edit
Create New Student
Show All Student Items
Index

Practical 7

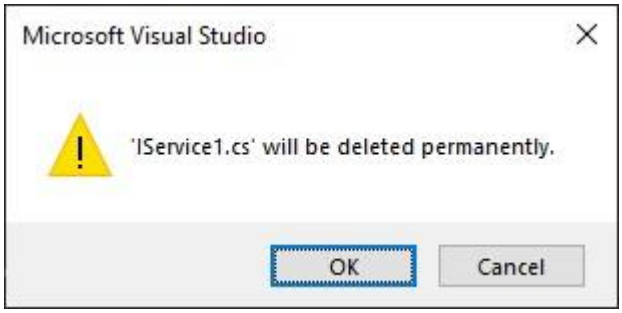
Implement a typical service and a typical client using WCF.

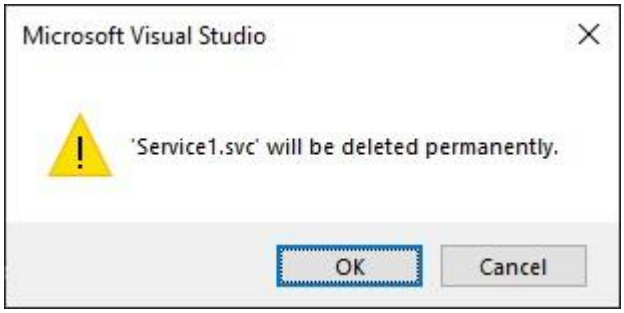
STEP 1: Create a **WCF Service Application Project** and name it.



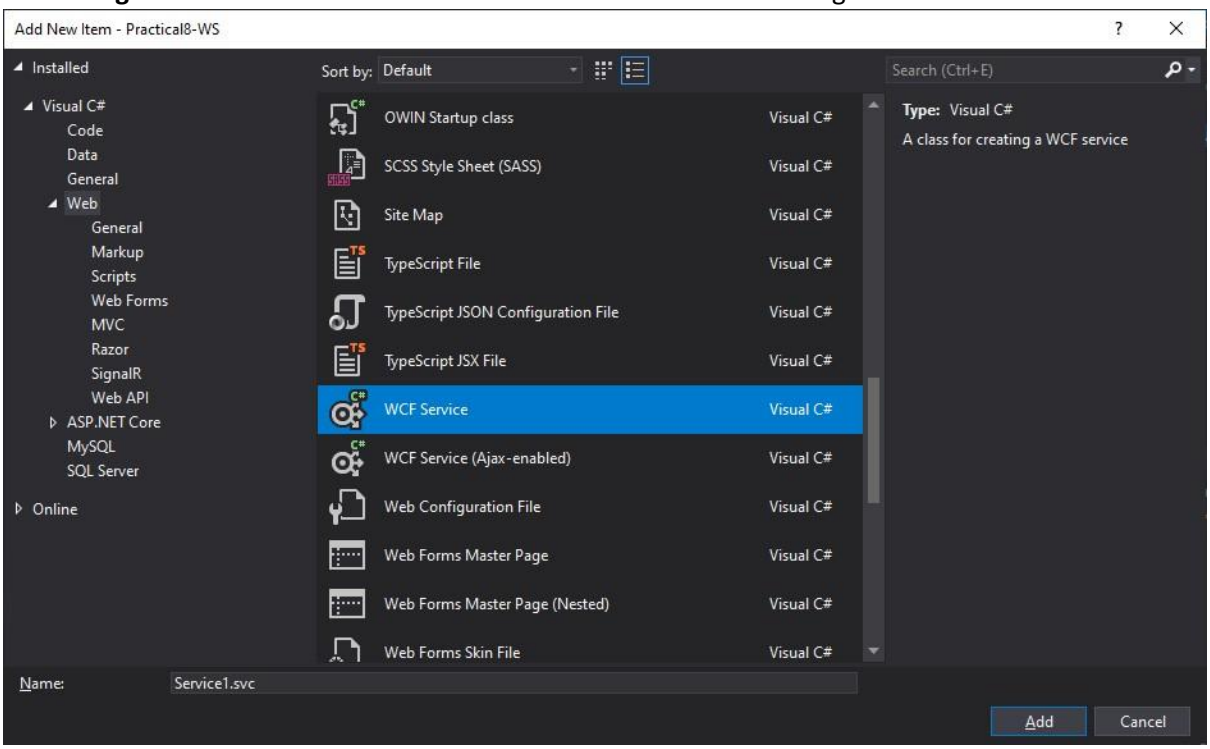


STEP 2: Delete **IService1.cs** and **Service1.cs files.**





STEP 3: Right Click – Add – New Item – WCF Service. Add the following code



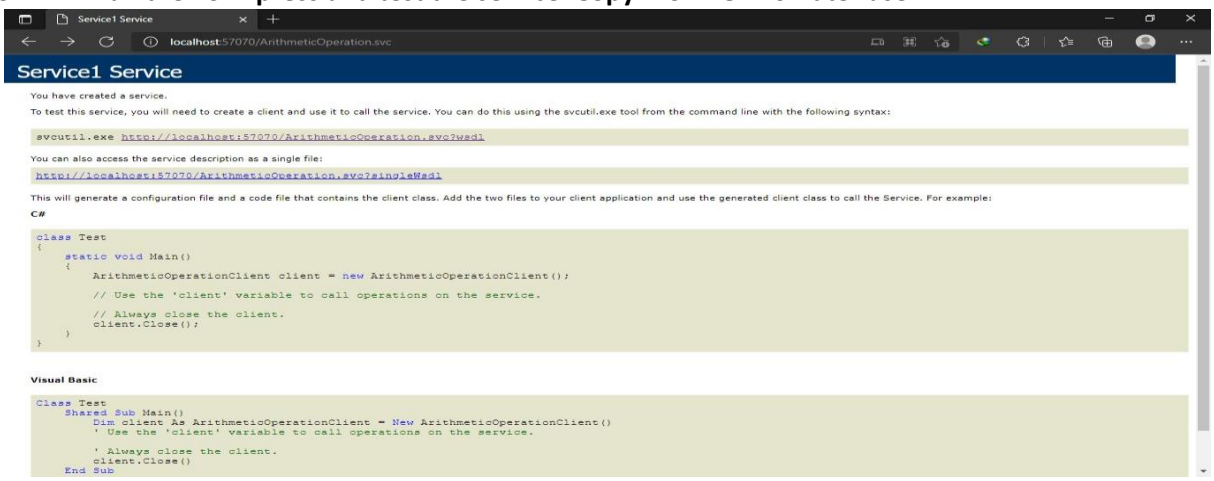
ArithmeticOperation.cs using
System;
using System.Collections.Generic;
using System.Linq; using
System.Runtime.Serialization; using
System.ServiceModel; using
System.Text;

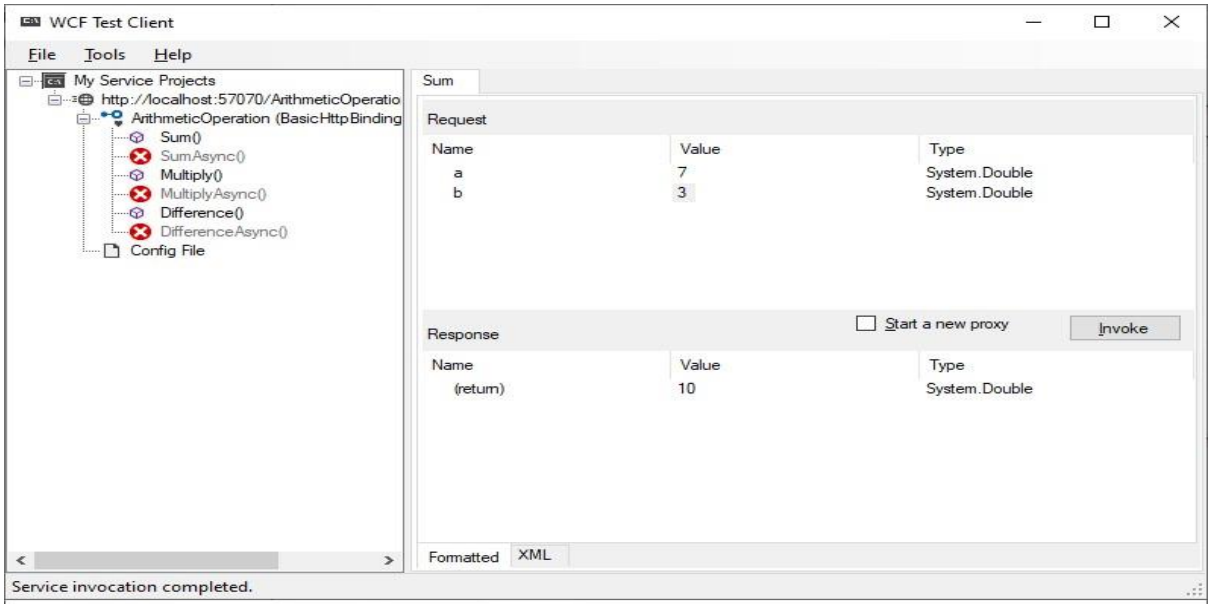
```
namespace Practical8_WS
{
   // NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface
name "IService1" in both code and config file together.
   [ServiceContract]
   public interface ArithmeticOperation
   {
      [OperationContract]
      double Sum(double a, double b);

      [OperationContract]
      double Multiply(double a, double b);

      [OperationContract]
      double Difference(double a, double b);


   }
}
```

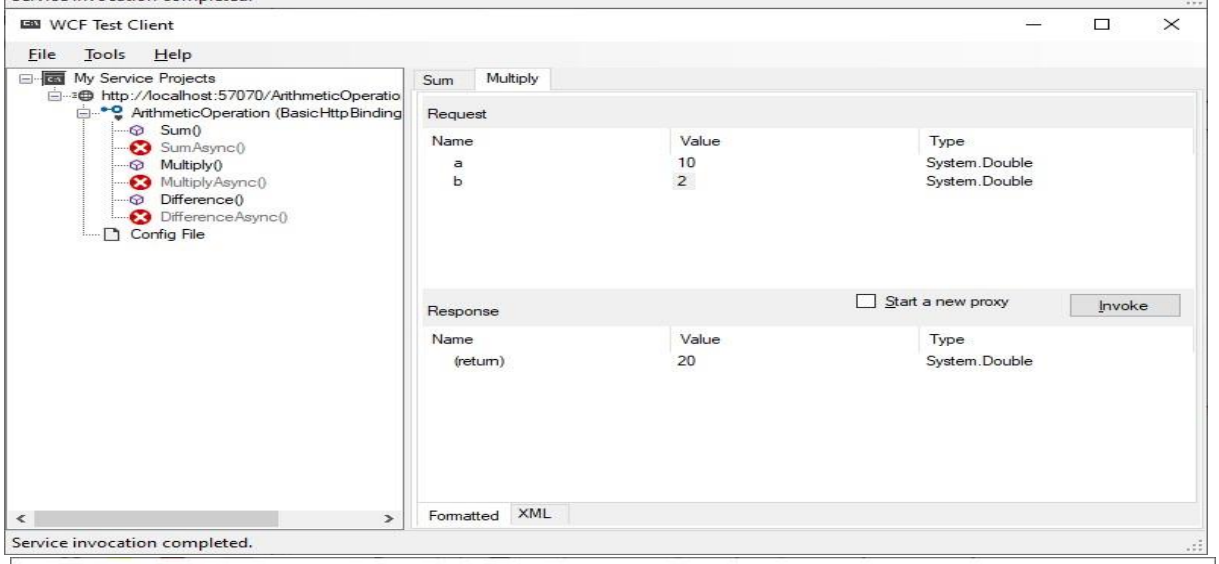
ArithmeticOperation.svc.cs using
System;
using System.Collections.Generic;
using System.Linq; using
System.Runtime.Serialization; using
System.ServiceModel; using
System.Text;

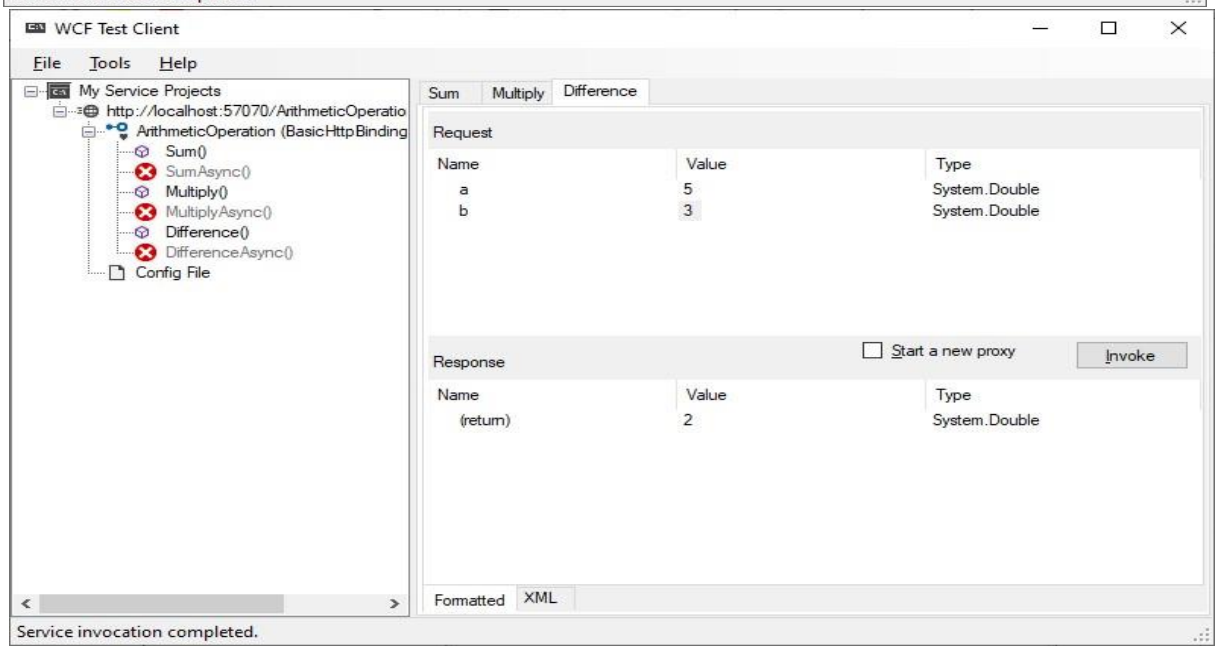
namespace Practical8_WS
{
// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.
// NOTE: In order to launch WCF Test Client for testing this service, please select Service1.svc or Service1.svc.cs at the Solution Explorer and start debugging. public class Service1 : ArithmeticOperation
{
public double Sum(double a, double b)
{
double result = a + b;
return result;
}
public double Multiply(double a, double b)
{
double result = a * b;
return result;
}
public double Difference(double a, double b)
{
double result = a - b;
return result;
}
} }

STEP 4: Run the **IIS Express** and test the service. **Copy WSDL Url** for later use.

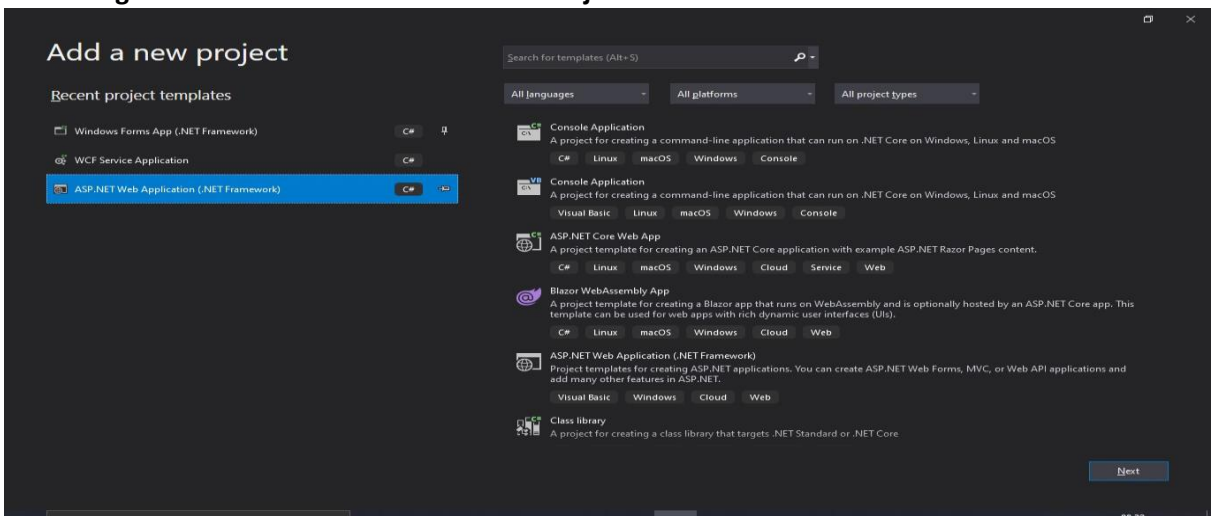



WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:57070/ArithmeticOperatio
ArithmeticOperation (BasicHttpBinding
Sum()
SumAsync()
Multiply()
MultiplyAsync()
Difference()
DifferenceAsync()
Config File
Sum
Request
Name
Value
Type
a
7
System.Double
b
3
System.Double
Start a new proxy
Invoke
Response
Name
Value
Type
(return)
10
System.Double
Formatted
XML
Service invocation completed.

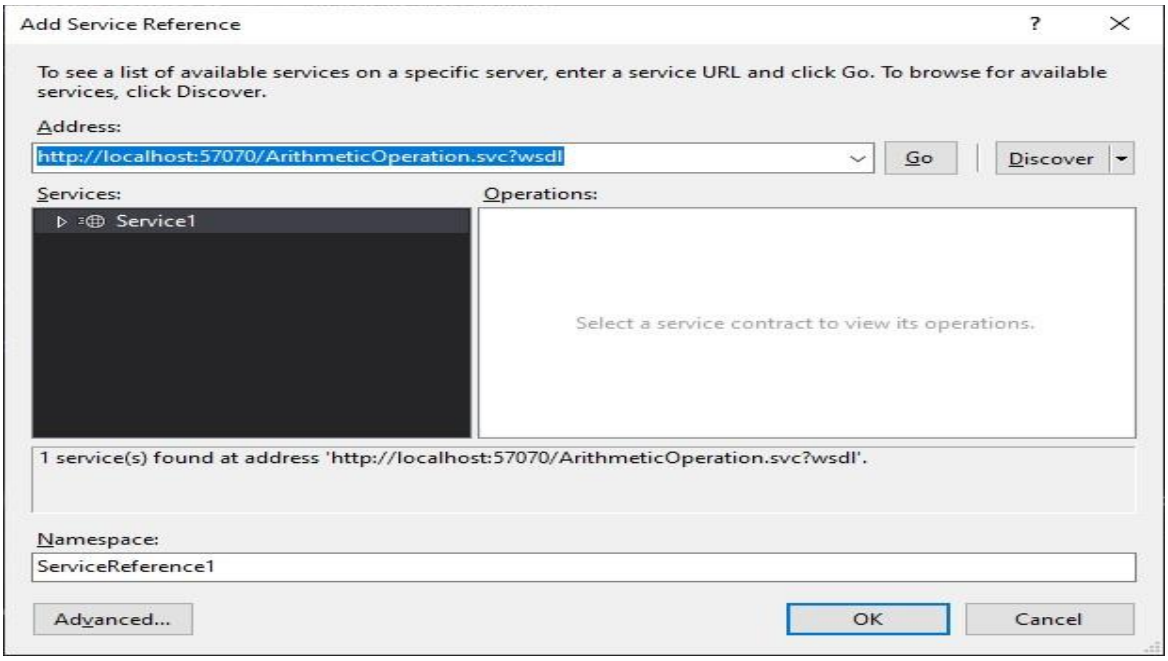

WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:57070/ArithmeticOperatio
ArithmeticOperation (BasicHttpBinding
Sum()
SumAsync()
Multiply()
MultiplyAsync()
Difference()
DifferenceAsync()
Config File
Sum
Multiply
Request
Name
Value
Type
a
10
System.Double
b
2
System.Double
Start a new proxy
Invoke
Response
Name
Value
Type
(return)
20
System.Double
Formatted
XML
Service invocation completed.


WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:57070/ArithmeticOperatio
ArithmeticOperation (BasicHttpBinding
Sum()
SumAsync()
Multiply()
MultiplyAsync()
Difference()
DifferenceAsync()
Config File
Sum
Multiply
Difference
Request
Name
Value
Type
a
5
System.Double
b
3
System.Double
Start a new proxy
Invoke
Response
Name
Value
Type
(return)
2
System.Double
Formatted
XML
Service invocation completed.

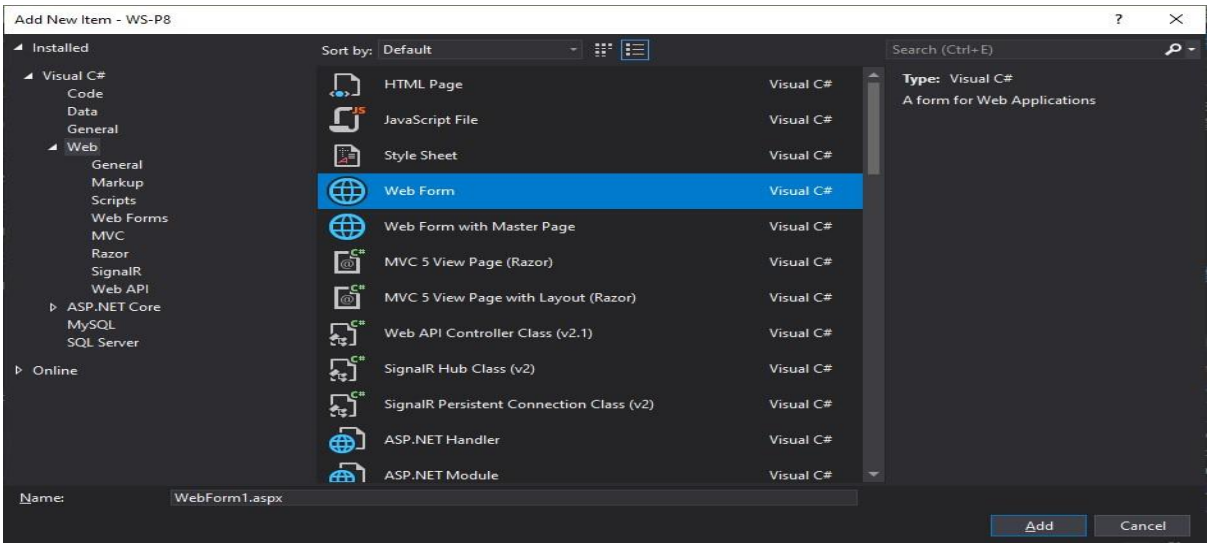
STEP 5: Right Click on Solution and Add New Project

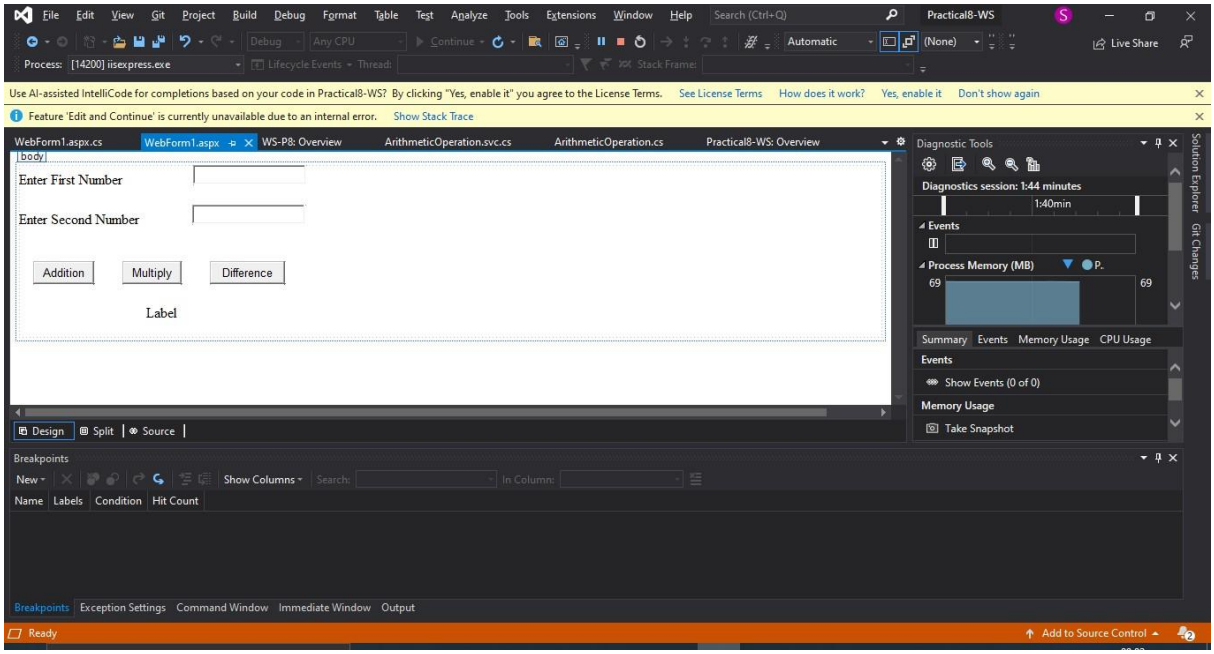


STEP 6: Right Click on Client Project – Reference – Add Service Reference – Add WSDL URL



STEP 7: Right Click – Add – New Item – Web Form. Create a form to accept values and generate result

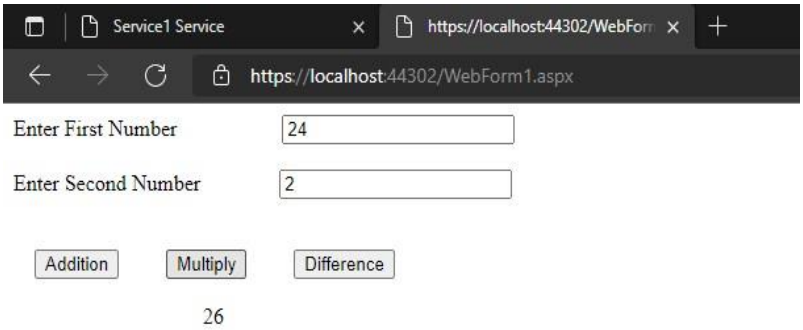


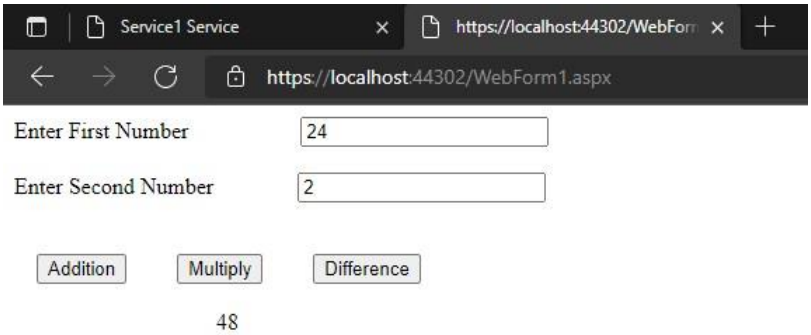


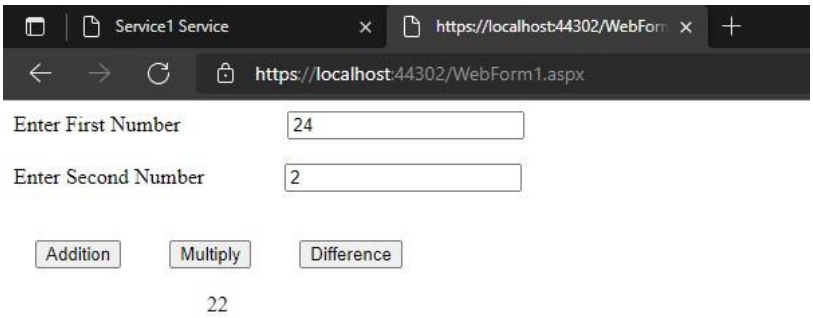

```
using System;
using System.Collections.Generic;
using System.Linq; using
System.Web; using
System.Web.UI;
using System.Web.UI.WebControls;

namespace WS_P8
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         ServiceReference1.ArithmeticOperationClient ob = new
ServiceReference1.ArithmeticOperationClient();
double num1 = double.Parse(TextBox1.Text);
double num2 = double.Parse(TextBox2.Text);
         Label1.Text = Convert.ToString(ob.Sum(num1, num2));
      }

      protected void Button2_Click(object sender, EventArgs e)
      {
         ServiceReference1.ArithmeticOperationClient ob = new
ServiceReference1.ArithmeticOperationClient();
double num1 = double.Parse(TextBox1.Text);
double num2 = double.Parse(TextBox2.Text);
         Label1.Text = Convert.ToString(ob.Multiply(num1, num2));

      }
```

```
        protected void Button3_Click(object sender, EventArgs e)
        {
            ServiceReference1.ArithmeticOperationClient ob = new
ServiceReference1.ArithmeticOperationClient();          double
num1 = double.Parse(TextBox1.Text);          double num2 =
double.Parse(TextBox2.Text);-=\          Label1.Text =
Convert.ToString(ob.Difference(num1, num2));

        }
    }
}
```

OUTPUT:



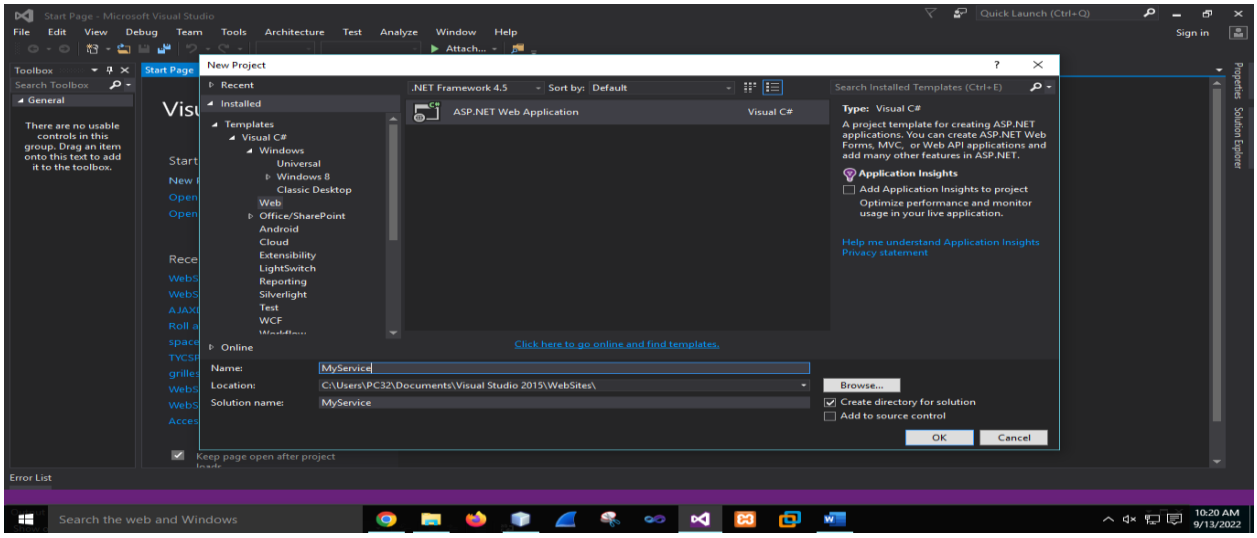


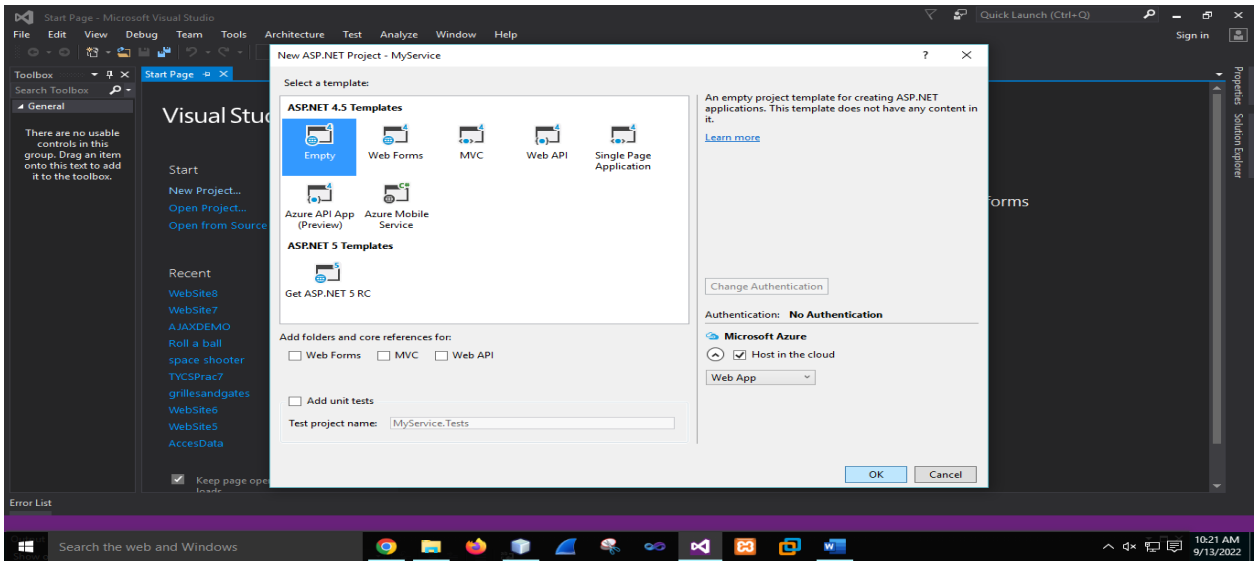


Practical 8

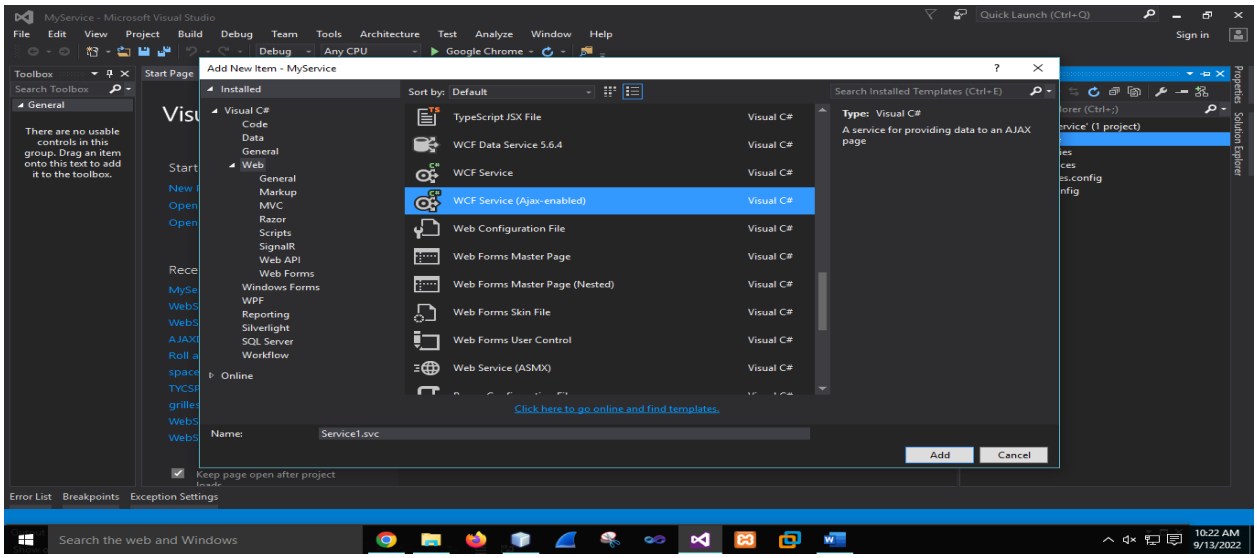
AIM: Use WCF to create a basic ASP.NET Asynchronous JavaScript and XML (AJAX) service

STEP 1: Create a new ASP .Net Web Application project. Name it MyService and Create





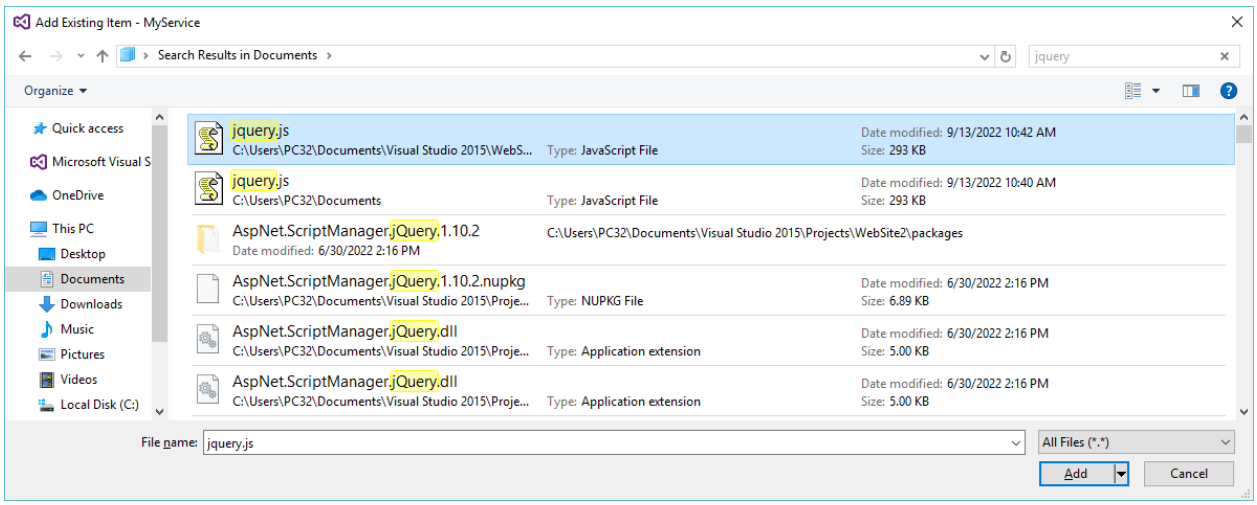
STEP 2: Click on PROJECT – Add New Item – WCF Service (Ajax enabled) – Add. Add the code to it



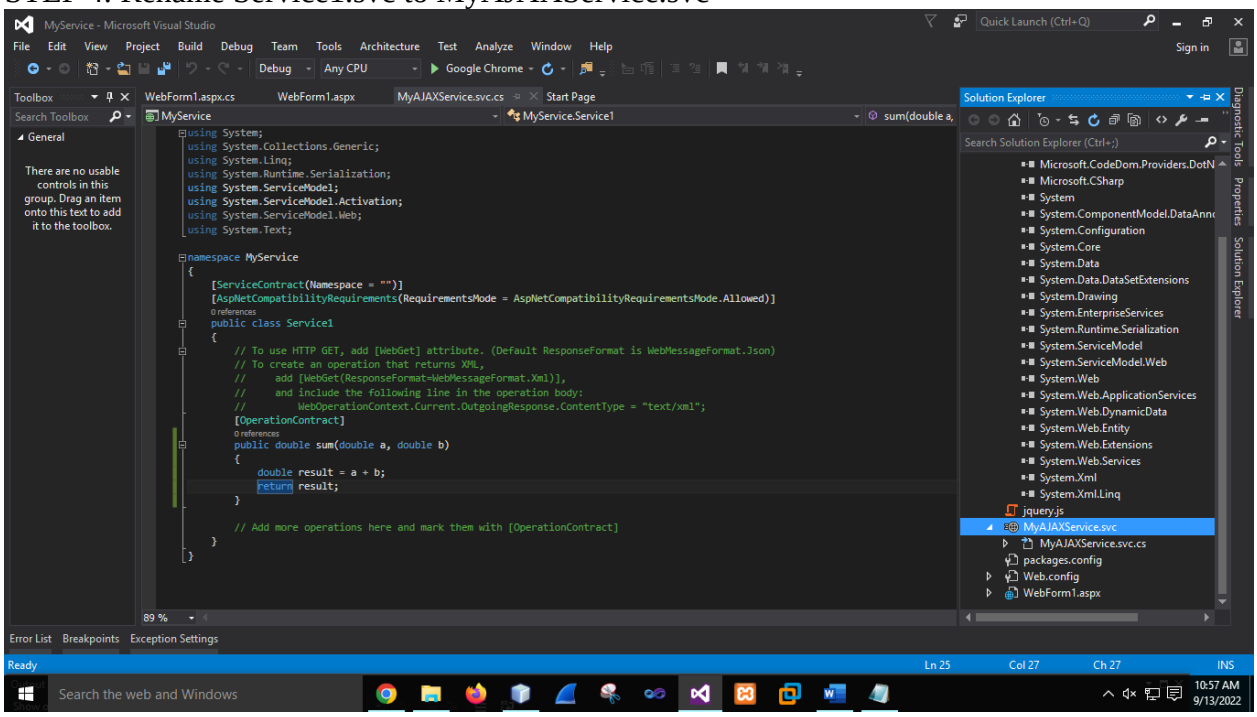
Service1.svc

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Activation;
using System.ServiceModel.Web;
using System.Text;

namespace MyService
{
   [ServiceContract(Namespace = "")]
   [AspNetCompatibilityRequirements(RequirementsMode =
AspNetCompatibilityRequirementsMode.Allowed)]
   public class Service1
   {
      // To use HTTP GET, add [WebGet] attribute. (Default ResponseFormat is
WebMessageFormat.Json)
      // To create an operation that returns XML,
      //     add [WebGet(ResponseFormat=WebMessageFormat.Xml)],
      //     and include the following line in the operation body:
      //         WebOperationContext.Current.OutgoingResponse.ContentType = "text/xml";
      [OperationContract]
      public double sum(double a, double b)
      {
         double result = a + b;
         return result;
      }

      // Add more operations here and mark them with [OperationContract]
   }
}
```

STEP 3: Right Click on project name – Add – Existing Item and attach the jquery file.



STEP 4: Rename Service1.svc to MyAJAXService.svc

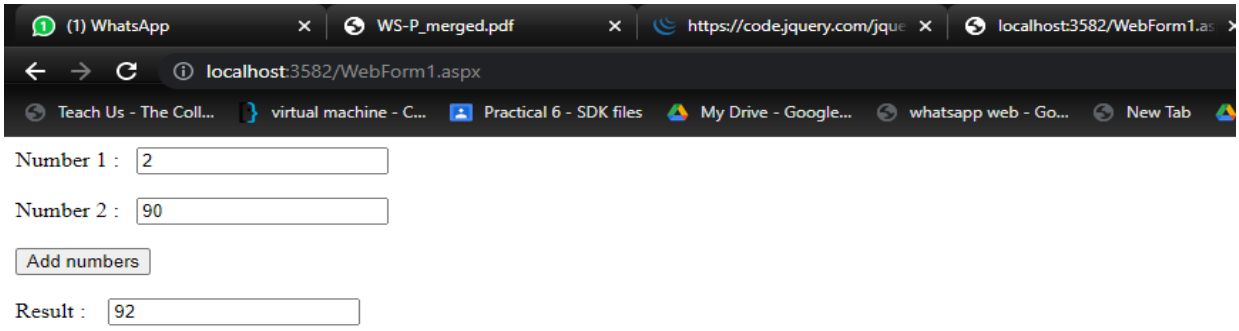


WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="MyService.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <script type="text/javascript" src="jquery.js"></script>
   <script type="text/javascript" >
   $(document).ready(function()
 {
 $("#Button1").click(function ()
 {
 var num1 = $("#TextBox1").val(); var num2 = $("#TextBox2").val();
 $.ajax
 ({
 url: "MyAjaxService.svc/Sum", type: "POST",
 contentType: "application/json; charset=utf-8", data: JSON.stringify({ a: num1, b: num2 }),
 dataType: "json", success: function (data)
 {
 $("#TextBox3").val(data.d);
 }
 ,error: function (err)
 {
 alert(err);
 }
 });
 });
 });
 </script>
</head>
```

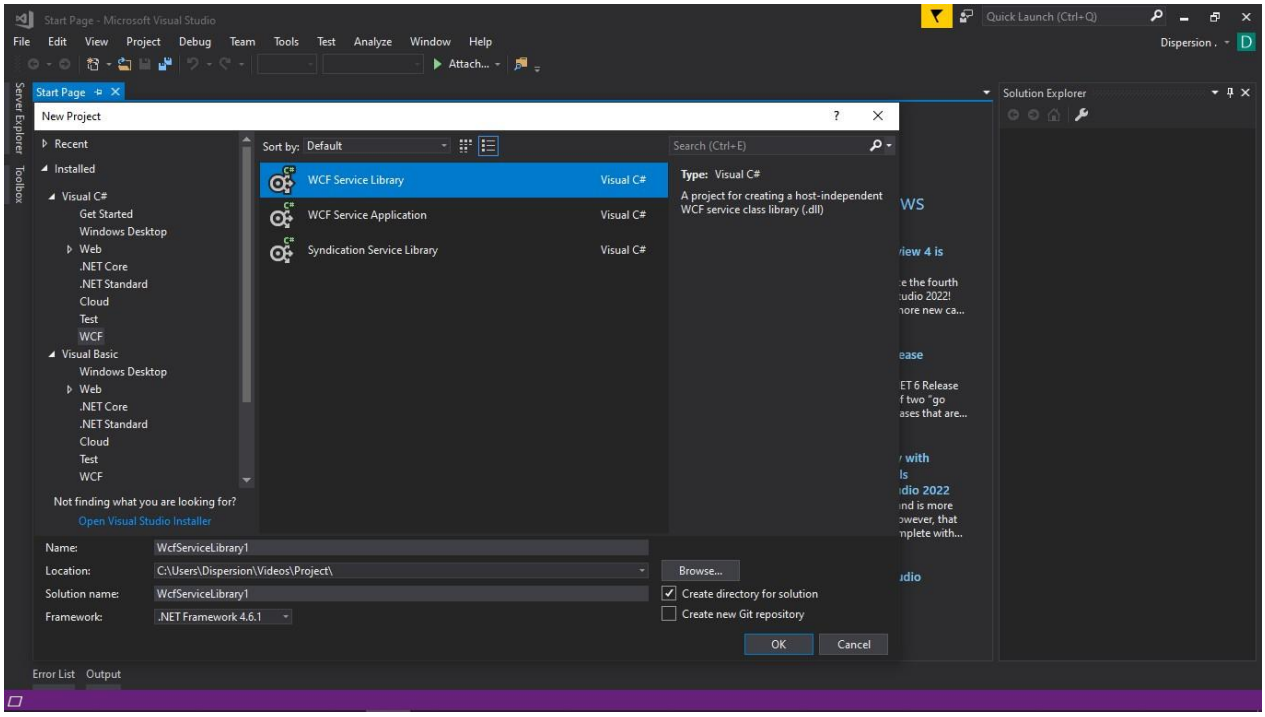
```
<body>
   <form id="form1" runat="server">
   <div>
      <asp:Label ID="Label1" runat="server" Text="Number 1"></asp:Label>
      :  
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <br />
      <br />
      <asp:Label ID="Label2" runat="server" Text="Number 2"></asp:Label>
      :  
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
      <br />
      <br />
      <asp:Button ID="Button1" runat="server" Text="Add numbers"
OnClick="Button1_Click" />
      <br />
      <br />
      <asp:Label ID="Label3" runat="server" Text="Result"></asp:Label>
      :   
      <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>

   </div>
   </form>
</body>
</html>
```

OUTPUT

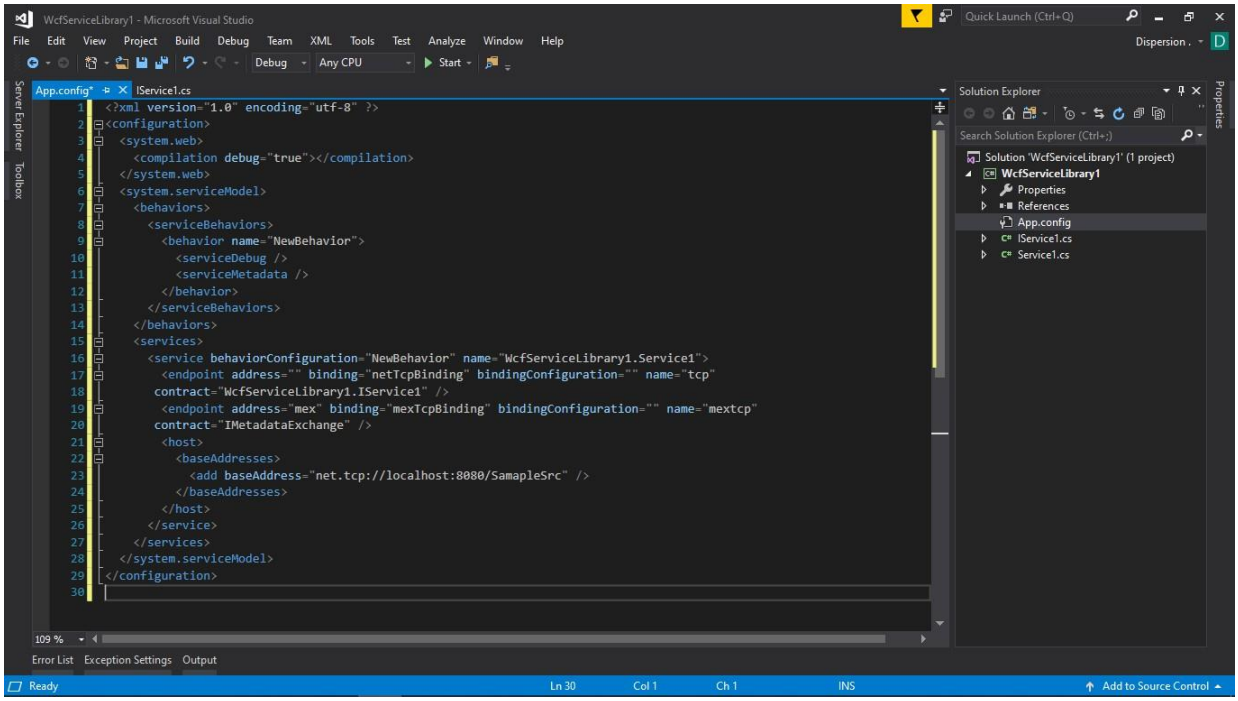


Practical 9
Demonstrates using the binding attribute of an endpoint element in WCF.

Step1: Create a WCF Service Library as shown below



Step2: Open App.config and replace the original code with the one given below. Build and Run the application

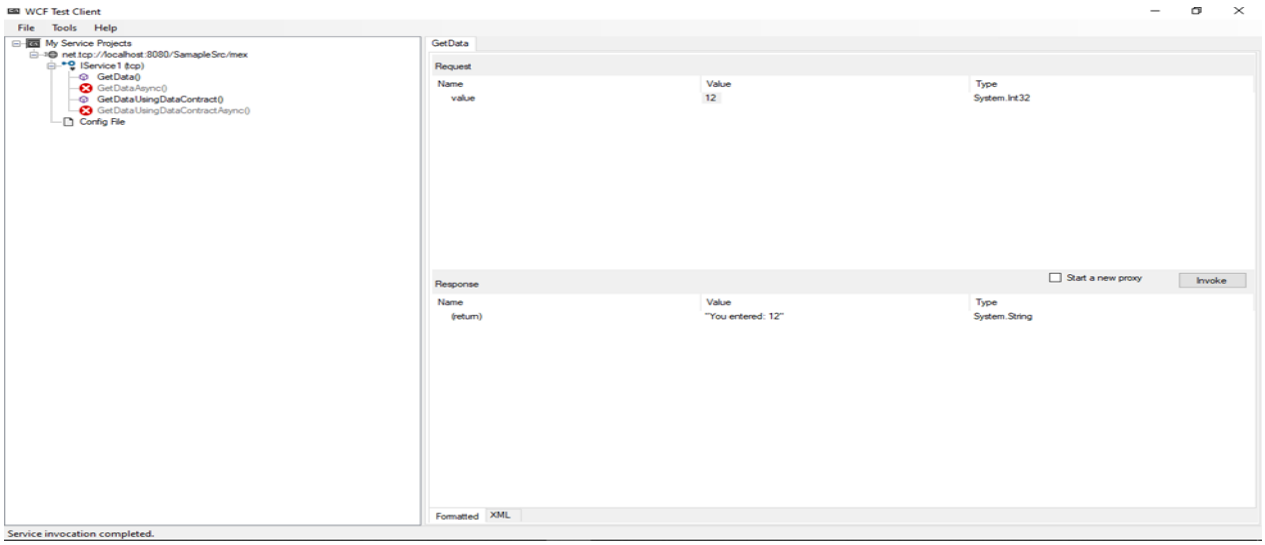


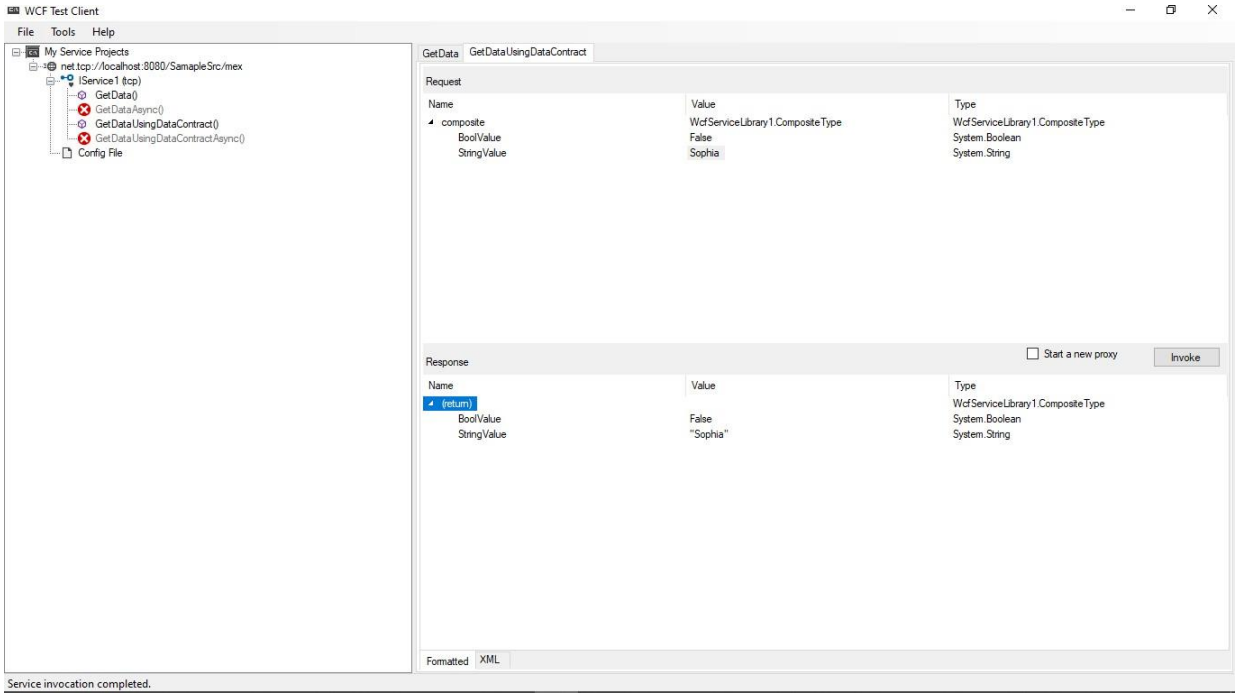

```xml
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <compilation debug="true"></compilation>
  </system.web>
  <system.serviceModel>
    <behaviors>
      <serviceBehaviors>
        <behavior name="NewBehavior">
          <serviceDebug />
          <serviceMetadata />
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <services>
      <service behaviorConfiguration="NewBehavior" name="WcfServiceLibrary1.Service1">
        <endpoint address="" binding="netTcpBinding" bindingConfiguration="" name="tcp"
       contract="WcfServiceLibrary1.IService1" />
        <endpoint address="mex" binding="mexTcpBinding" bindingConfiguration="" name="mextcp"
       contract="IMetadataExchange" />
        <host>
          <baseAddresses>
            <add baseAddress="net.tcp://localhost:8080/SamapleSrc" />
          </baseAddresses>
        </host>
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<system.web>
<compilation debug="true"></compilation>
</system.web>
<system.serviceModel>
<behaviors>
<serviceBehaviors>
<behavior name="NewBehavior">
<serviceDebug />
<serviceMetadata />
</behavior>
</serviceBehaviors>
</behaviors>
<services>
<service behaviorConfiguration="NewBehavior" name="WcfServiceLibrary1.Service1">
<endpoint address="" binding="netTcpBinding" bindingConfiguration="" name="tcp"
contract="WcfServiceLibrary1.IService1" />
<endpoint address="mex" binding="mexTcpBinding" bindingConfiguration=""name="mextcp"
contract="IMetadataExchange" />
<host>
<baseAddresses>
<add baseAddress="net.tcp://localhost:8080/SamapleSrc" />
</baseAddresses>
</host>
</service>
</services>
</system.serviceModel>
</configuration>
```

OUTPUT




WCF Test Client
File
Tools
Help
My Service Projects
net.tcp://localhost:8080/SamapleSrc/mex
IService1 (tcp)
GetData()
GetDataAsync()
GetDataUsingDataContract()
GetDataUsingDataContractAsync()
Config File
GetData
GetDataUsingDataContract
Request
Name
Value
Type
composite
WcfServiceLibrary1.CompositeType
WcfServiceLibrary1.CompositeType
BoolValue
False
System.Boolean
StringValue
Sophia
System.String
Start a new proxy
Invoke
Response
Name
Value
Type
(return)
WcfServiceLibrary1.CompositeType
BoolValue
False
System.Boolean
StringValue
"Sophia"
System.String
Formatted
XML
Service invocation completed.