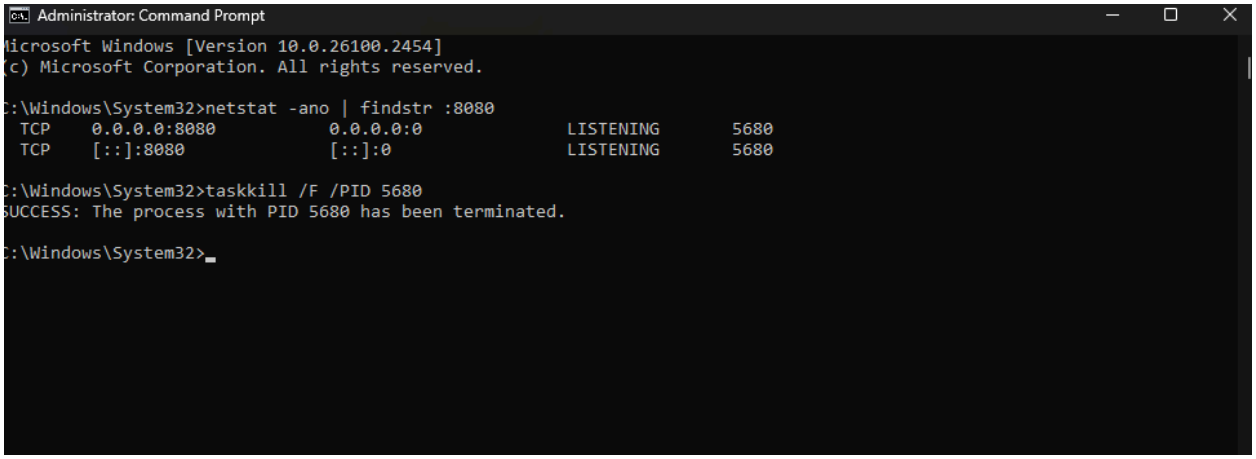
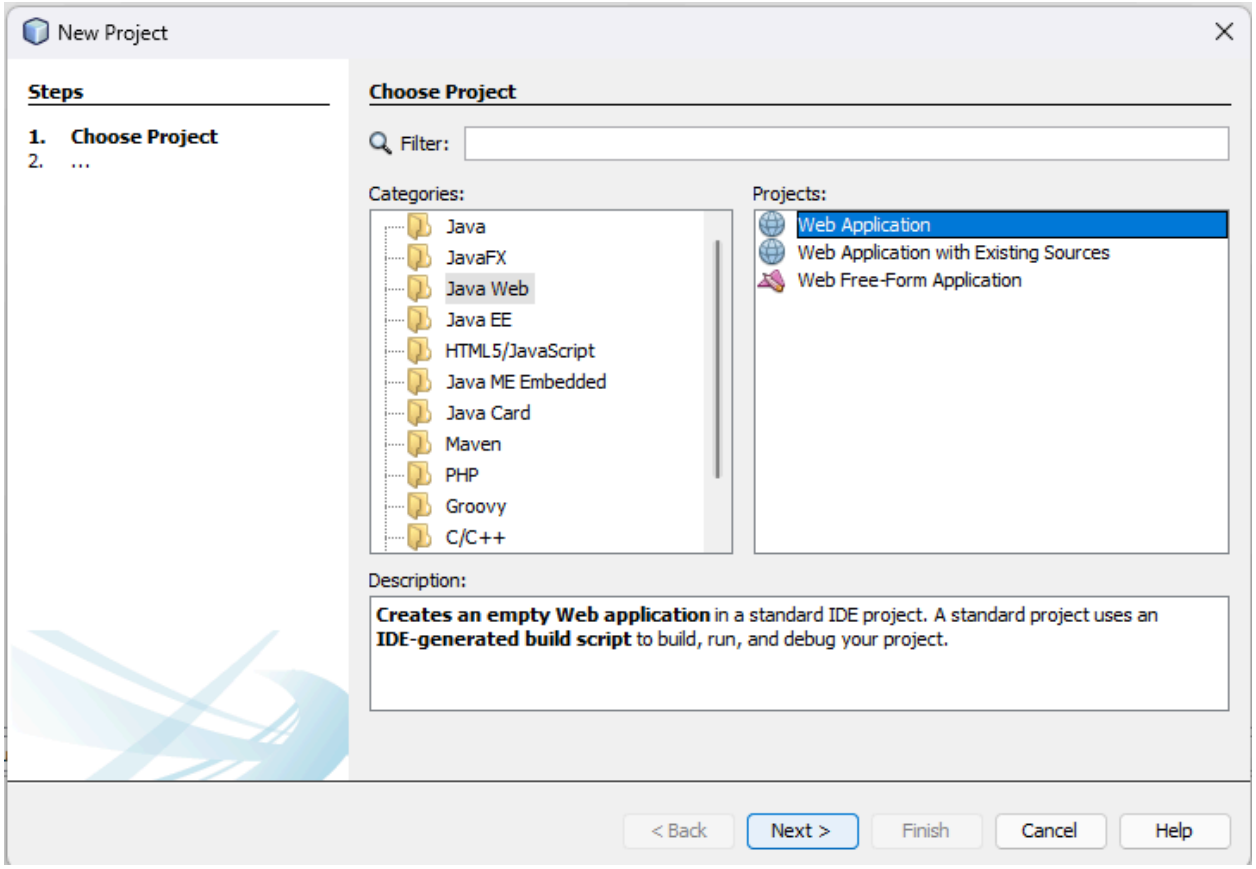
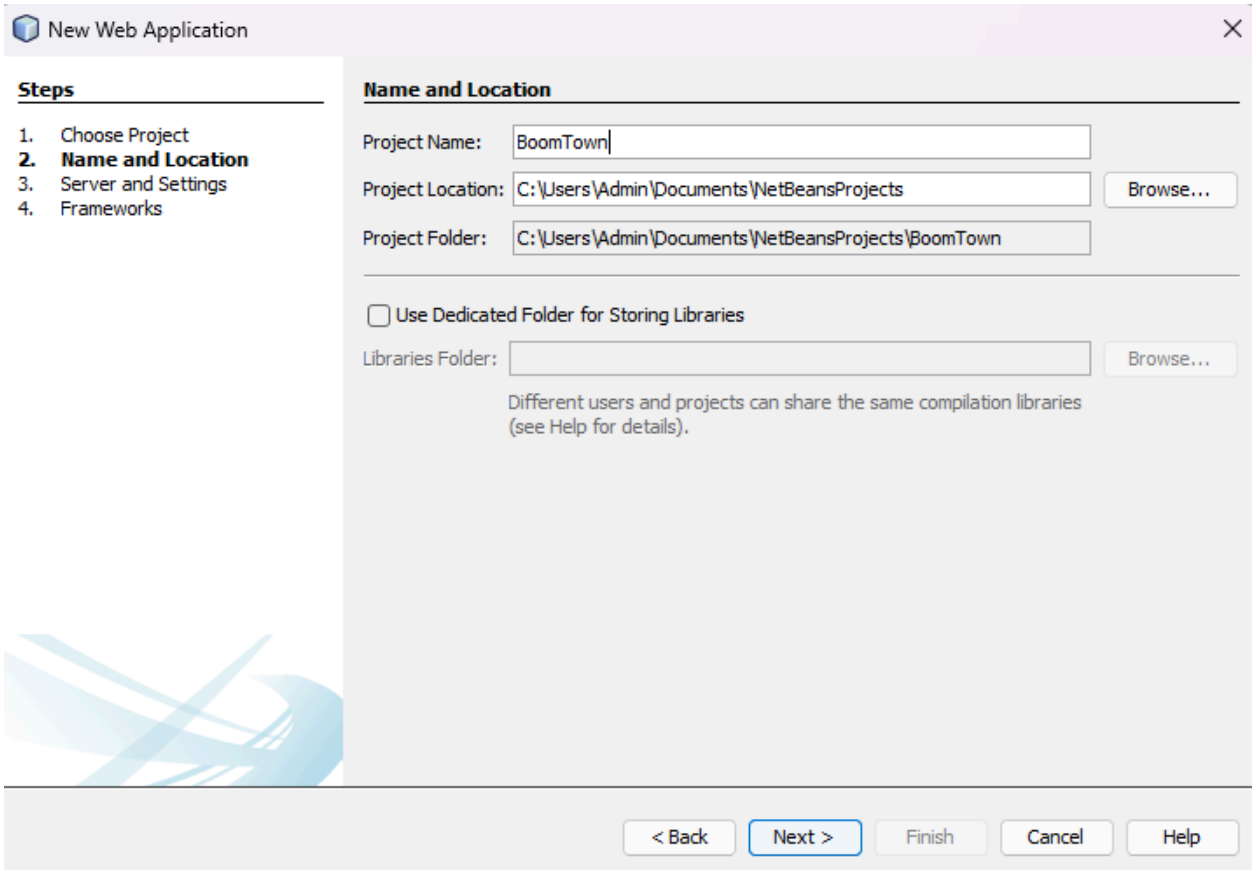
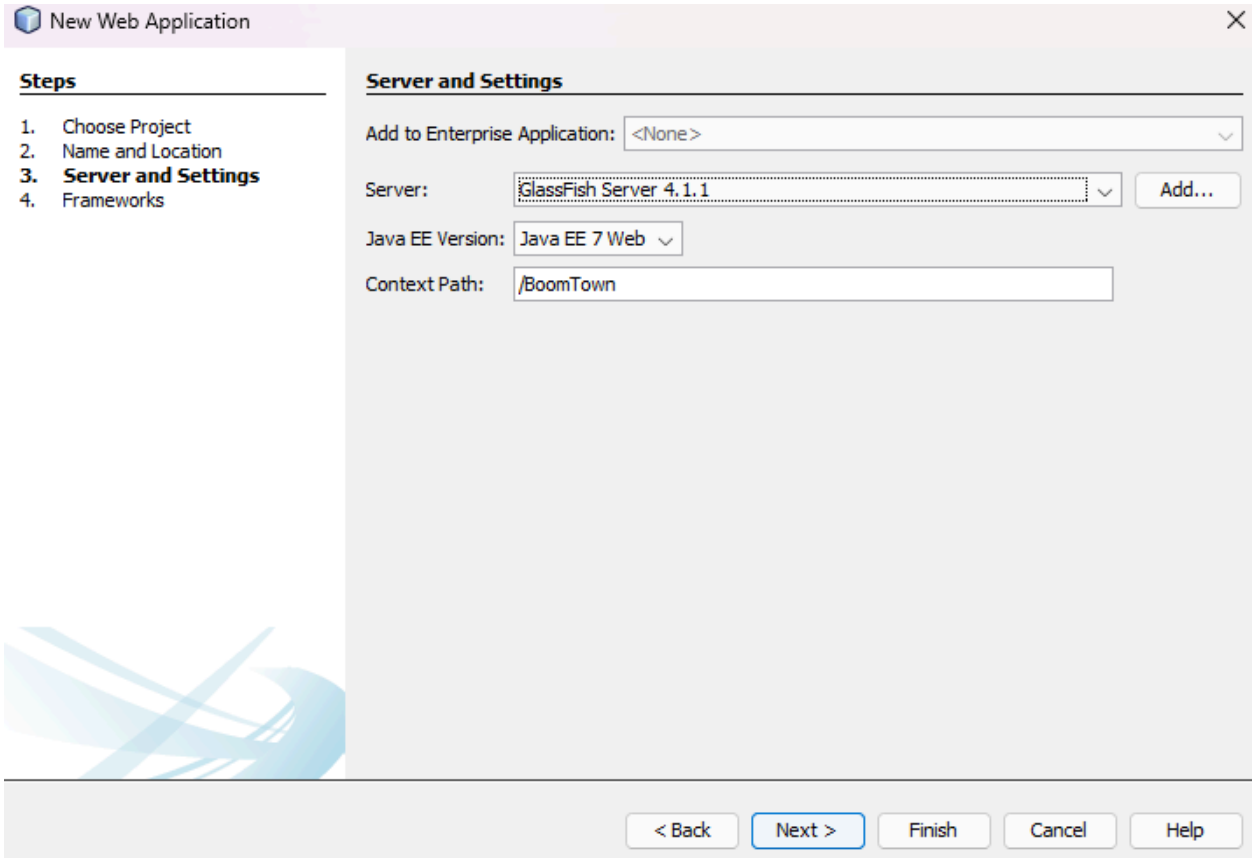
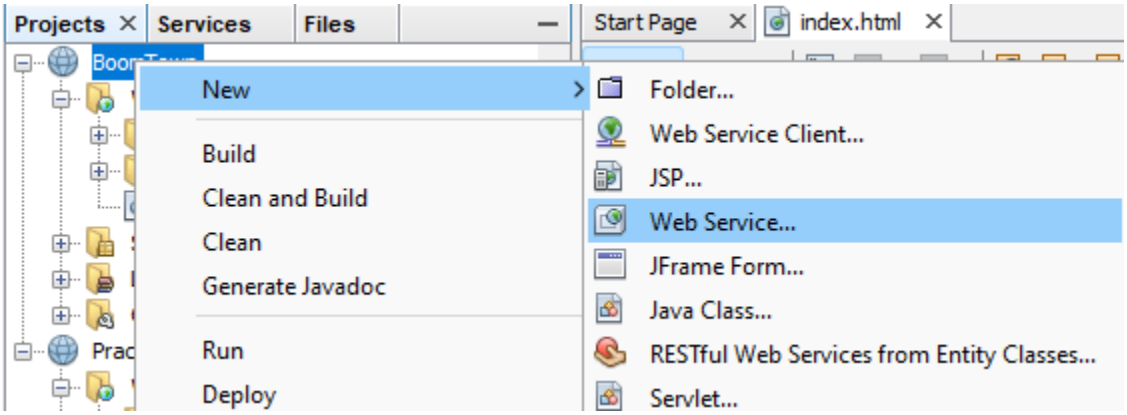
Practical 1

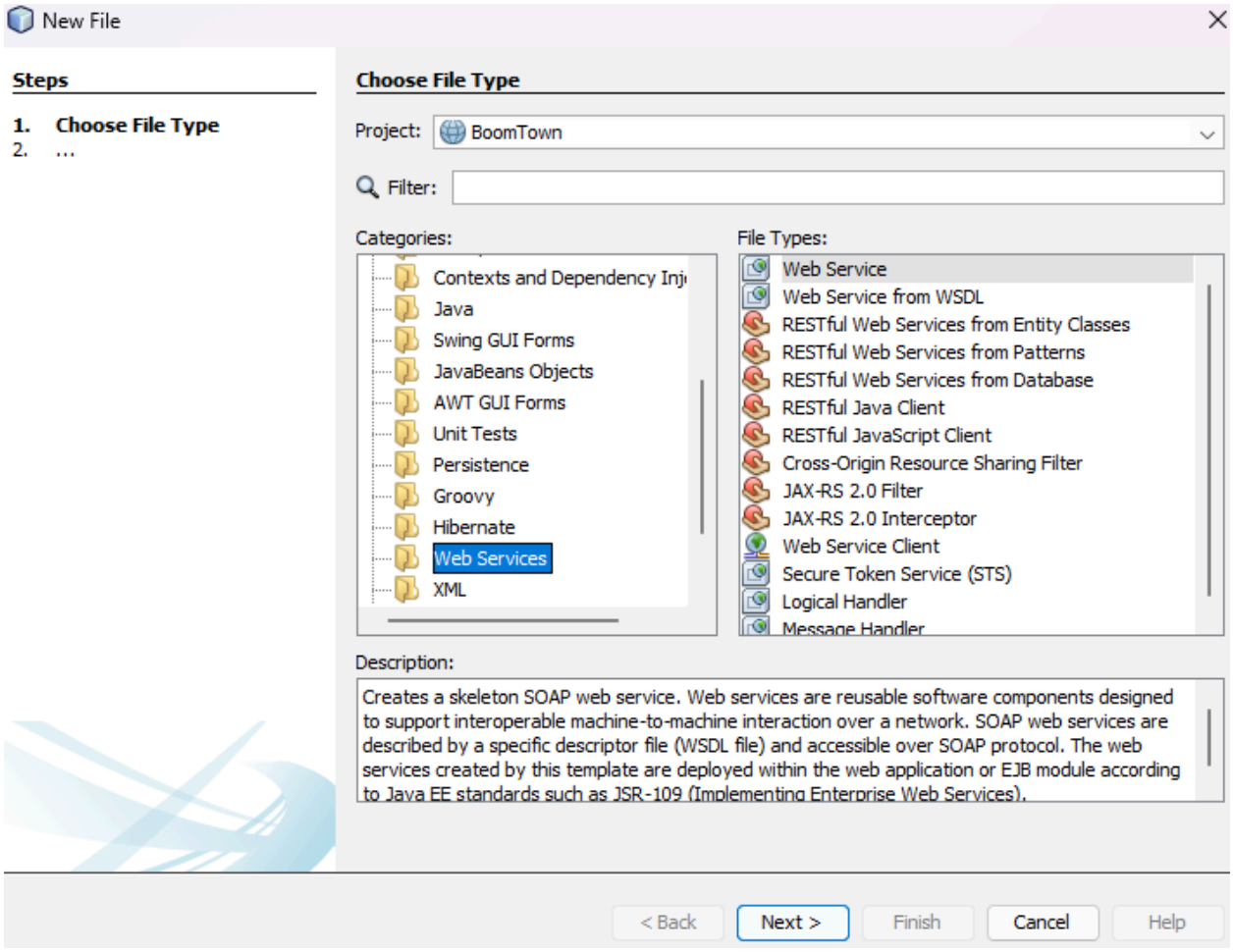

Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\System32>netstat -ano | findstr :8080
TCP 0.0.0.0:8080 0.0.0.0:0 LISTENING 5680
TCP [::]:8080 [::]:0 LISTENING 5680
C:\Windows\System32>taskkill /F /PID 5680
SUCCESS: The process with PID 5680 has been terminated.
C:\Windows\System32>

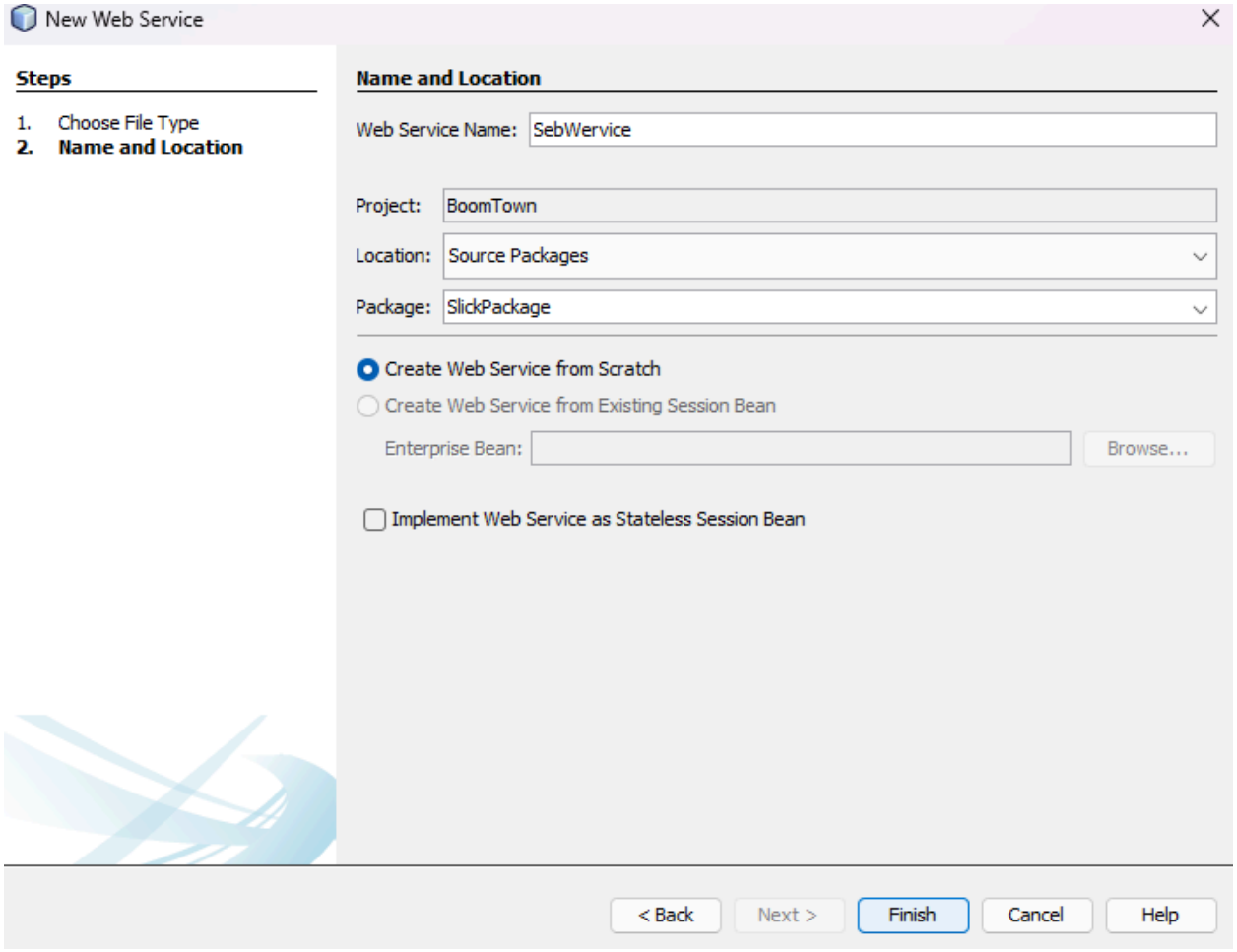

New Project
Steps
1. Choose Project
2. ...
Choose Project
Filter:
Categories:
Java
JavaFX
Java Web
Java EE
HTML5/JavaScript
Java ME Embedded
Java Card
Maven
PHP
Groovy
C/C++
Projects:
Web Application
Web Application with Existing Sources
Web Free-Form Application
Description:
Creates an empty Web application in a standard IDE project. A standard project uses an IDE-generated build script to build, run, and debug your project.
< Back
Next >
Finish
Cancel
Help


New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: BoomTown
Project Location: C:\Users\Admin\Documents\NetBeansProjects
Browse...
Project Folder: C:\Users\Admin\Documents\NetBeansProjects\BoomTown
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 4.1.1
Add...
Java EE Version:
Java EE 7 Web
Context Path:
/BoomTown
< Back
Next >
Finish
Cancel
Help


Projects
Services
Files
Start Page
index.html
New
Build
Clean and Build
Clean
Generate Javadoc
Run
Deploy
Folder...
Web Service Client...
JSP...
Web Service...
JFrame Form...
Java Class...
RESTful Web Services from Entity Classes...
Servlet...


New File
Steps
1. Choose File Type
2. ...
Choose File Type
Project:
BoomTown
Filter:
Categories:
Contexts and Dependency Inj
Java
Swing GUI Forms
JavaBeans Objects
AWT GUI Forms
Unit Tests
Persistence
Groovy
Hibernate
Web Services
XML
File Types:
Web Service
Web Service from WSDL
RESTful Web Services from Entity Classes
RESTful Web Services from Patterns
RESTful Web Services from Database
RESTful Java Client
RESTful JavaScript Client
Cross-Origin Resource Sharing Filter
JAX-RS 2.0 Filter
JAX-RS 2.0 Interceptor
Web Service Client
Secure Token Service (STS)
Logical Handler
Message Handler
Description:
Creates a skeleton SOAP web service. Web services are reusable software components designed to support interoperable machine-to-machine interaction over a network. SOAP web services are described by a specific descriptor file (WSDL file) and accessible over SOAP protocol. The web services created by this template are deployed within the web application or EJB module according to Java EE standards such as JSR-109 (Implementing Enterprise Web Services).
< Back
Next >
Finish
Cancel
Help

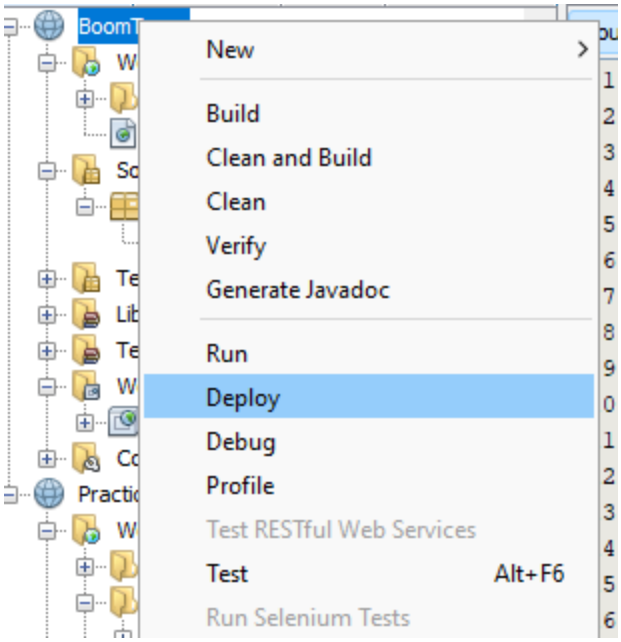


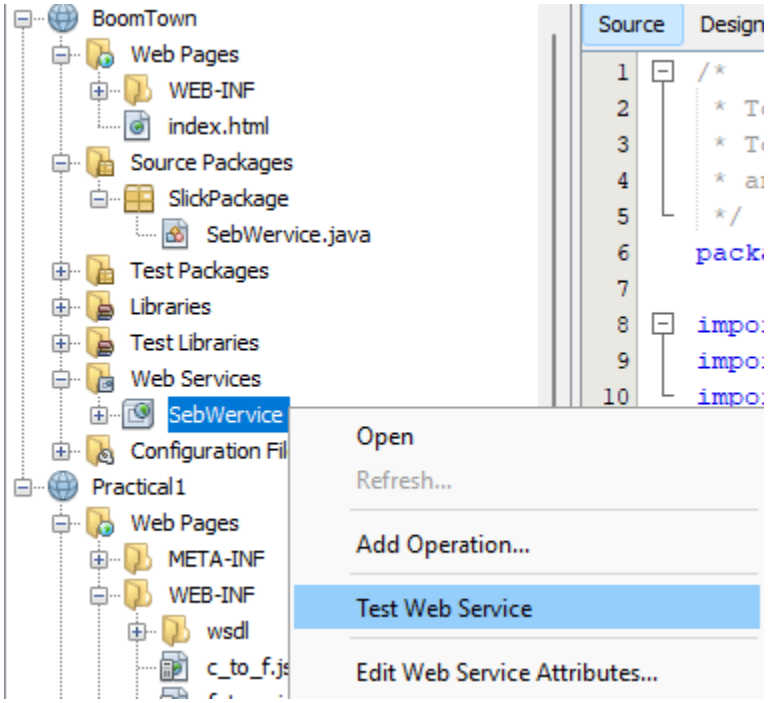
SebWervice.java

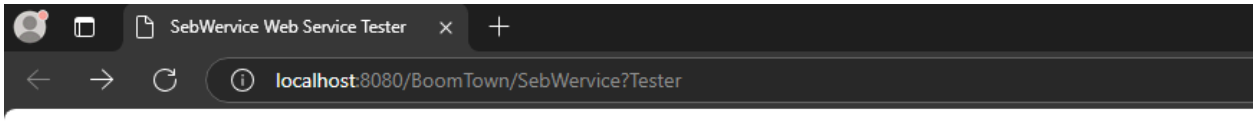
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package SlickPackage;

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

/**
 *
 * @author Admin
 */
```

```
@WebService(serviceName = "SebWervice")
public class SebWervice {

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

        @WebMethod(operationName = "F_To_C")
        public double F_To_C(@WebParam(name="f") double f){
        double c = (f-32)*1.8;
        return c;
        }

        @WebMethod(operationName = "C_To_F")
        public double C_To_F(@WebParam(name="c") double c){
        double f = (c*1.8)+32;
        return f;
        }
}
```

Click on Design and then Add operation to add the 2 methods created in the SebWervices.java file


New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests


BoomTown
Web Pages
WEB-INF
index.html
Source Packages
SlickPackage
SebWervice.java
Test Packages
Libraries
Test Libraries
Web Services
SebWervice
Configuration Fil
Practical1
Web Pages
META-INF
WEB-INF
wsdl
Source
Design
Open
Refresh...
Add Operation...
Test Web Service
Edit Web Service Attributes...



SebWervice Web Service Tester

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

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

Methods :

public abstract double slickpackage.SebWervice.cToF(double)
cToF (25)

---

public abstract double slickpackage.SebWervice.fToC(double)
fToC ( )

---

public abstract java.lang.String slickpackage.SebWervice.hello(java.lang.String)
hello ( )

cToF Method invocation

Method parameter(s)

Type	Value
double	25

Method returned

double : "77.0"

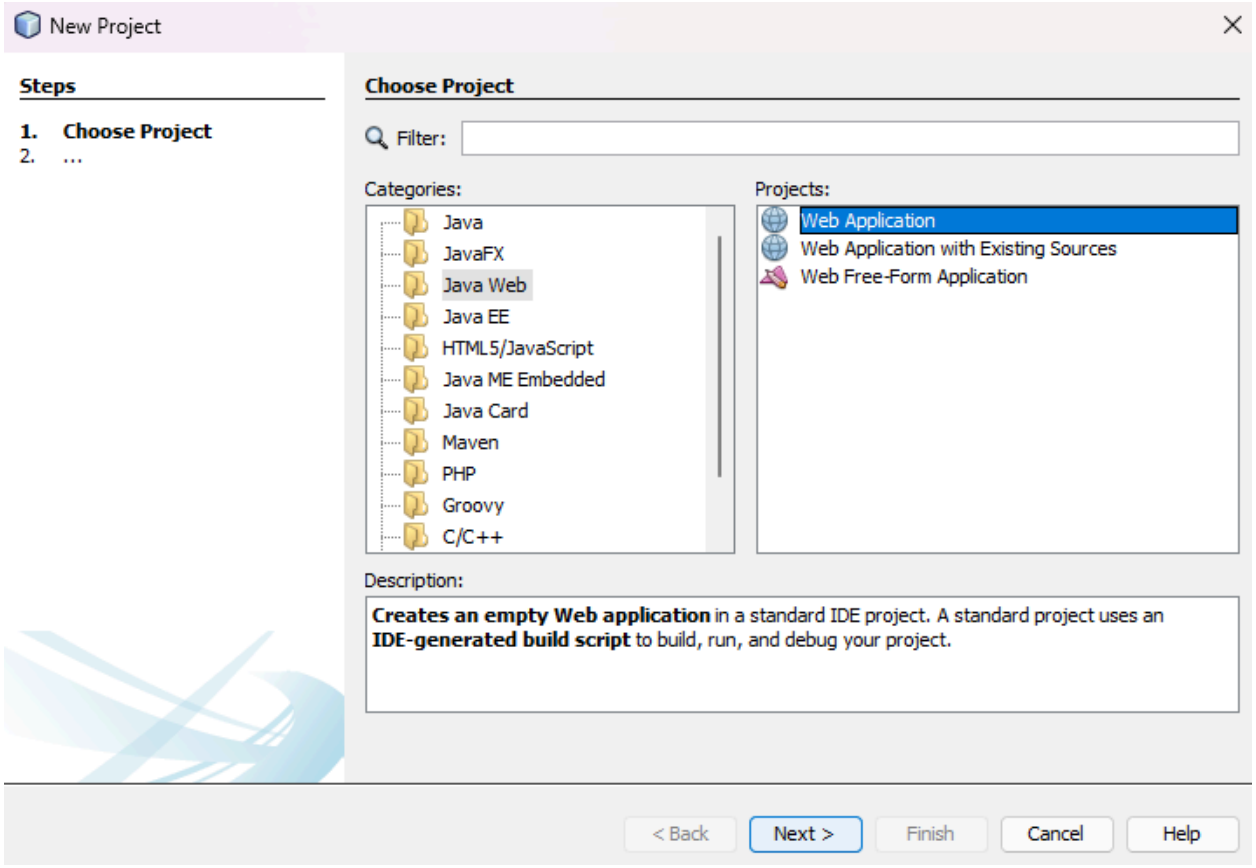
SOAP Request

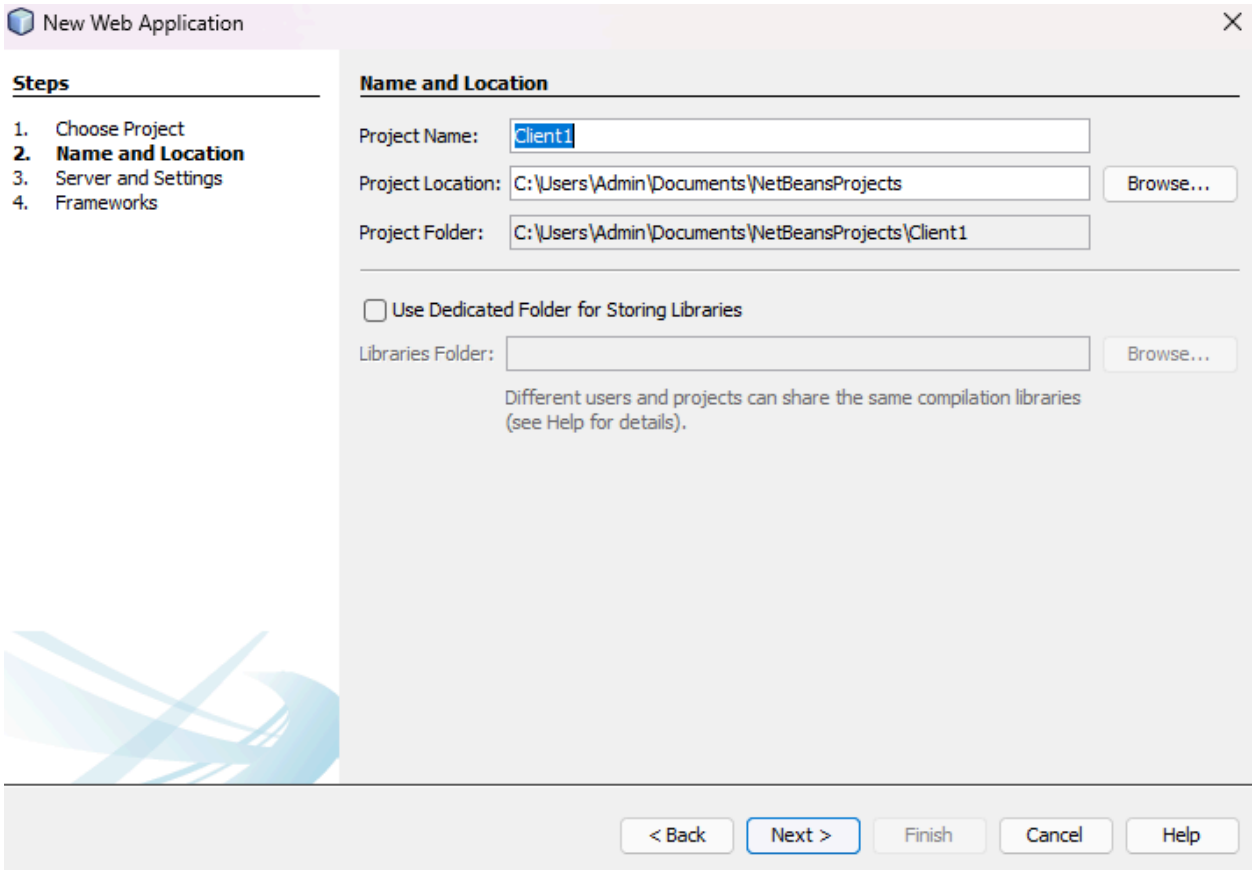
```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:C_To_F xmlns:ns2="http://SlickPackage/">
            <c>25.0</c>
        </ns2:C_To_F>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

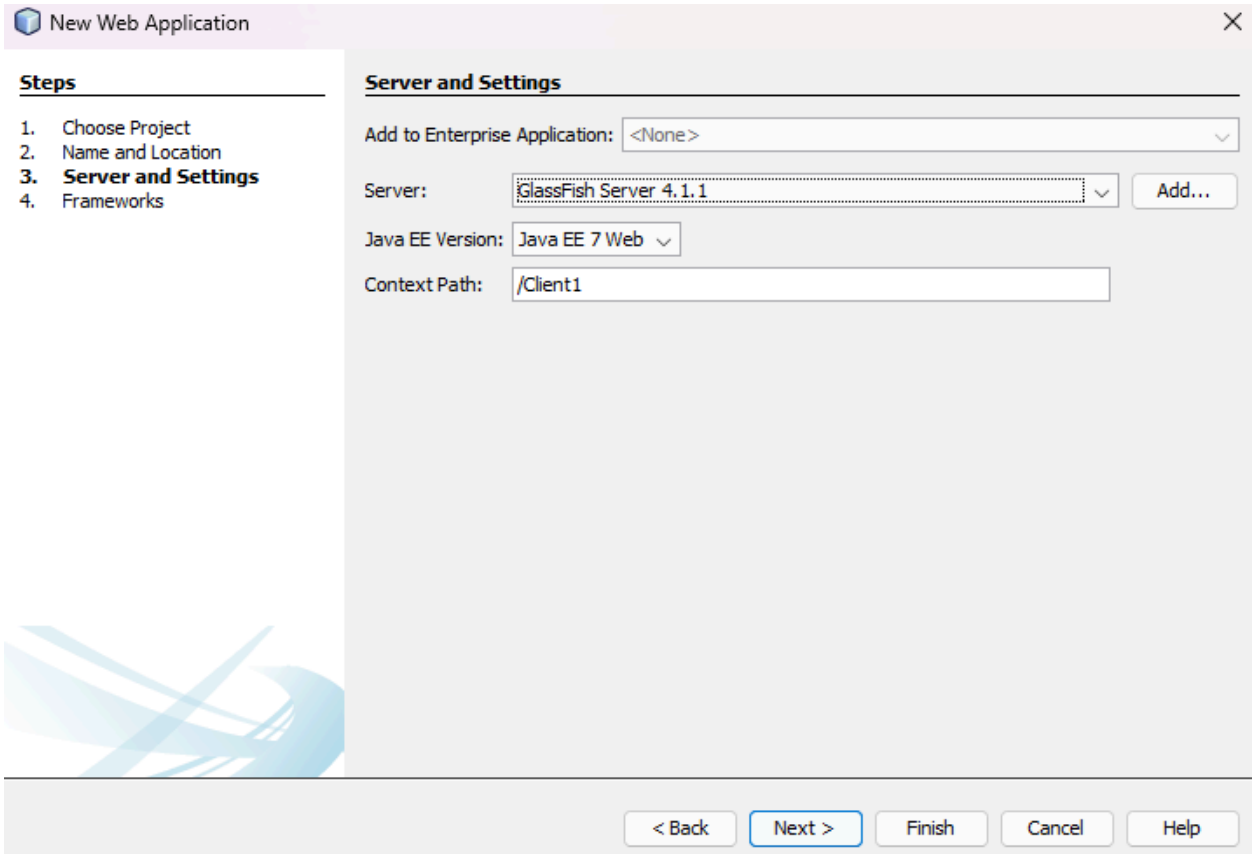
SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:C_To_FResponse xmlns:ns2="http://SlickPackage/">
            <return>77.0</return>
        </ns2:C_To_FResponse>
    </S:Body>
</S:Envelope>
```

Create new project



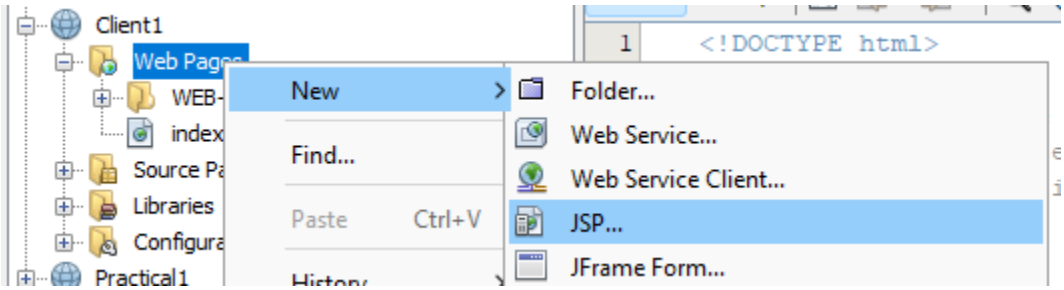

New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: Client1
Project Location: C:\Users\Admin\Documents\NetBeansProjects
Browse...
Project Folder: C:\Users\Admin\Documents\NetBeansProjects\Client1
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Browse...
Different users and projects can share the same compilation libraries
(see Help for details).
< Back
Next >
Finish
Cancel
Help

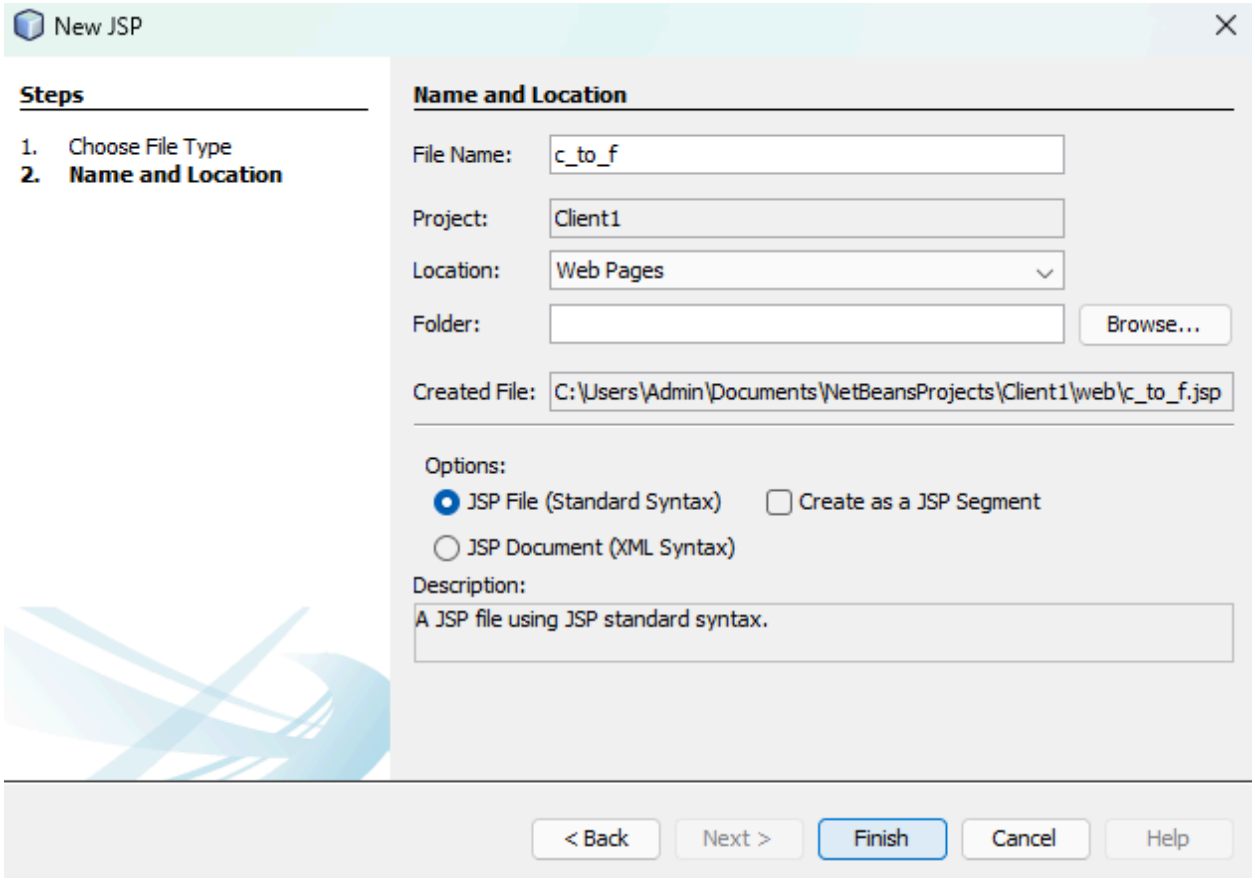


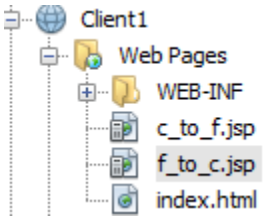
Index.html

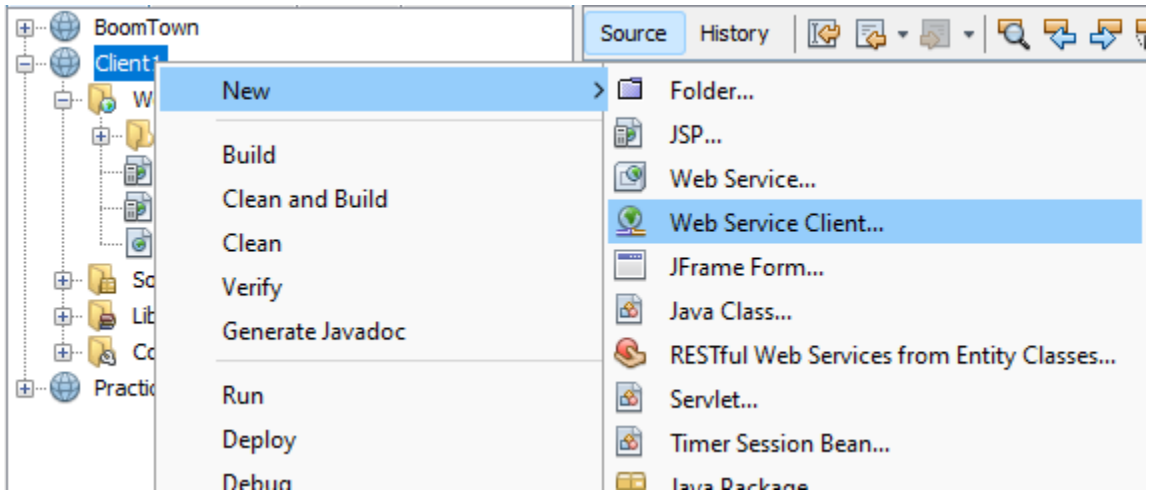
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
```

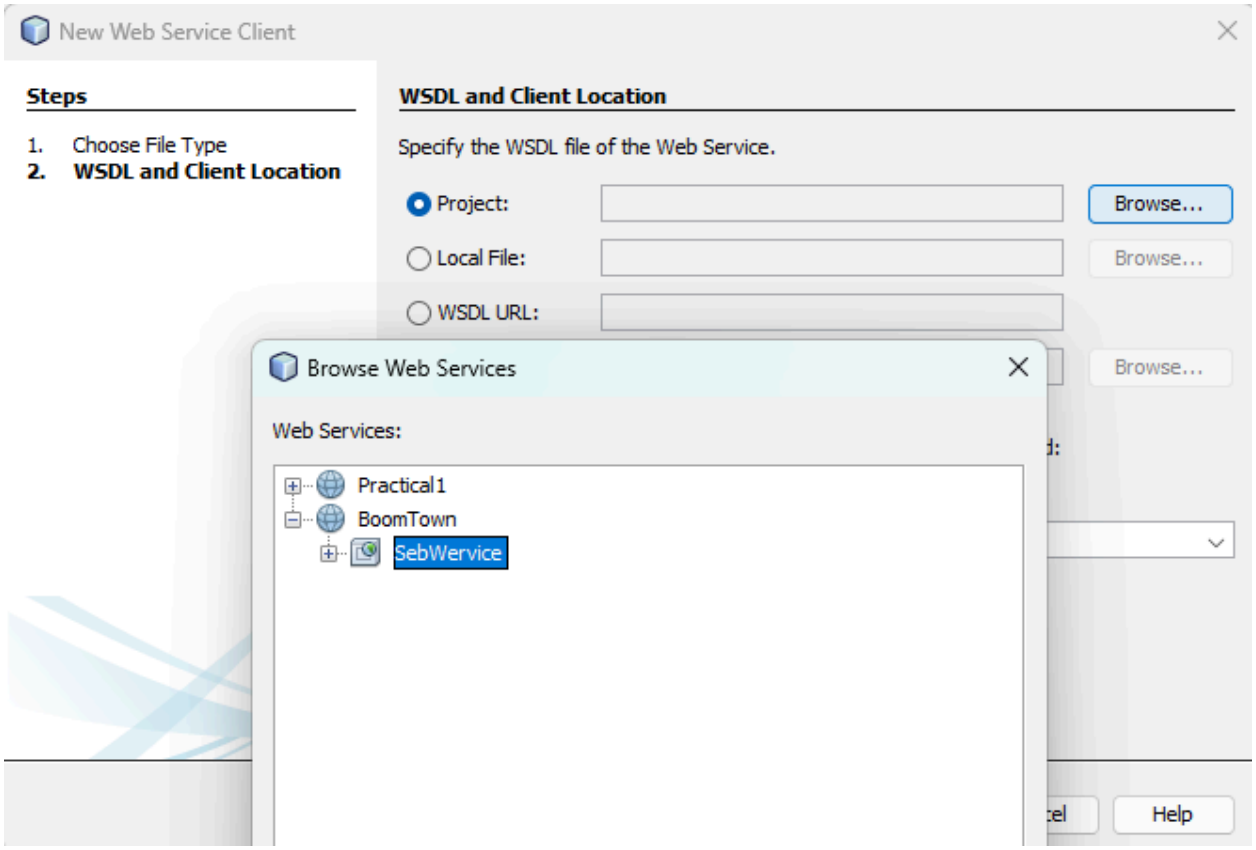
```
        <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        </head>
        <body>
        <div>TODO write content</div>
        <form>
        <br> <input type="text" name="data">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

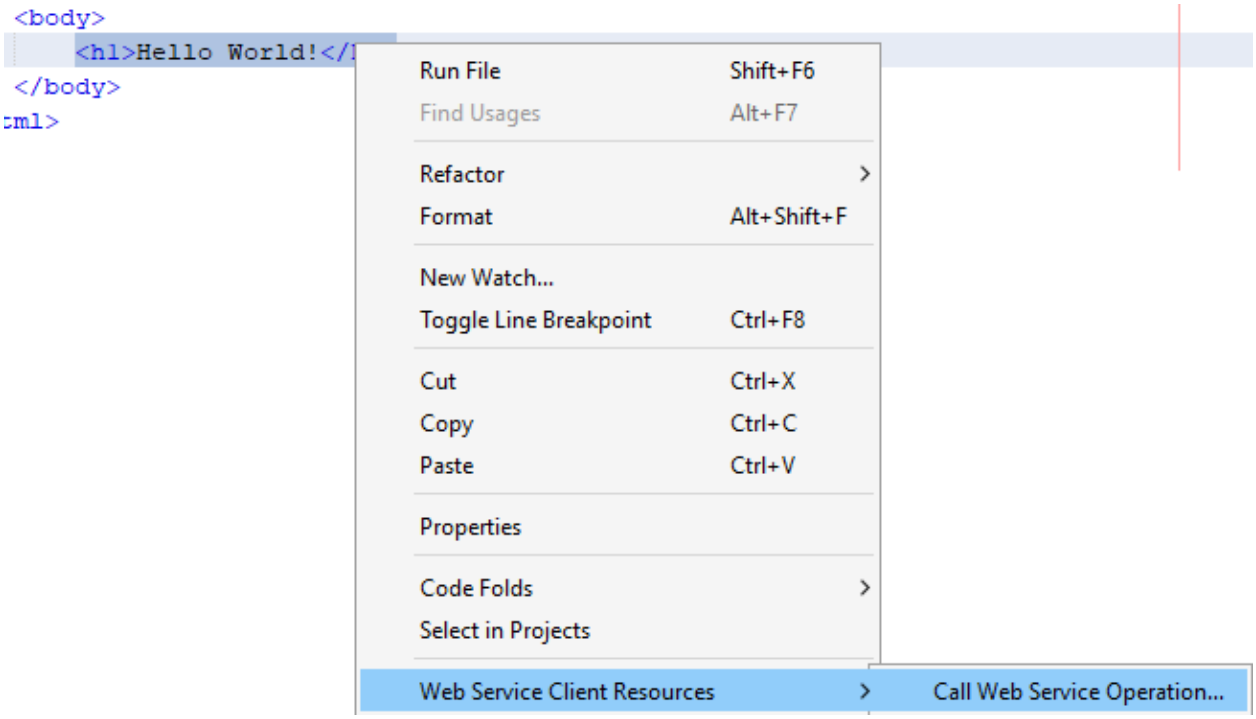


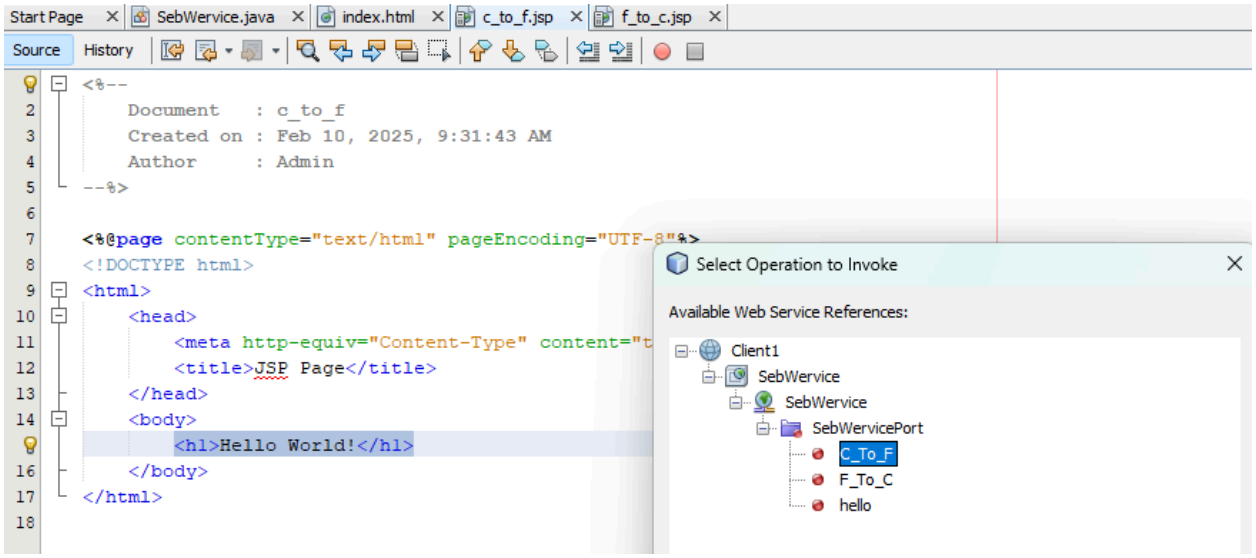

New JSP
Steps
1. Choose File Type
2. Name and Location
Name and Location
File Name: c_to_f
Project: Client1
Location: Web Pages
Folder:
Browse...
Created File: C:\Users\Admin\Documents\NetBeansProjects\Client1\web\c_to_f.jsp
Options:
JSP File (Standard Syntax)
Create as a JSP Segment
JSP Document (XML Syntax)
Description:
A JSP file using JSP standard syntax.
< Back
Next >
Finish
Cancel
Help


Client1
Web Pages
WEB-INF
c_to_f.jsp
f_to_c.jsp
index.html


BoomTown
Client1
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Source
History
Folder...
JSP...
Web Service...
Web Service Client...
JFrame Form...
Java Class...
RESTful Web Services from Entity Classes...
Servlet...
Timer Session Bean...


New Web Service Client
Steps
1. Choose File Type
2. WSDL and Client Location
WSDL and Client Location
Specify the WSDL file of the Web Service.
Project:
Local File:
WSDL URL:
Browse...
Browse...
Browse...
Browse Web Services
Web Services:
Practical1
BoomTown
SebWervice
Help


<body>
<h1>Hello World!</h1>
</body>
</html>
Run File Shift+F6
Find Usages Alt+F7
Refactor
Format Alt+Shift+F
New Watch...
Toggle Line Breakpoint Ctrl+F8
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Properties
Code Folds
Select in Projects
Web Service Client Resources
Call Web Service Operation...



AUTO GENERATED CODE (changes in grey, do in both)

f_to_c .jsp

```
<%
        String d = request.getParameter("data");
        Integer fer = Integer.parseInt(d);
    try {
        temperature.WSTemp_Service service = new
temperature.WSTemp_Service();
        temperature.WSTemp port = service.getWSTempPort();
         // TODO initialize WS operation arguments here
        double f = fer;
        // TODO process result here
        double result = port.fToC(f);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
```

c_to_f.jsp

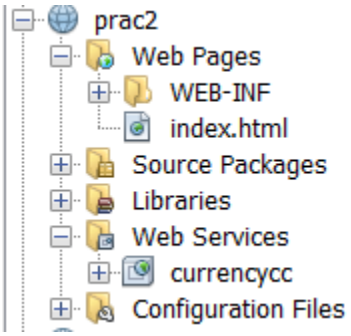
```
<%
```

```
        String d = request.getParameter("data");
        Integer fer = Integer.parseInt(d);
    try {
        temperature.WSTemp_Service service = new
temperature.WSTemp_Service();
        temperature.WSTemp port = service.getWSTempPort();
         // TODO initialize WS operation arguments here
        double f = fer;
        // TODO process result here
        double result = port.cToF(c);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
```

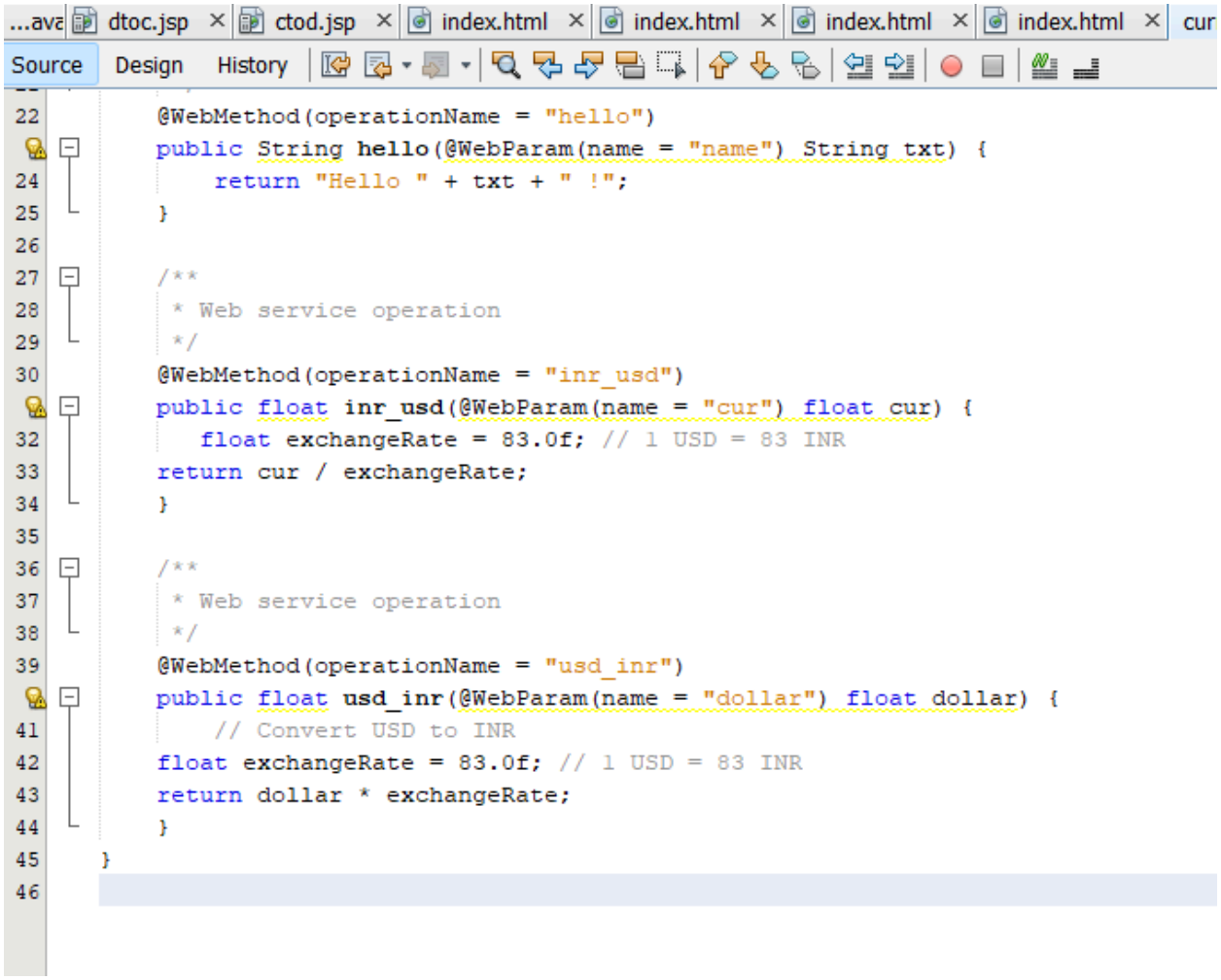
Dollar to INR

INR to Dollar

Step 1:- create a new project then web service (name as currencycc)

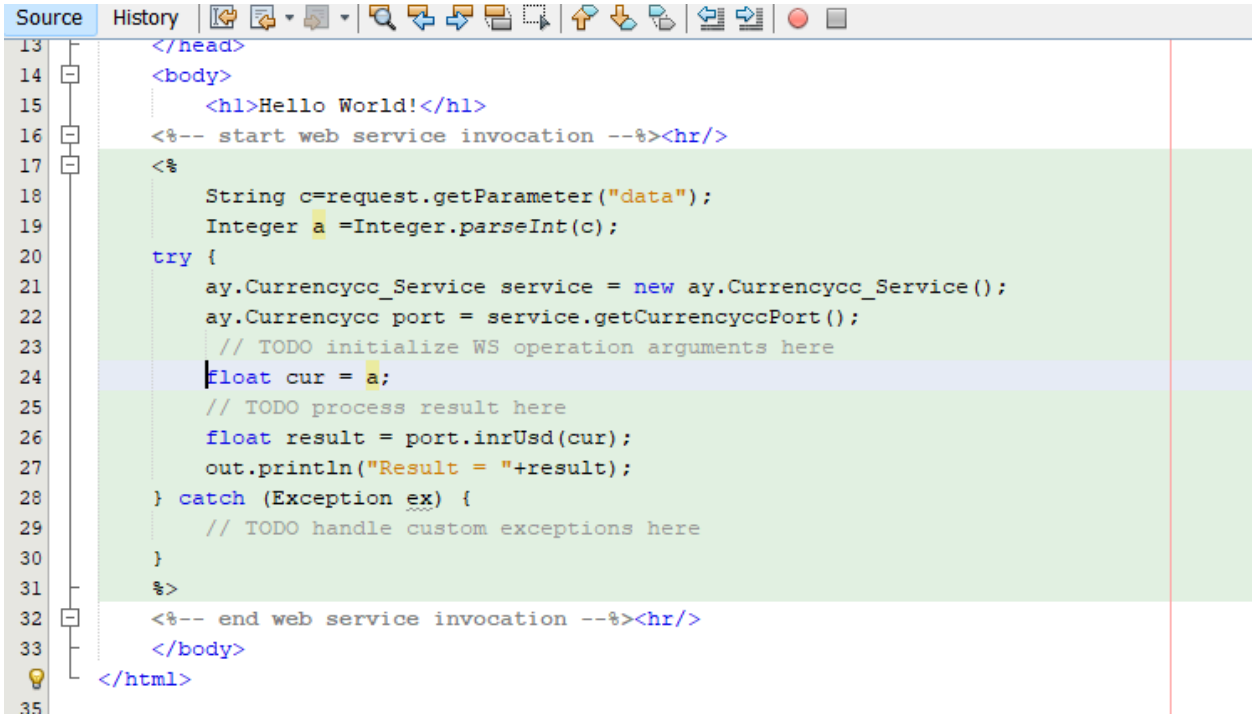


Step 2:- do the following things add the operations by right click on the web service

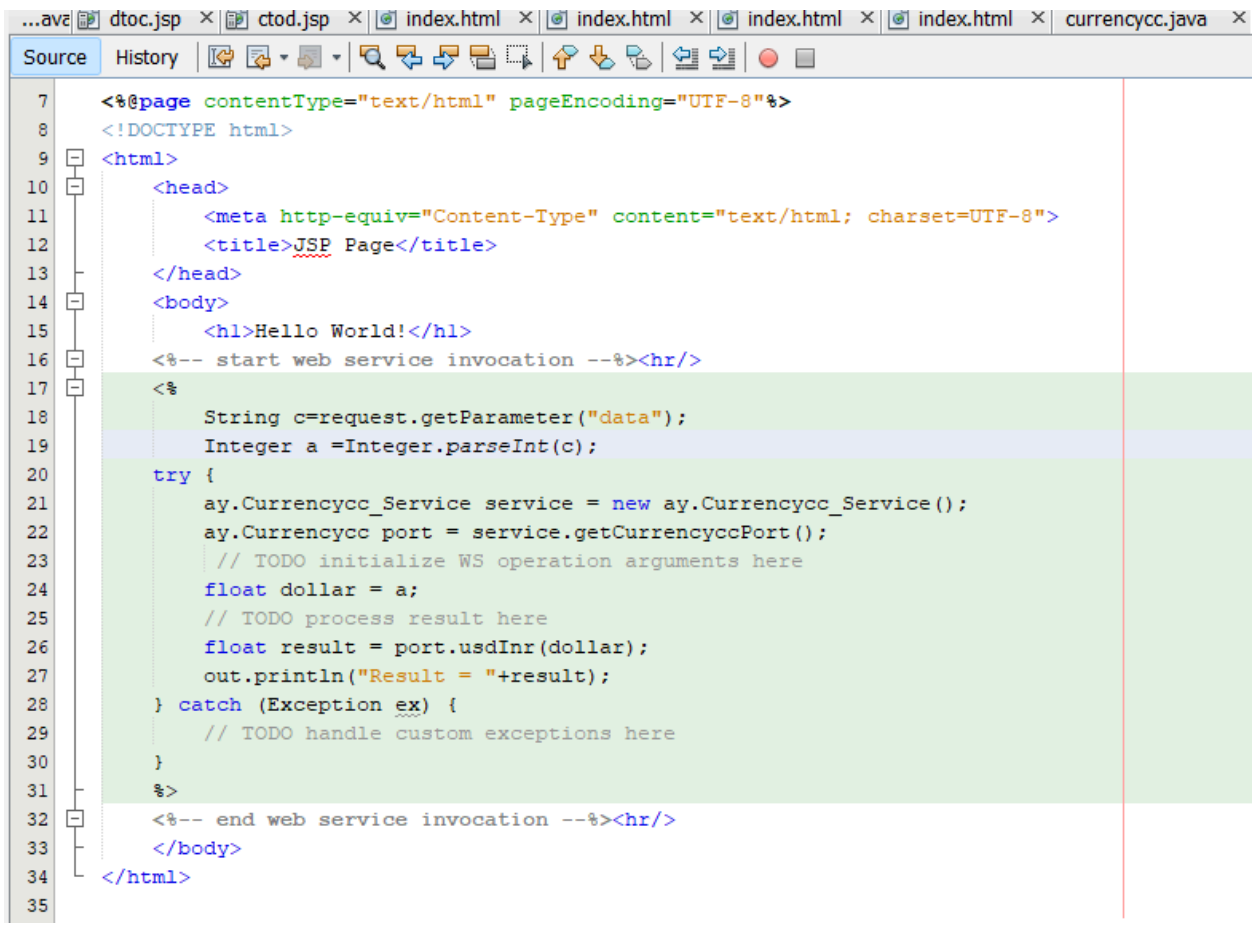


Step 3:- Add the new project and create an webservice client and call the all methods which u have created to the new project client

Step 4:- Create 2 jsp pages as inr_usd &usd_inr and input the method by right click select call operations and proceed further.

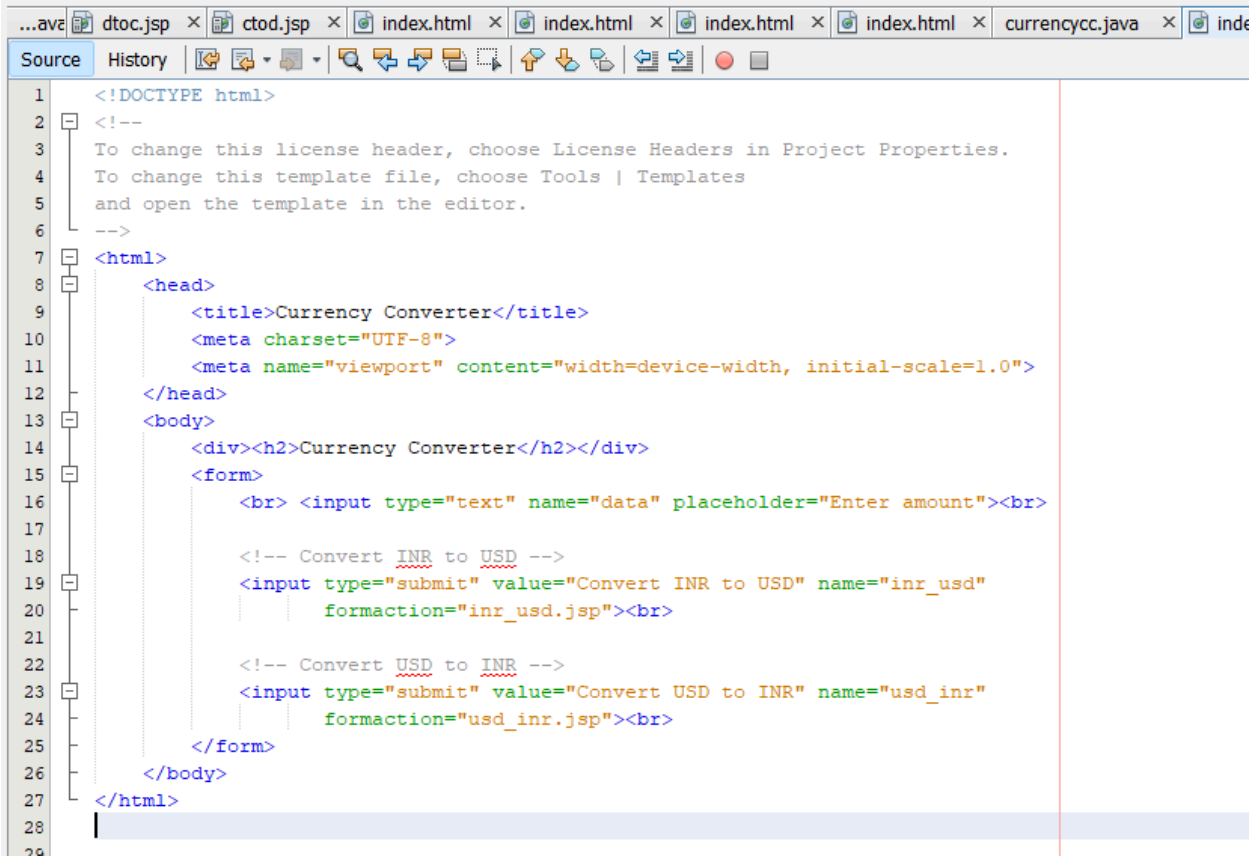


```
    </head>
    <body>
        <h1>Hello World!</h1>
    <%-- start web service invocation --%><hr/>
    <%
        String c=request.getParameter("data");
        Integer a =Integer.parseInt(c);
    try {
        ay.Currencycc_Service service = new ay.Currencycc_Service();
        ay.Currencycc port = service.getCurrencyccPort();
         // TODO initialize WS operation arguments here
        float cur = a;
        // TODO process result here
        float result = port.inrUsd(cur);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
    </body>
</html>
```



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    <%-- start web service invocation --%><hr/>
    <%
        String c=request.getParameter("data");
        Integer a =Integer.parseInt(c);
    try {
        ay.Currencycc_Service service = new ay.Currencycc_Service();
        ay.Currencycc port = service.getCurrencyccPort();
         // TODO initialize WS operation arguments here
        float dollar = a;
        // TODO process result here
        float result = port.usdInr(dollar);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
    </body>
</html>
```

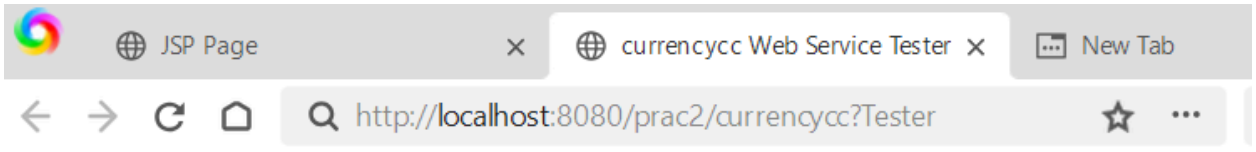
Then make changes in your index.html




```
<!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>Currency Converter</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div><h2>Currency Converter</h2></div>
        <form>
            <br> <input type="text" name="data" placeholder="Enter amount"><br>

            <!-- Convert INR to USD -->
            <input type="submit" value="Convert INR to USD" name="inr_usd"
                   formaction="inr_usd.jsp"><br>

            <!-- Convert USD to INR -->
            <input type="submit" value="Convert USD to INR" name="usd_inr"
                   formaction="usd_inr.jsp"><br>
        </form>
    </body>
</html>
```

Output:-

At the server side



currencycc Web Service Tester

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

To invoke an operation, fill the method parameter(s) input boxes and click on the button labe

Methods :

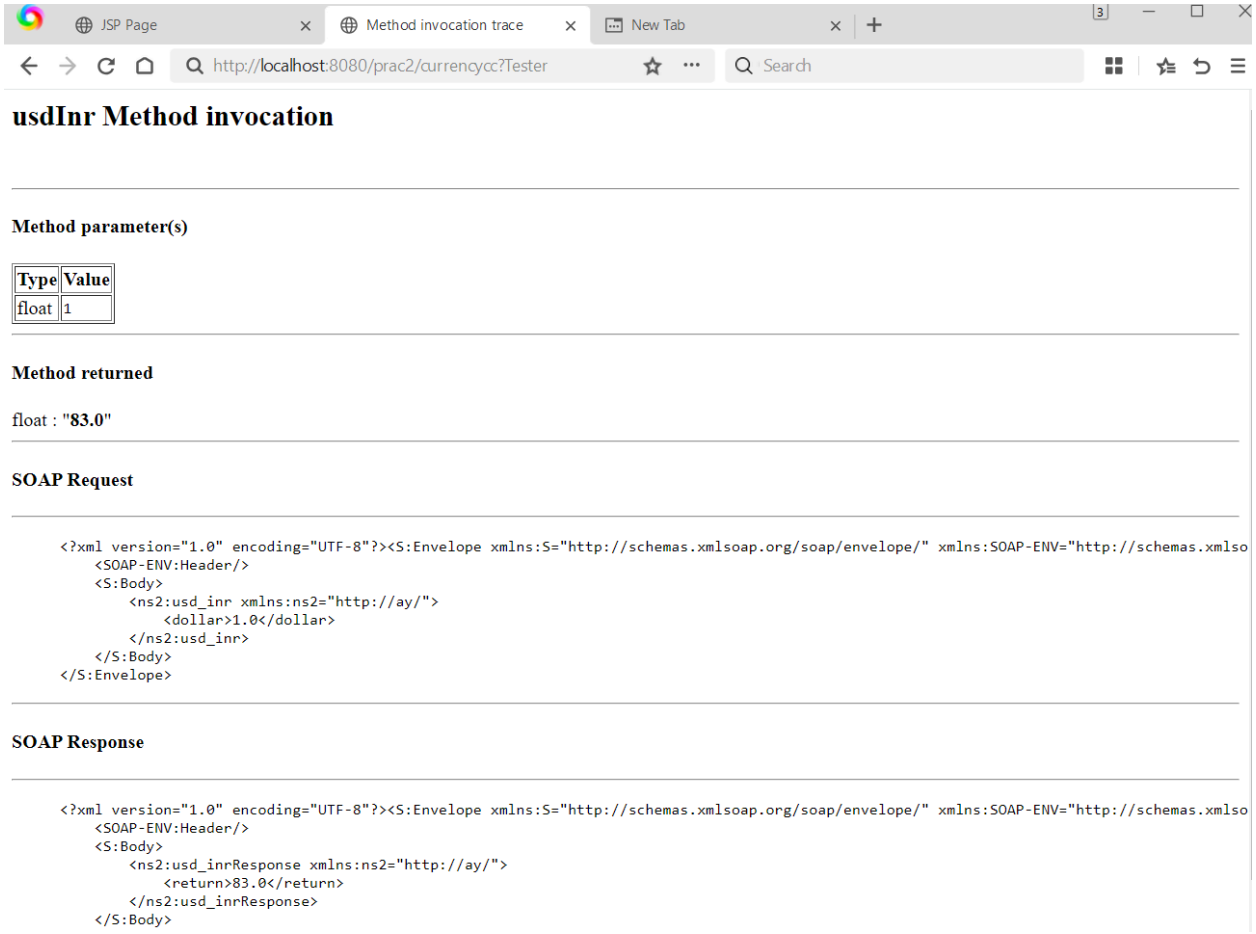
public abstract java.lang.String ay.Currencycc.hello(java.lang.String)
hello ( )

---

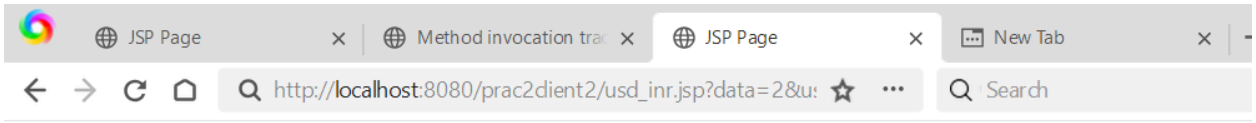
public abstract float ay.Currencycc.usdInr(float)
usdInr (1)

---

public abstract float ay.Currencycc.inrUsd(float)
inrUsd ( )



Client side output :-

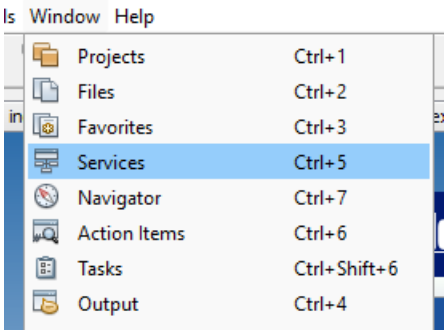


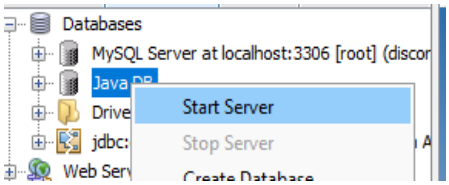
Hello World!

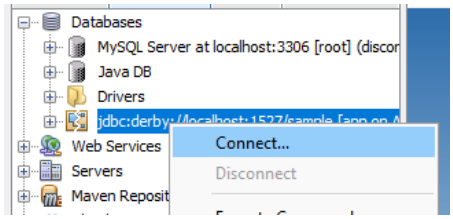
Result = 166.0

Practical 2

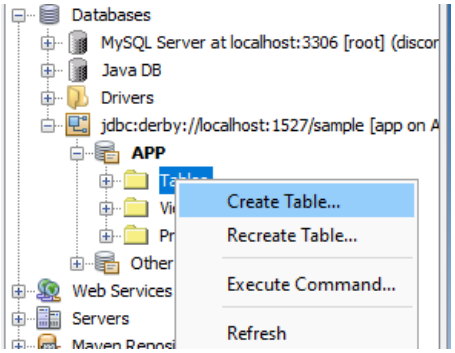
Open services and start the Java DB and sample server



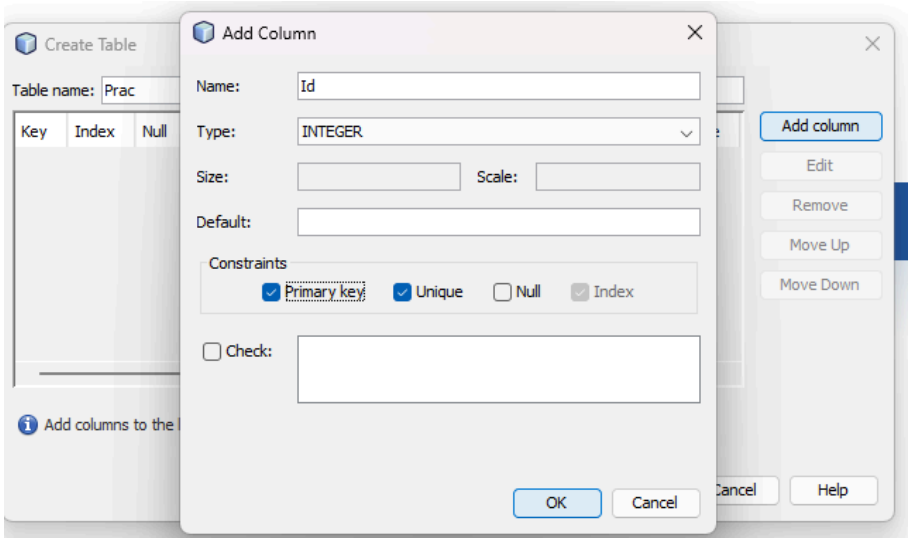


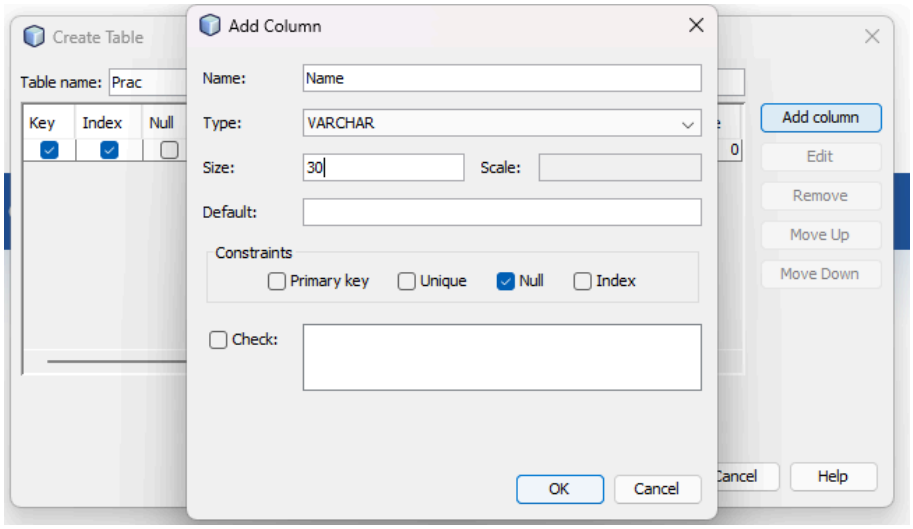


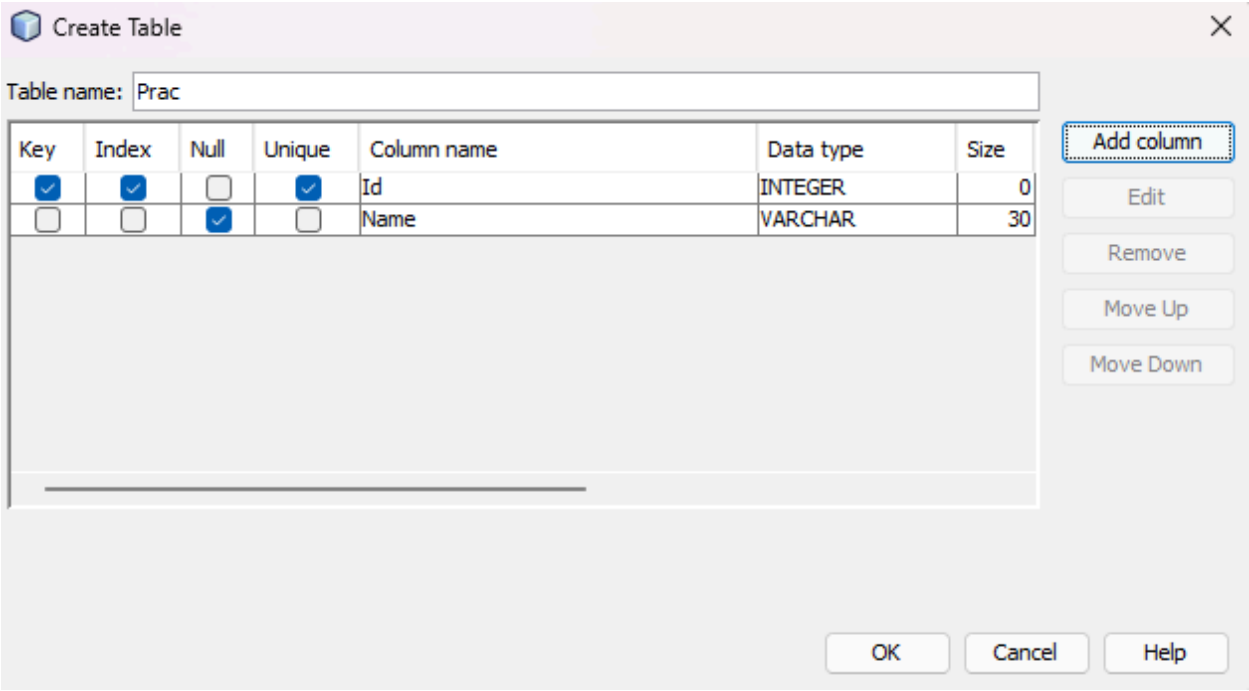
In the sample server, create a table

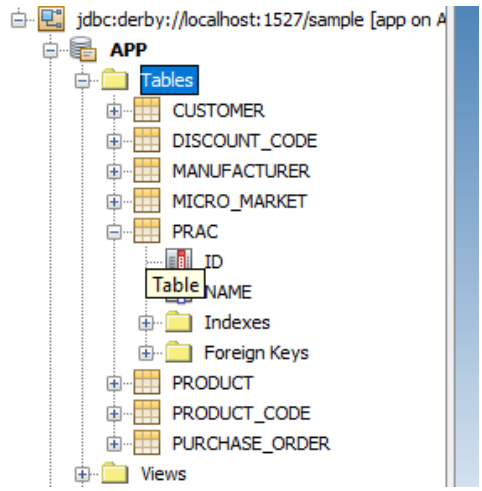


Add columns

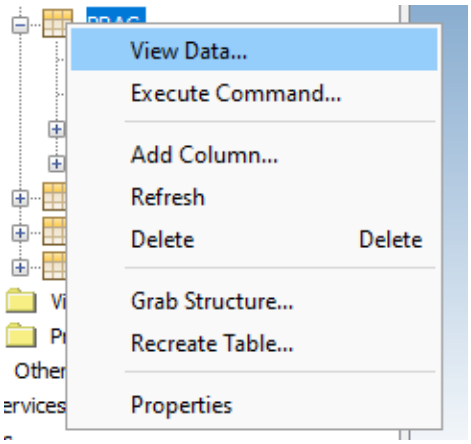

Create Table
Table name: Prac
Key
Index
Null
Add column
Edit
Remove
Move Up
Move Down
Add columns to the
Cancel
Help
Add Column
Name: Id
Type: INTEGER
Size:
Scale:
Default:
Constraints
Primary key
Unique
Null
Index
Check:
OK
Cancel


Create Table
Table name: Prac
Key
Index
Null
Add column
Edit
Remove
Move Up
Move Down
Cancel
Help
Add Column
Name: Name
Type: VARCHAR
Size: 30
Scale:
Default:
Constraints
Primary key
Unique
Null
Index
Check:
OK
Cancel

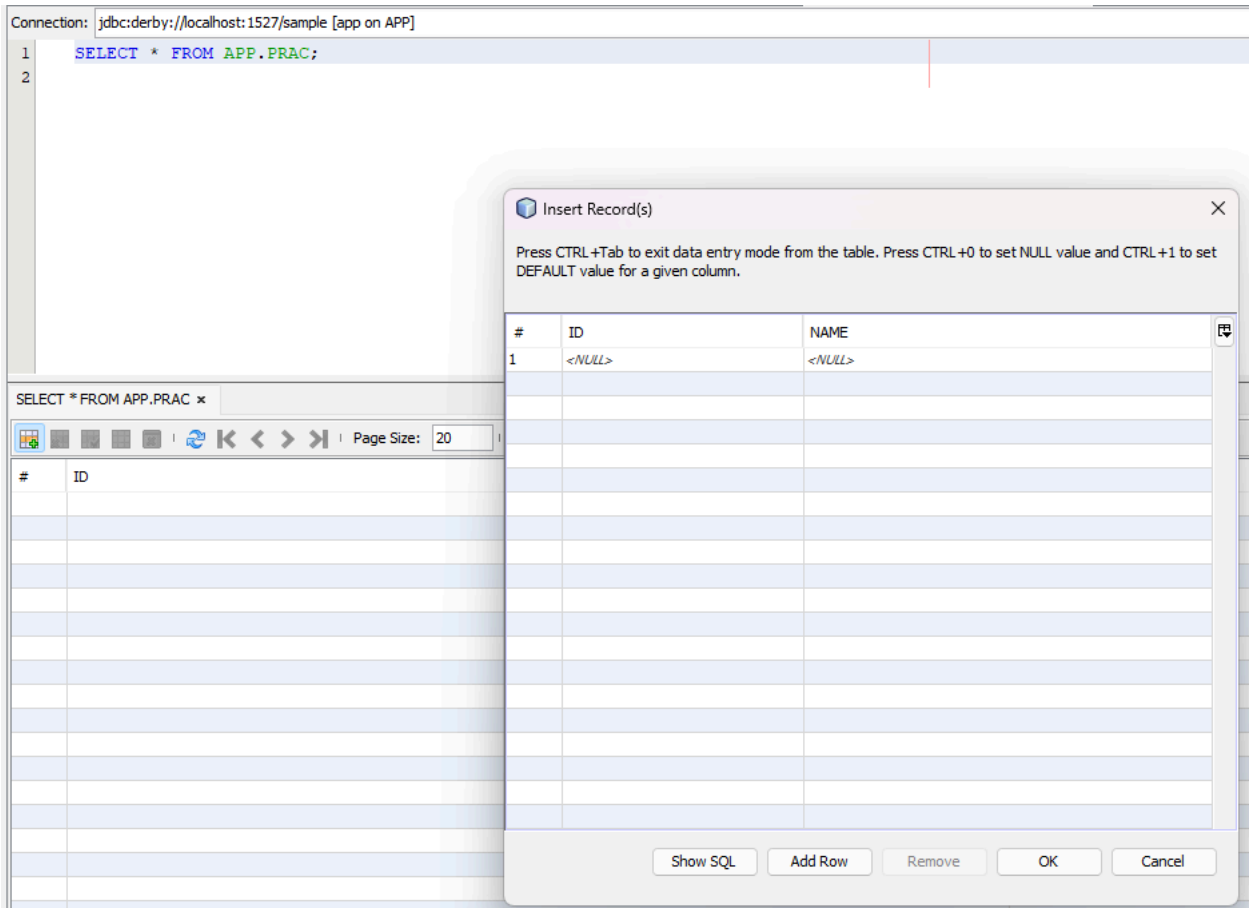




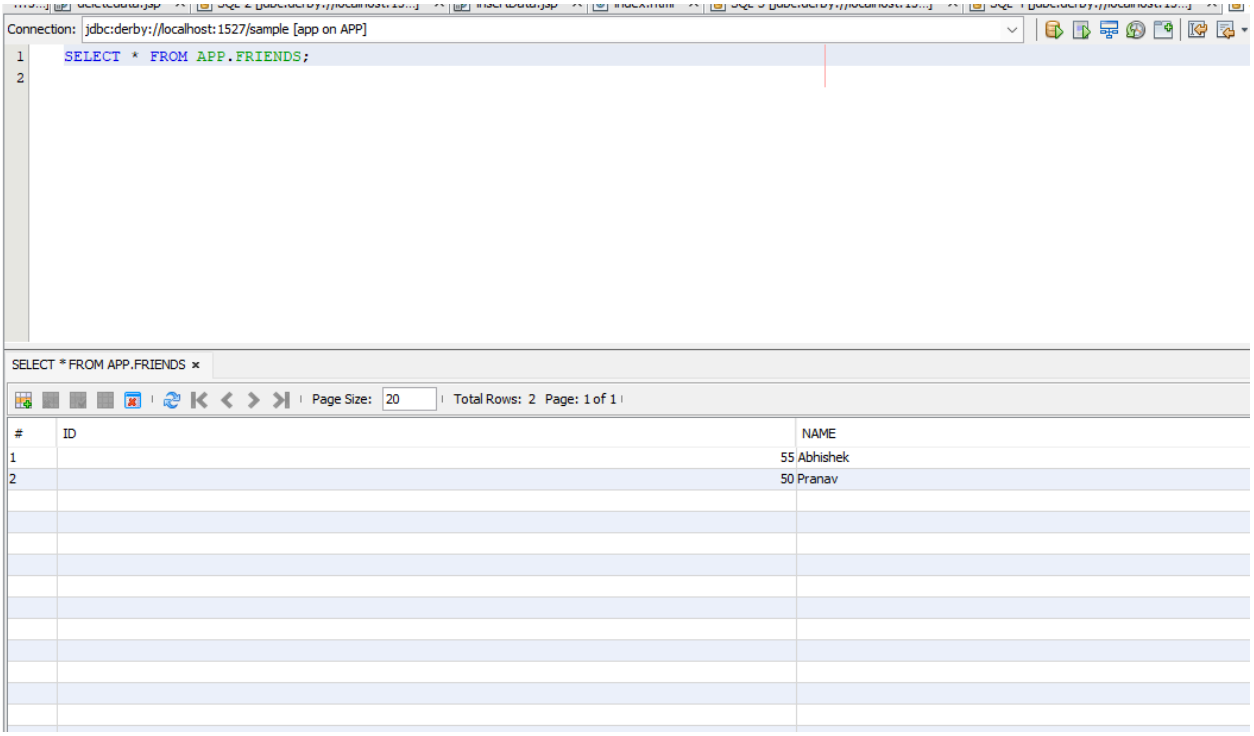
Right click on the created table and click on view data



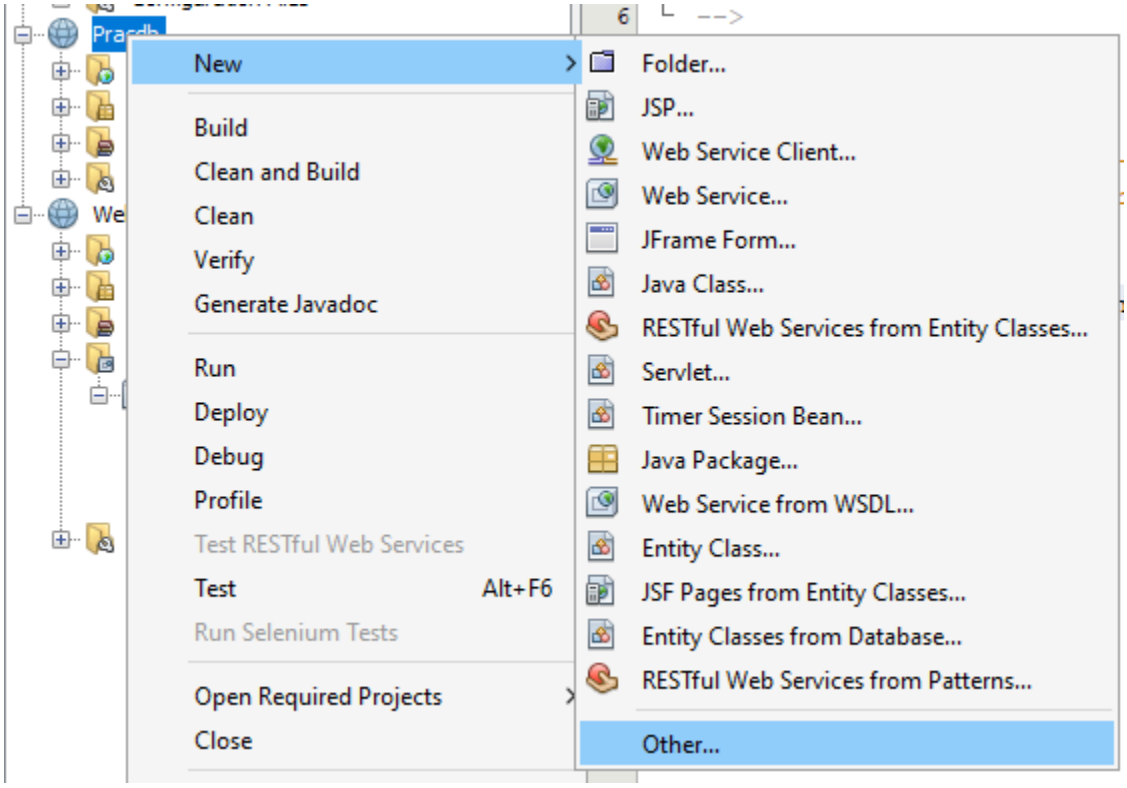
Click on the highlighted thing to insert a record into the table

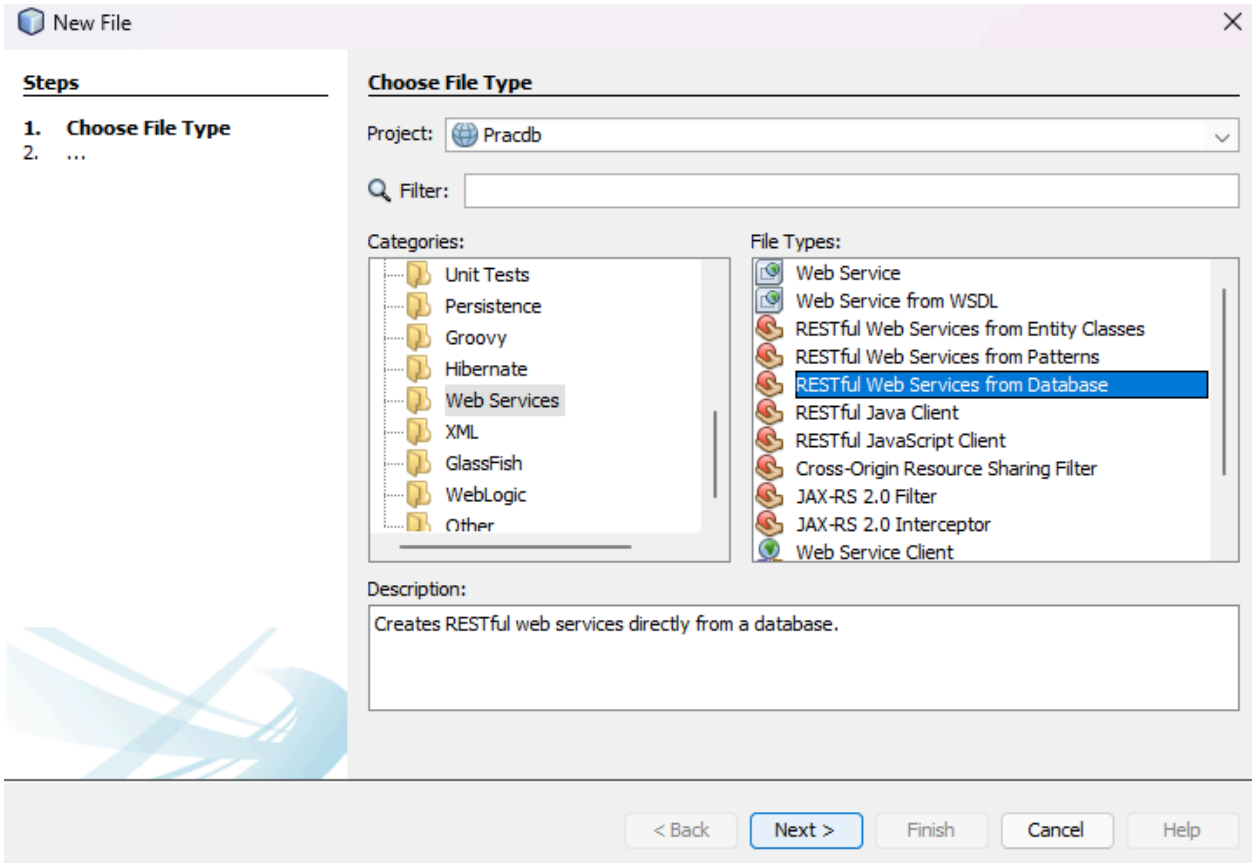


Click on Add Row for every new record

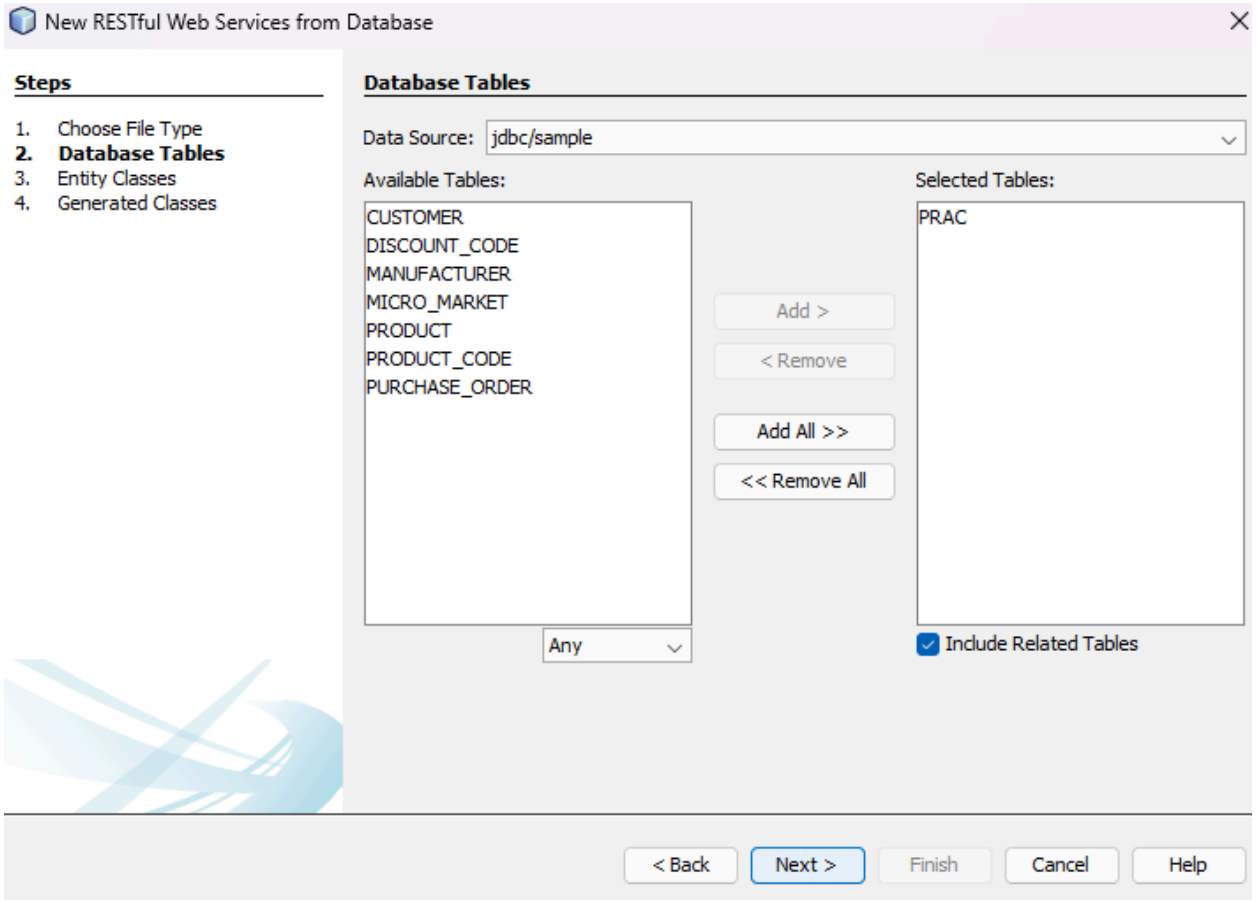

Connection: jdbc:derby://localhost:1527/sample [app on APP]
1 SELECT * FROM APP.FRIENDS;
2
SELECT * FROM APP.FRIENDS ×
Page Size: 20
Total Rows: 2 Page: 1 of 1
ID NAME
1 55 Abhishek
2 50 Pranav

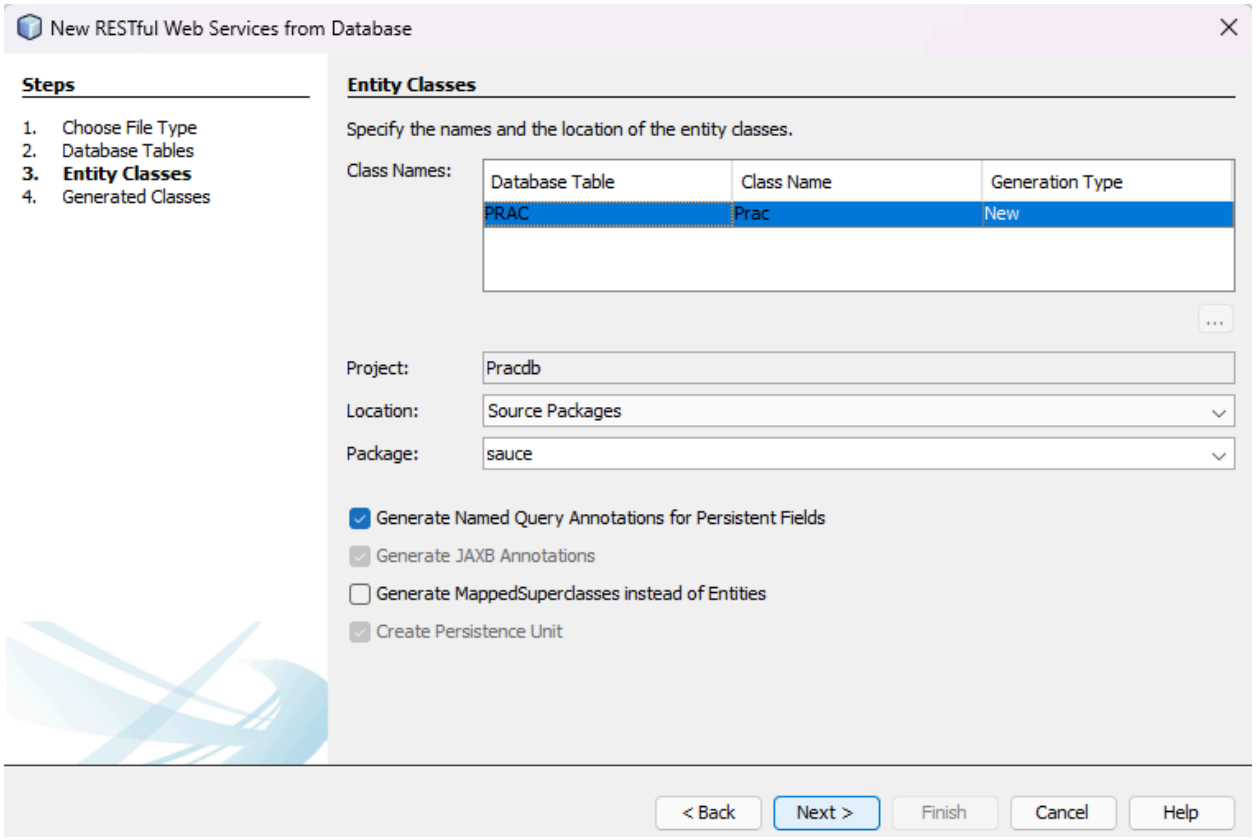
Create a restful webservice from database

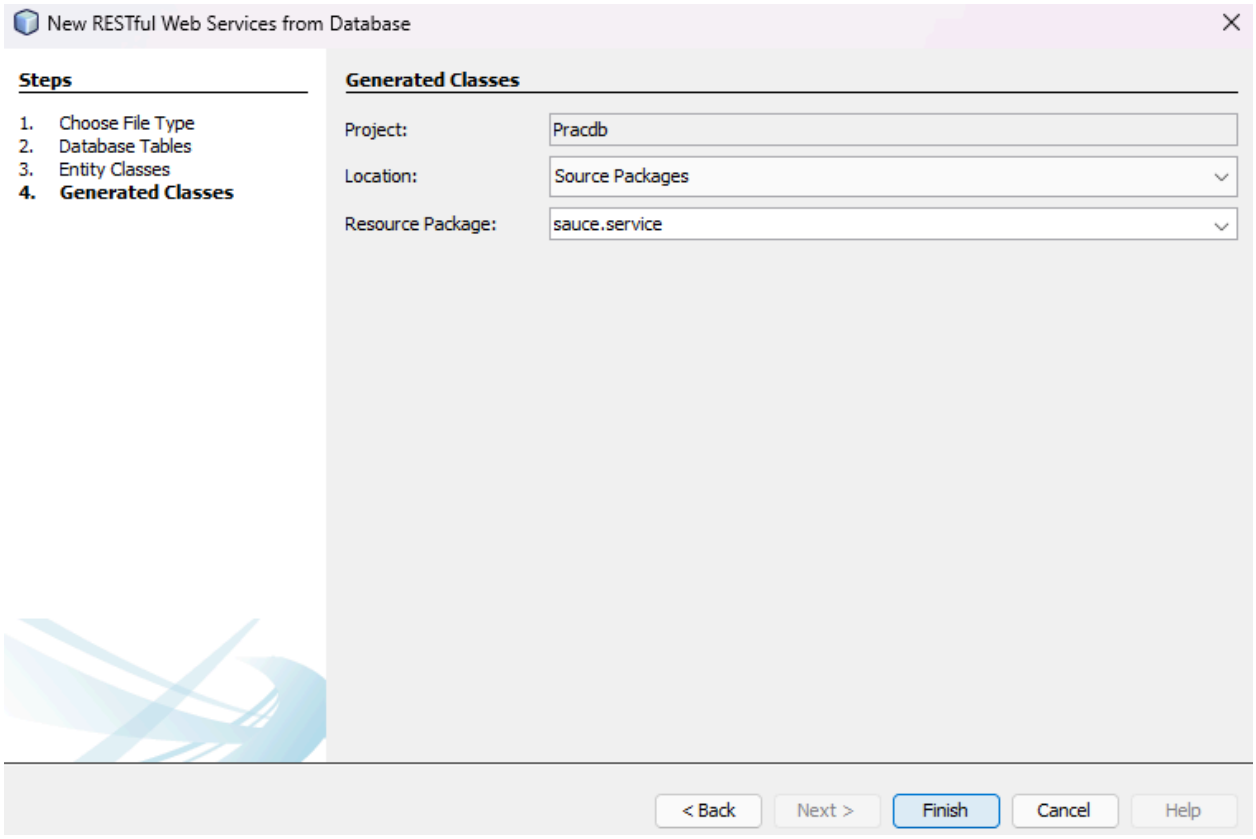


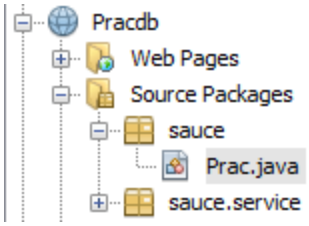


Add the created table

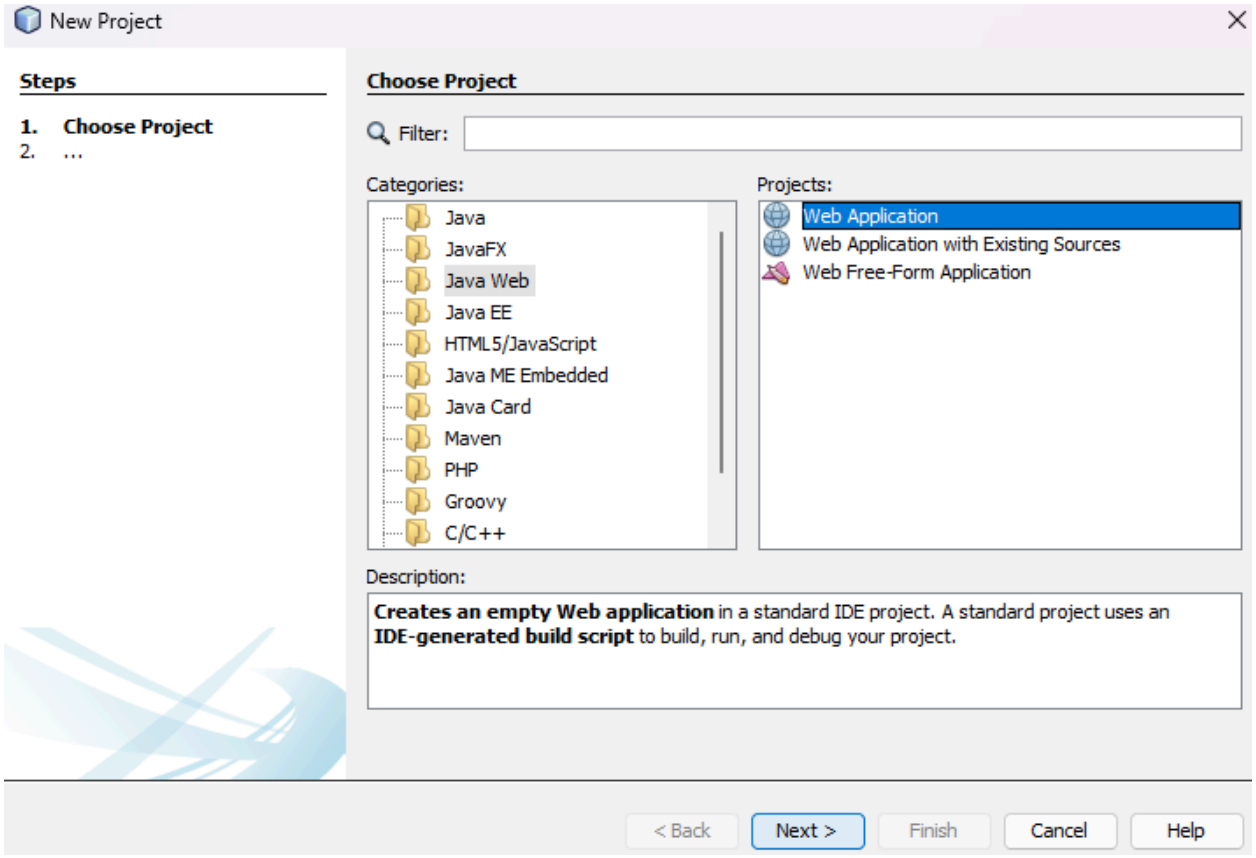


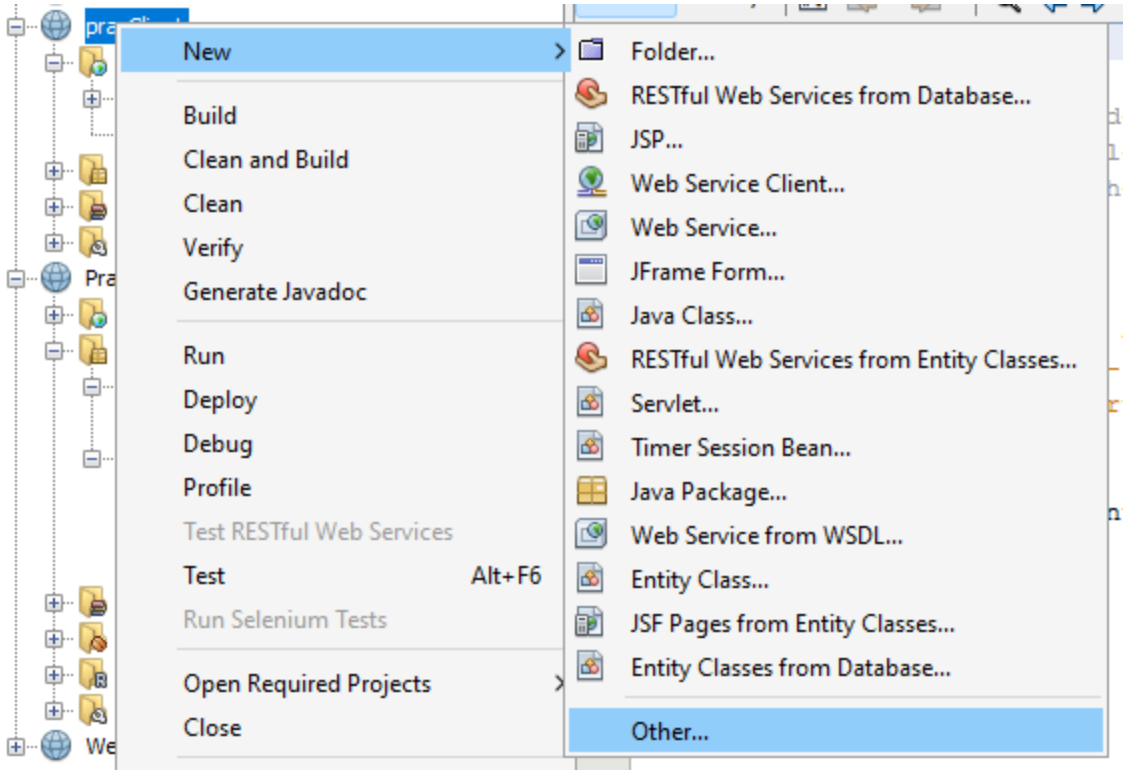

New RESTful Web Services from Database
Steps
1. Choose File Type
2. Database Tables
3. Entity Classes
4. Generated Classes
Entity Classes
Specify the names and the location of the entity classes.
Class Names:
Database Table
Class Name
Generation Type
PRAC
Prac
New
Project:
Pracdb
Location:
Source Packages
Package:
sauce
Generate Named Query Annotations for Persistent Fields
Generate JAXB Annotations
Generate MappedSuperclasses instead of Entities
Create Persistence Unit
< Back
Next >
Finish
Cancel
Help


New RESTful Web Services from Database
Steps
1. Choose File Type
2. Database Tables
3. Entity Classes
4. Generated Classes
Generated Classes
Project:
Pracdb
Location:
Source Packages
Resource Package:
sauce.service
< Back
Next >
Finish
Cancel
Help

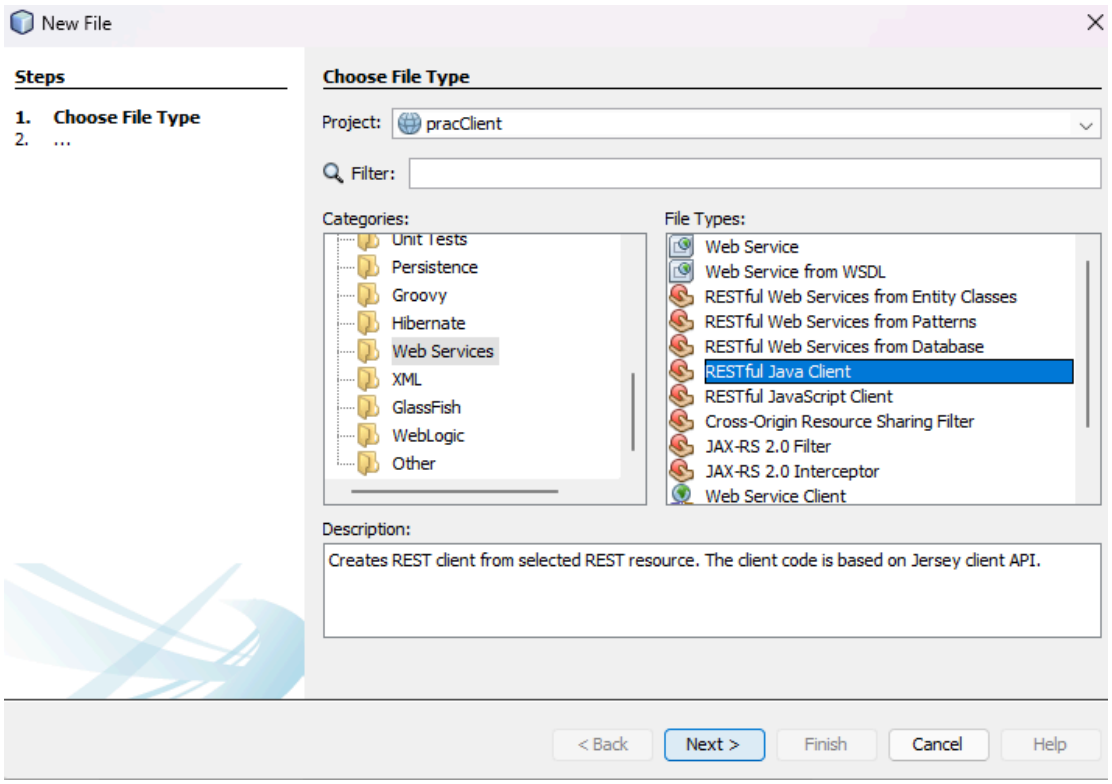


Create a 2nd project



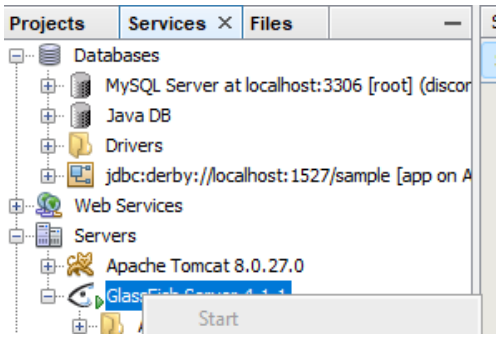


Create a restful java client

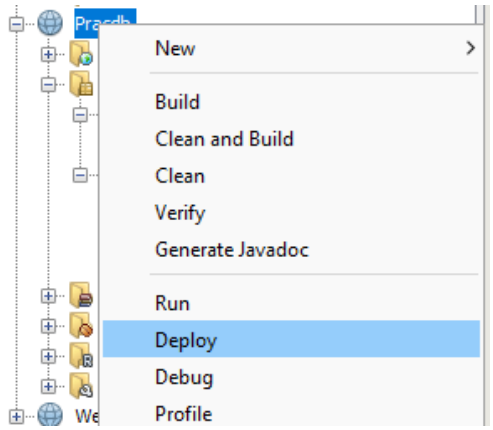


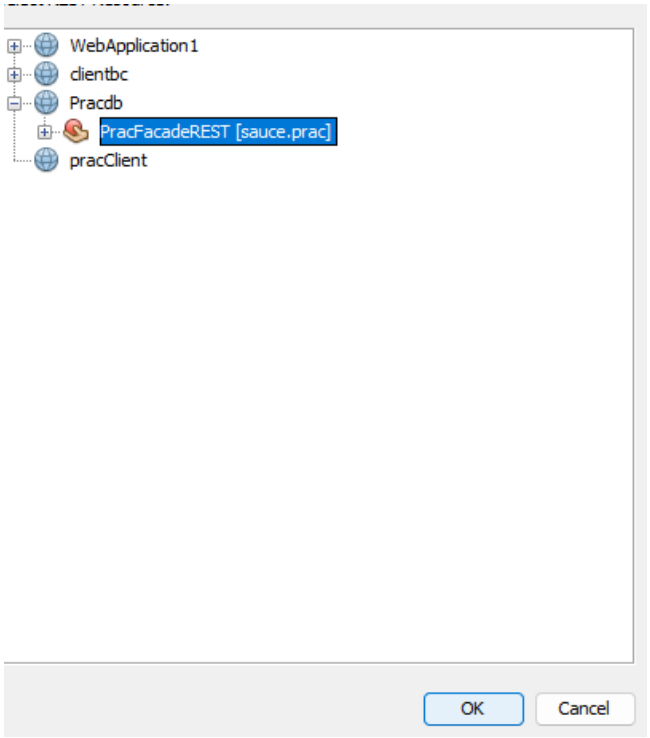
Enter the following command in cmd admin mode
netstat -aon | find ":808" | find "LISTENING"

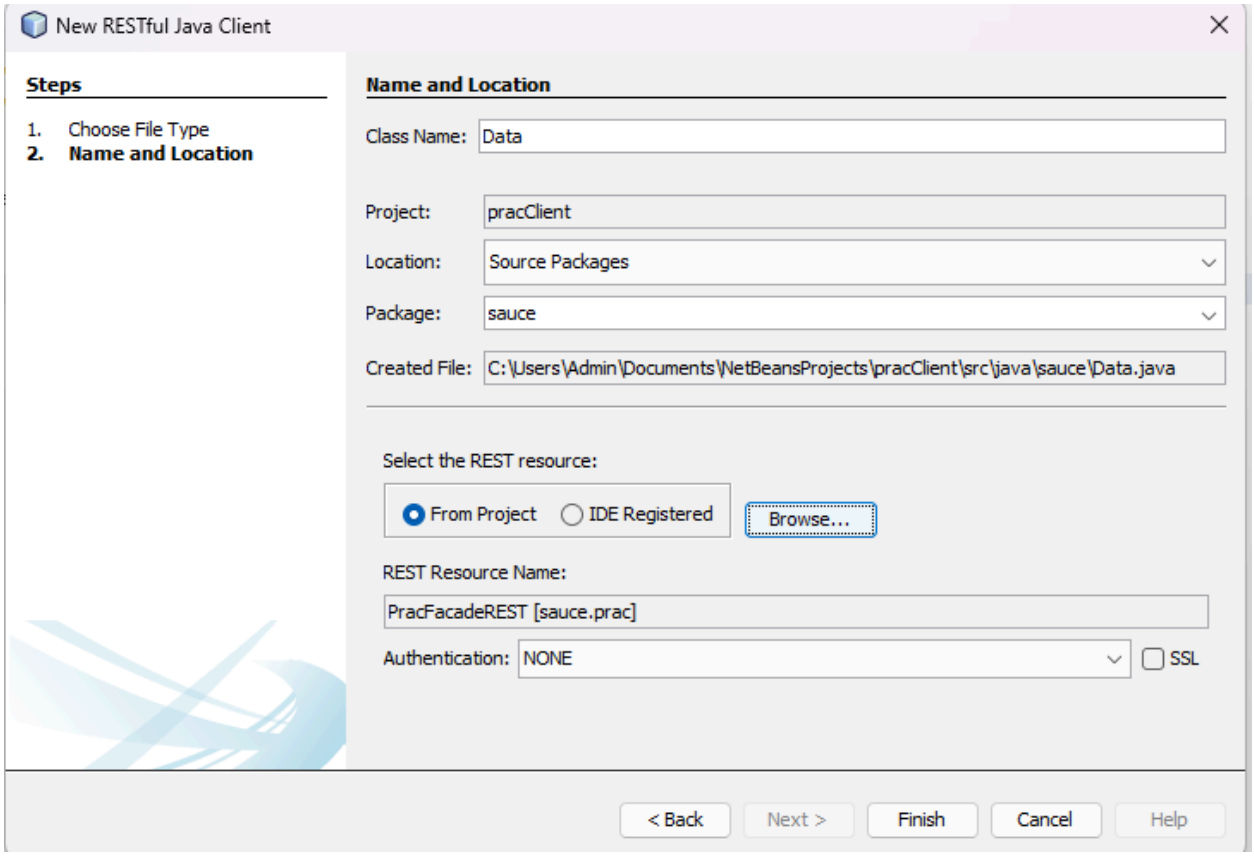
Delete the process which contains the same port as the glassfish server and start the glassfish server in the services section

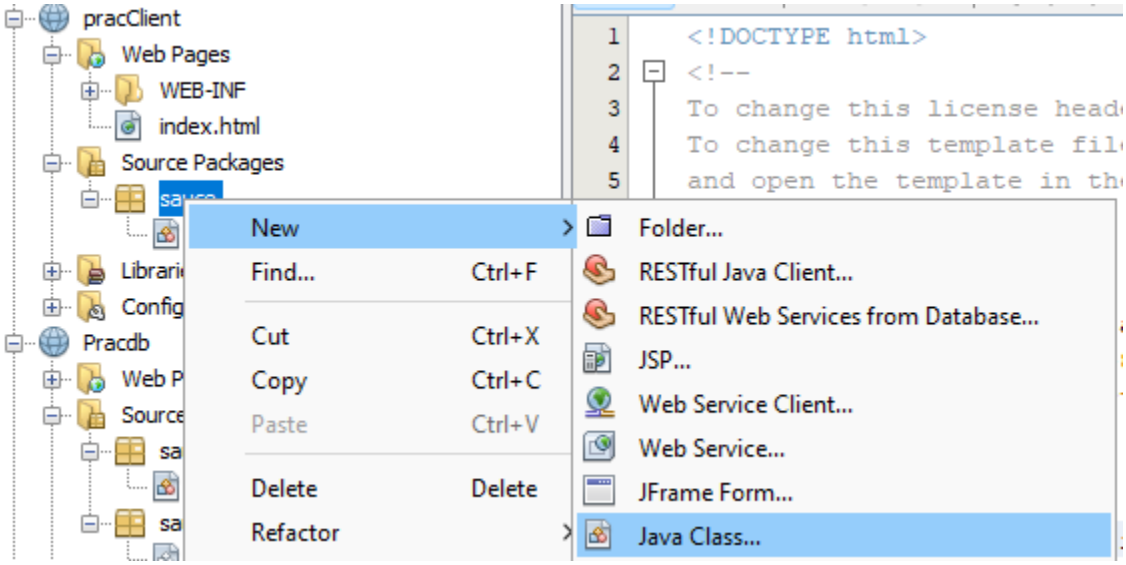


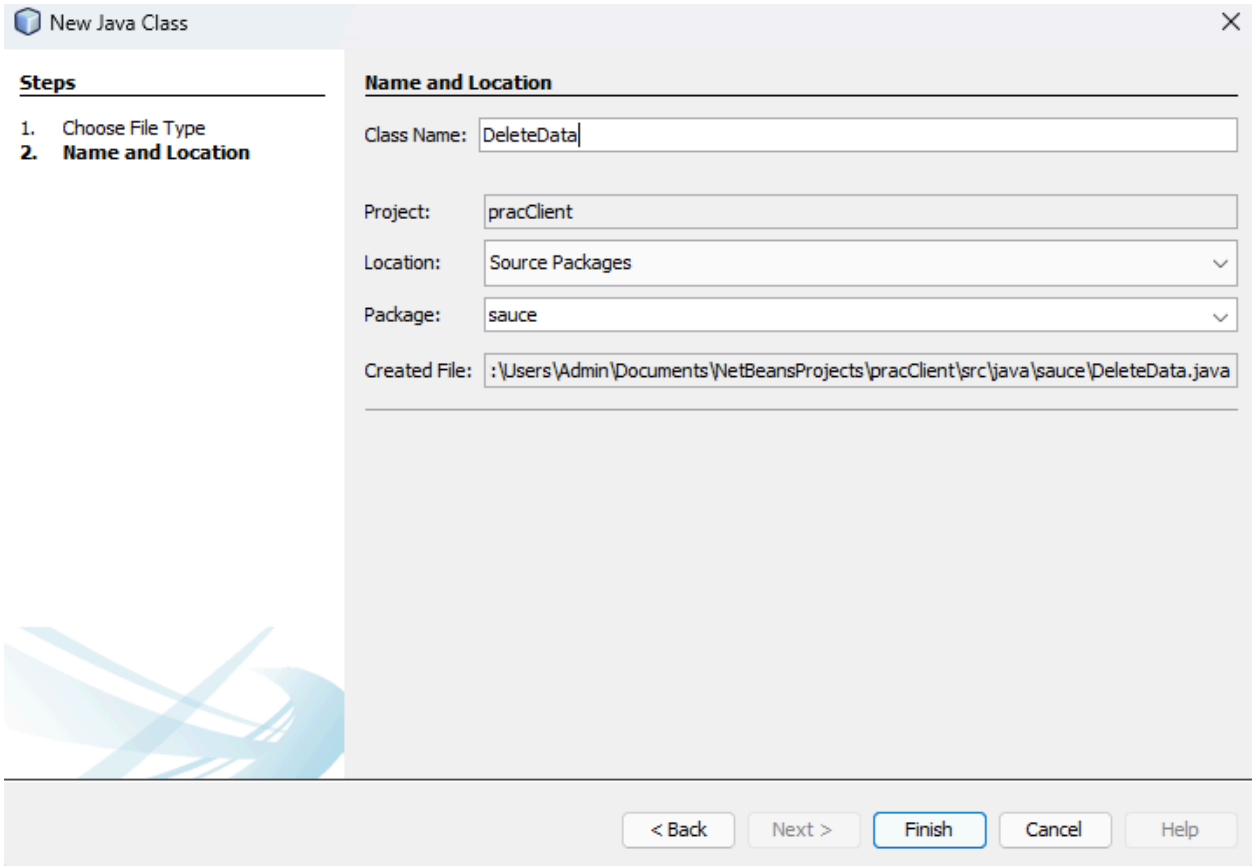
Deploy the pracdb project





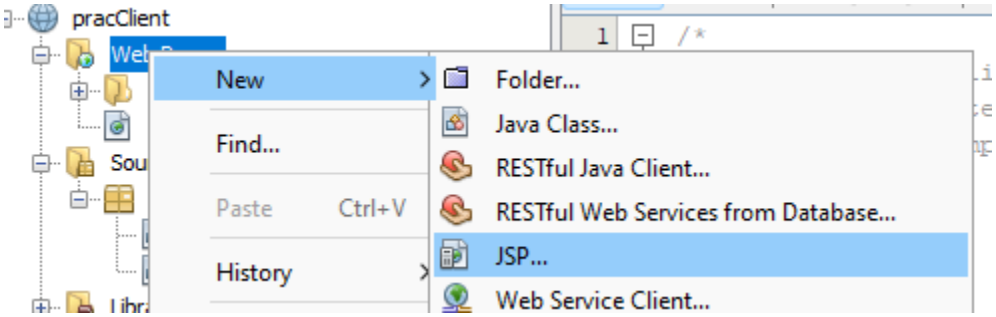

New RESTful Java Client
Steps
1. Choose File Type
2. Name and Location
Name and Location
Class Name: Data
Project: pracClient
Location: Source Packages
Package: sauce
Created File: C:\Users\Admin\Documents\NetBeansProjects\pracClient\src\java\sauce\Data.java
Select the REST resource:
From Project
IDE Registered
Browse...
REST Resource Name:
PracFacadeREST [sauce.prac]
Authentication: NONE
SSL
< Back
Next >
Finish
Cancel
Help

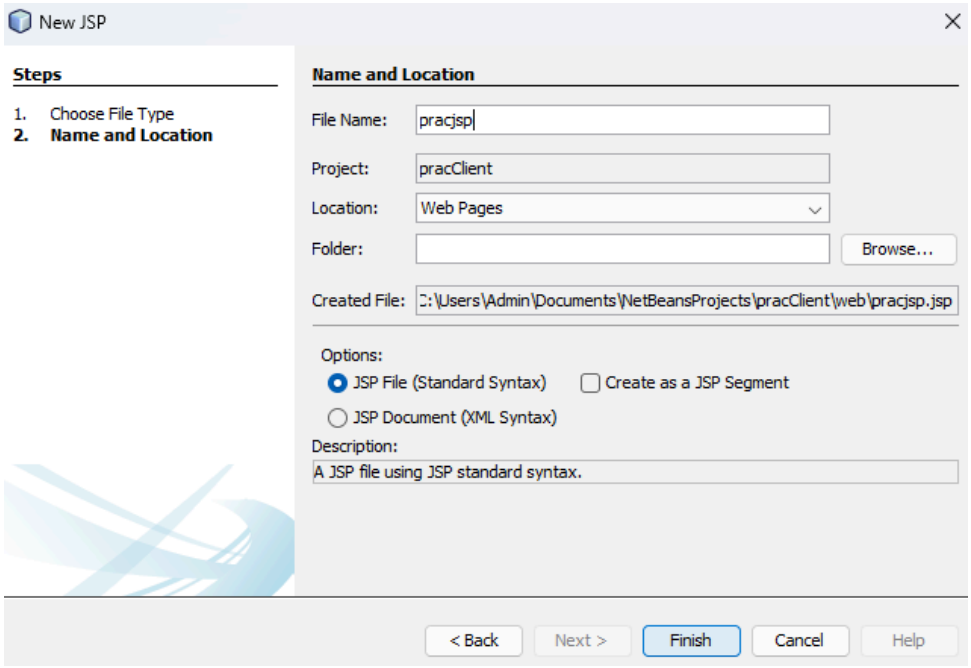

pracClient
Web Pages
WEB-INF
index.html
Source Packages
Pracdb
<!DOCTYPE html>
<!--
To change this license head
To change this template fil
and open the template in th
New
Find... Ctrl+F
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Refactor
Folder...
RESTful Java Client...
RESTful Web Services from Database...
JSP...
Web Service Client...
Web Service...
JFrame Form...
Java Class...



DeleteData.java

```
package sauce;

/**
 *
 * @author Admin
 */
public class DeleteData {
        public static void deldata(String id){
        String a = id;
        Data ob = new Data();
        ob.remove(a);
        System.out.println("Delete data");
        }
}
```





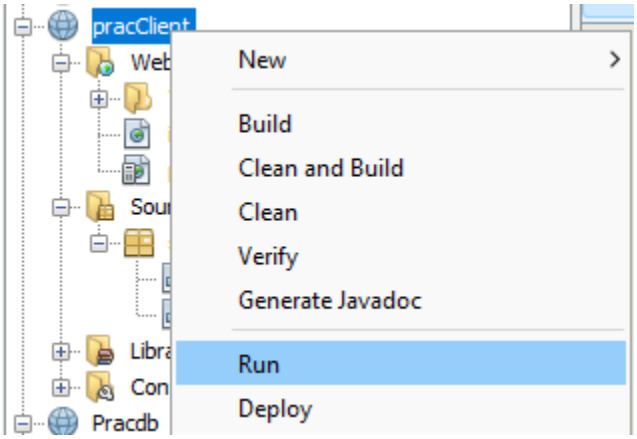
In pracjsp.jsp

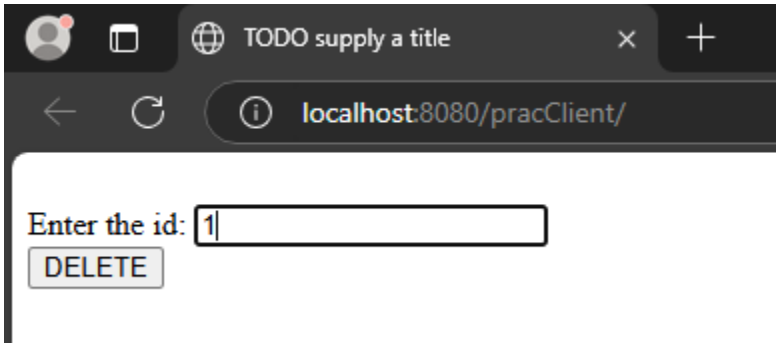
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        </head>
        <body>
        <%@page import = "sauce.DeleteData"%>
        <% String id = request.getParameter("id");
        DeleteData.deldata(id);
        %>
```

```
        </body>
</html>
```

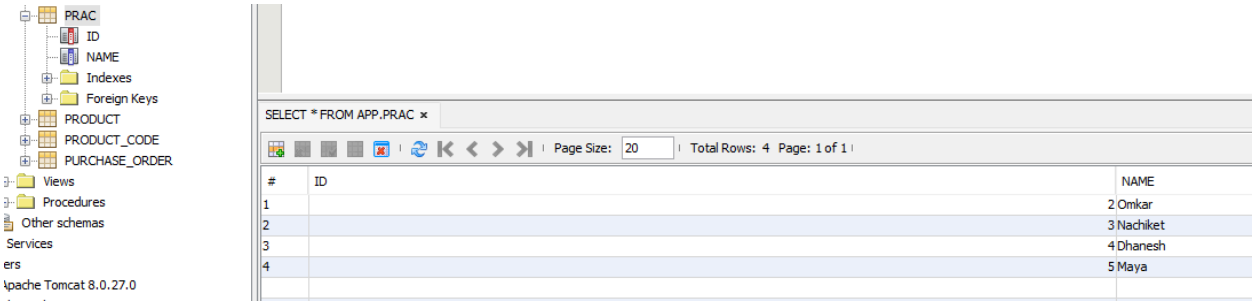
index.html

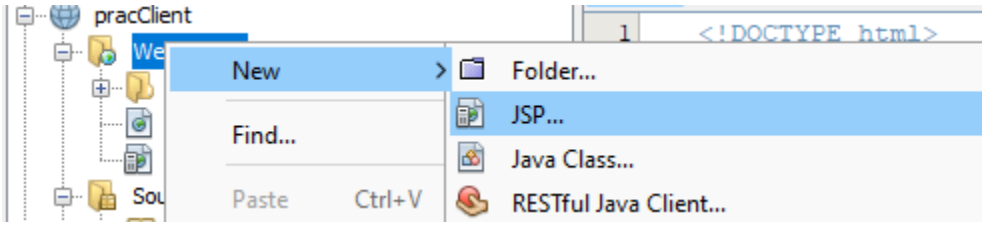
```
<html>
        <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        </head>
        <body>
        <form>
        <br> Enter the id: <input type="Text" name="id"><br>
        <input type ="submit" value ="DELETE" formaction="pracjsp.jsp">
        <h1>Request Response</h1>
        <input type="submit" formaction="getdata.jsp" value="GET DATA">
        </form>
        </body>
</html>
```

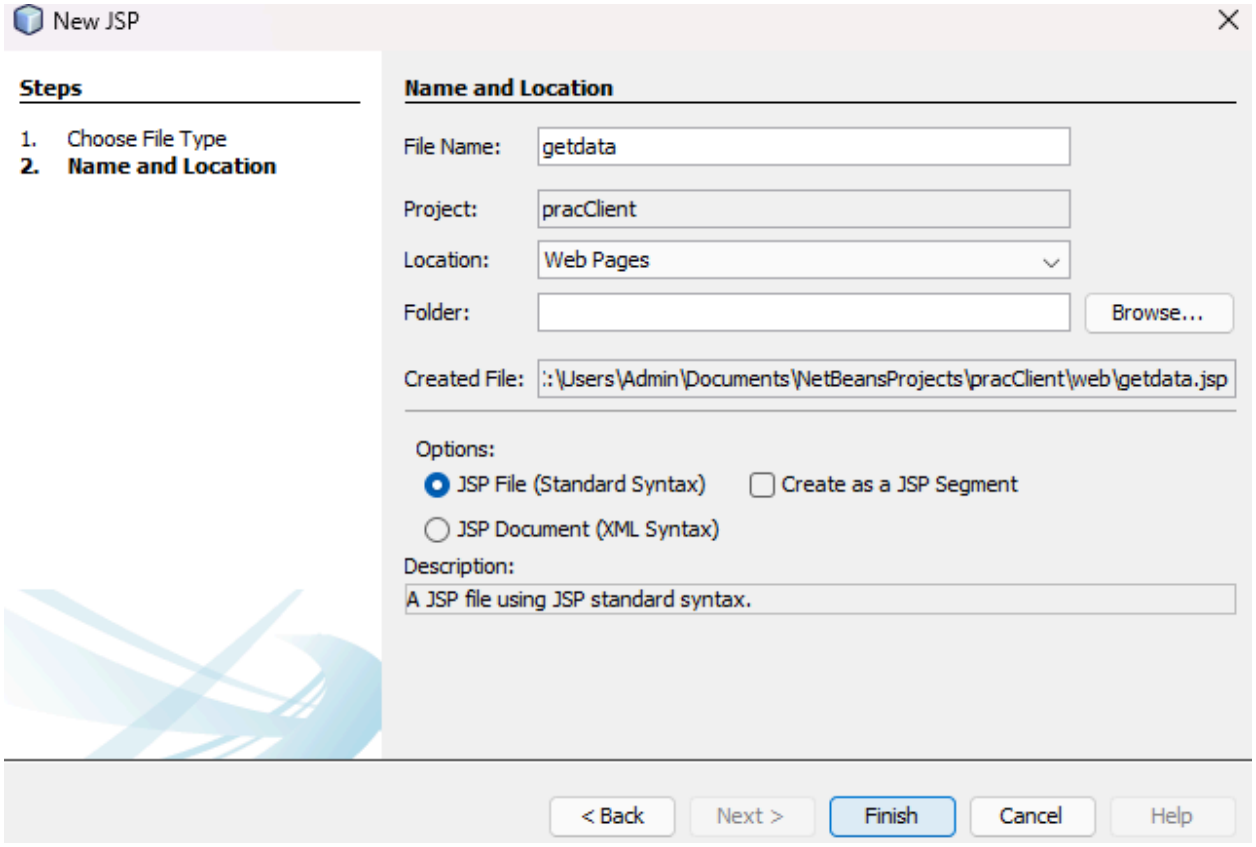




After clicking on delete button





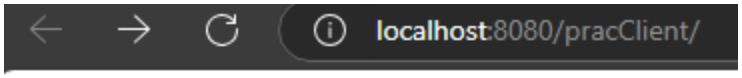


Getdata.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

```
<!DOCTYPE html>
<html>
	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<style>
	   table{font-family: arial, sans-serif;
	   border-collapse: collapse;
	   }
	   th,td{
	      border: 1px solid#000000;
	      text-align: center;
	      padding: 8px;
	   }
	</style>
	<script>
   var request = new XMLHttpRequest();
   request.open('GET', 'http://localhost:8080/Server/webresources/sahil.friends1/', true);
/*
Server=Projectname
sahil=servicename
tablename=friends1
*/
   request.onload = function () {
      var xml = this.responseXML;

      if (xml) {
         var friends = xml.getElementsByTagName("friends1");
         for (var i = 0; i < friends.length; i++) {
            var id = friends[i].getElementsByTagName("id")[0].textContent;
            var name = friends[i].getElementsByTagName("name")[0].textContent;

            var table = document.getElementById("mytable");
            var row = table.insertRow();
            var cell1 = row.insertCell(0);
            var cell2 = row.insertCell(1);
            cell1.innerHTML = id;
            cell2.innerHTML = name;
         }
      } else {
         console.error("XML parsing failed.");
      }
   };
   request.send();
</script>
```

```
    </head>
    <body>
        <table id="mytable">
            <tr>
                <th>ID</th>
                <th>Name</th>
            </tr>
        </table>
    </body>
</html>
```



Enter the id: 2

DELETE

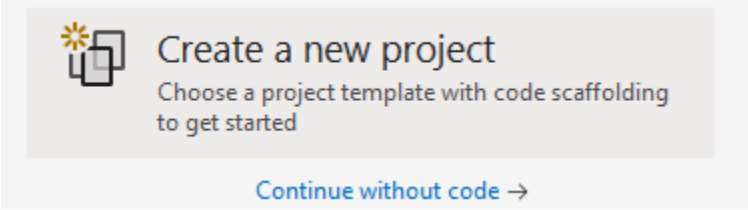
Request Response

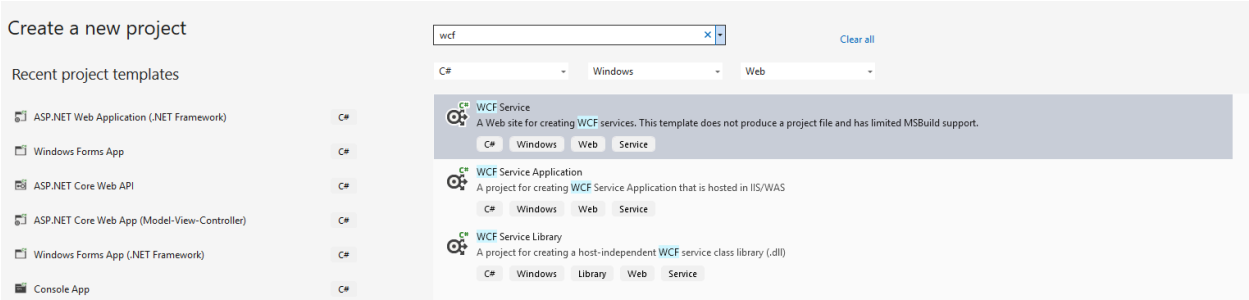
GET DATA

ID	Name

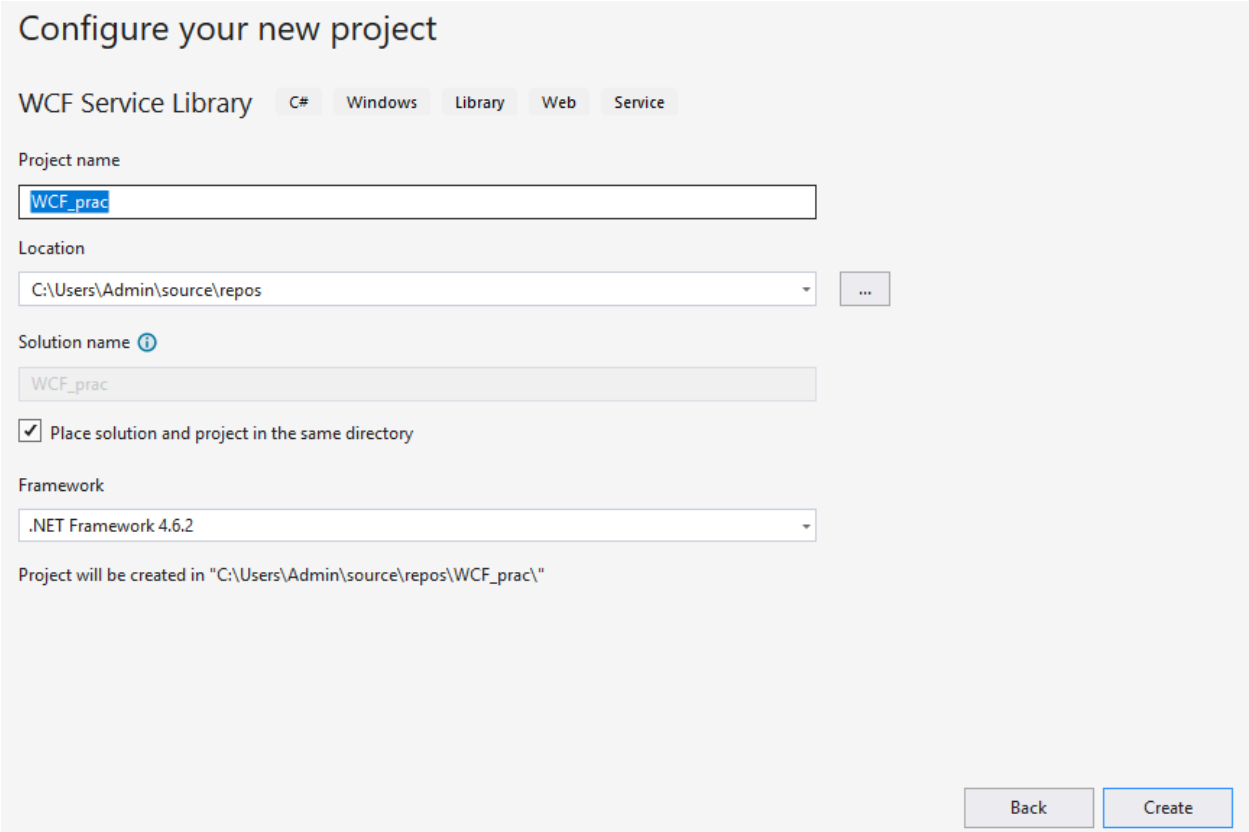
Practical 3

In VIsual Studio, create a new project and search for wcf services





Select wcf service library

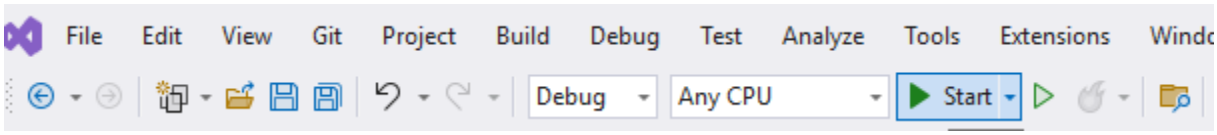


Service1.cs

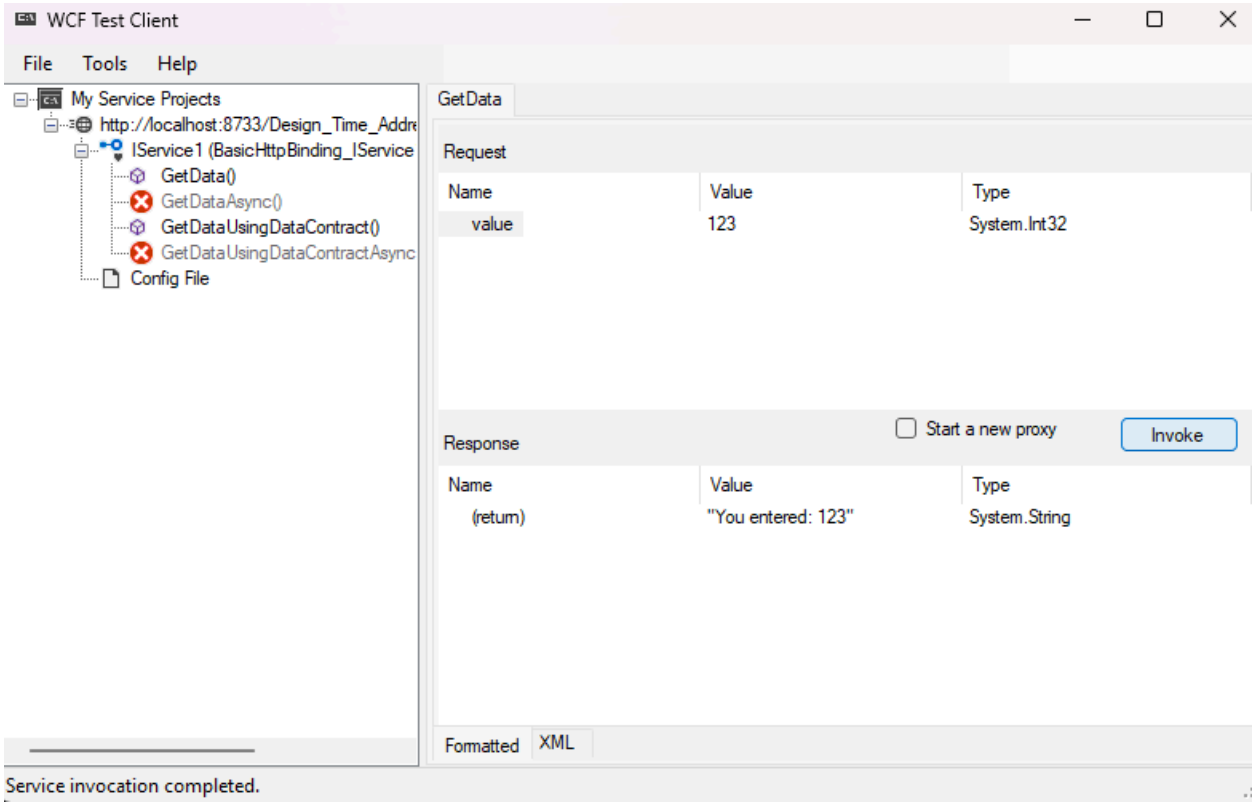
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace WCF_prac
{
        // NOTE: You can use the "Rename" command on the "Refactor" menu to change the
class name "Service1" in both code and config file together.
        public class Service1 : IService1
        {
        public long sqrno(int num) { return num * num; }
        public long cubeno(long num) { return num * num * num; }
        public string GetData(int value)
        {
        return String.Format("You entered: {0}", value);
```

```
        }
        public CompositeType GetDataUsingDataContract(CompositeType composite)
        {
        if (composite == null)
                throw new ArgumentNullException("Composite");
        if (composite.BoolValue)
                composite.StringValue += "Suffix";
        return composite;
        }
        }
}
```

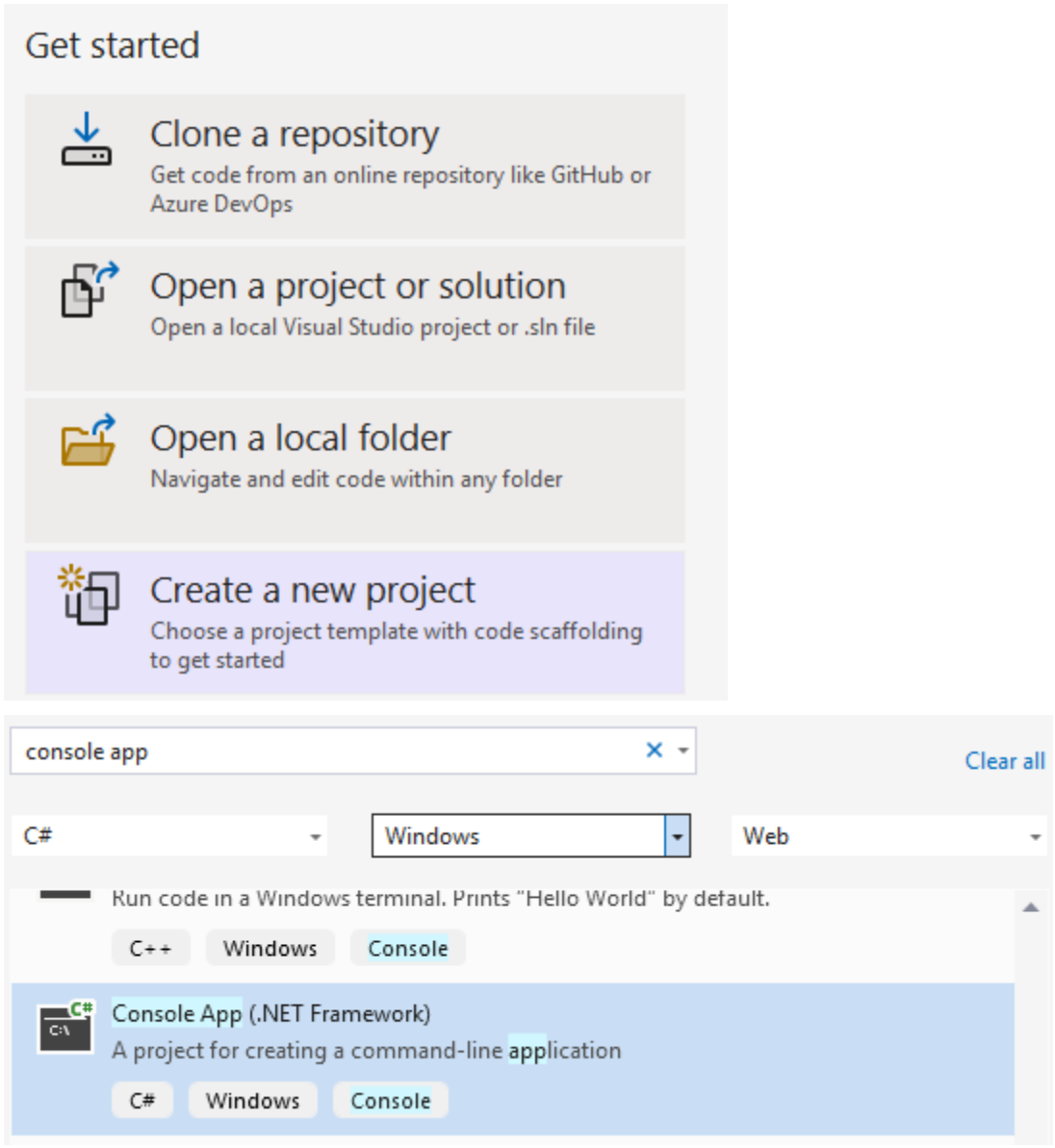
Click on start

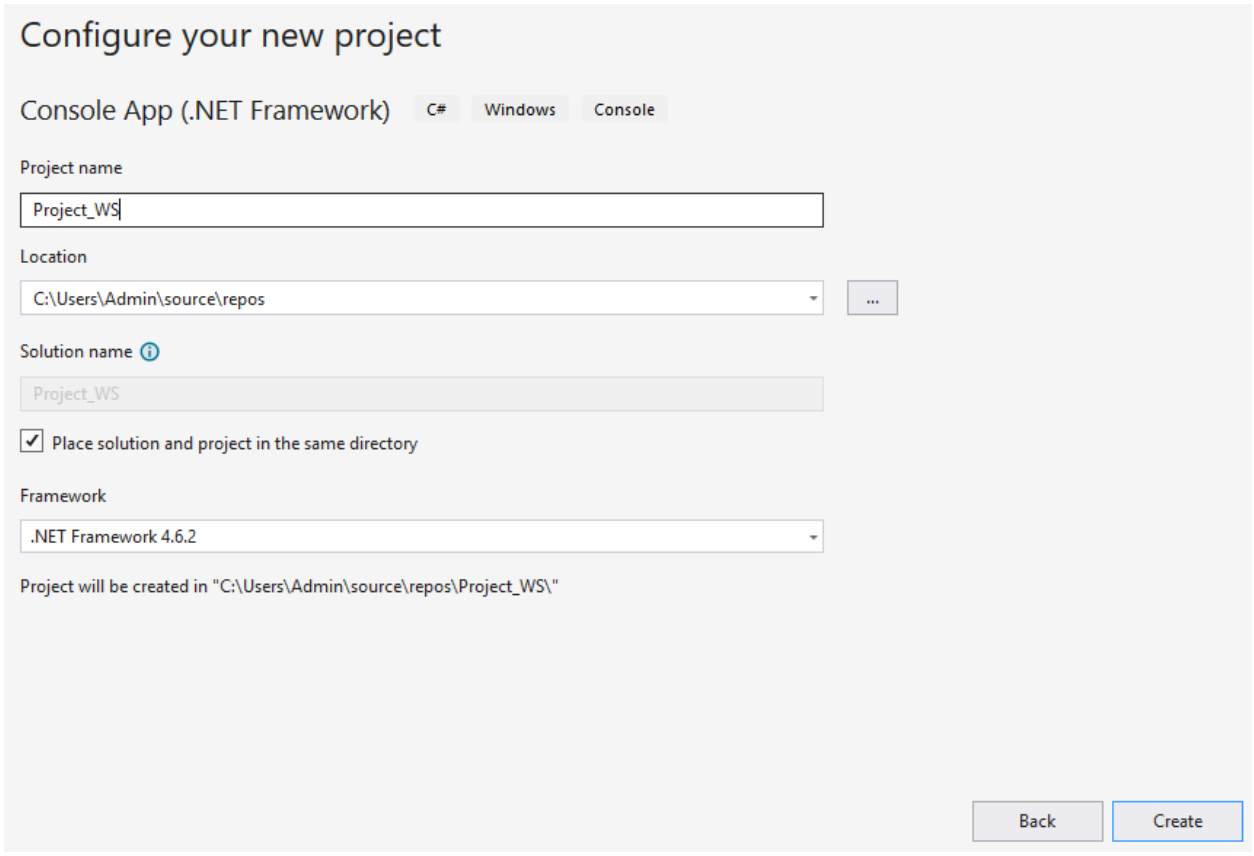


Double click on the function [GetData() in this case], change value, and click on invoke

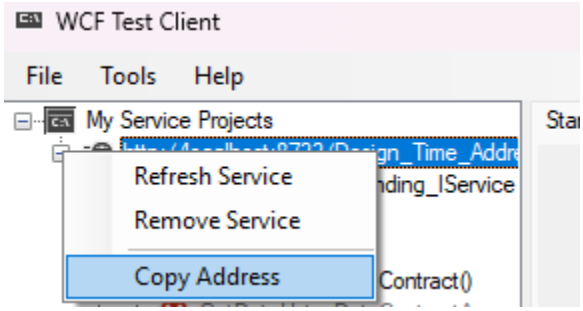


Create a new project

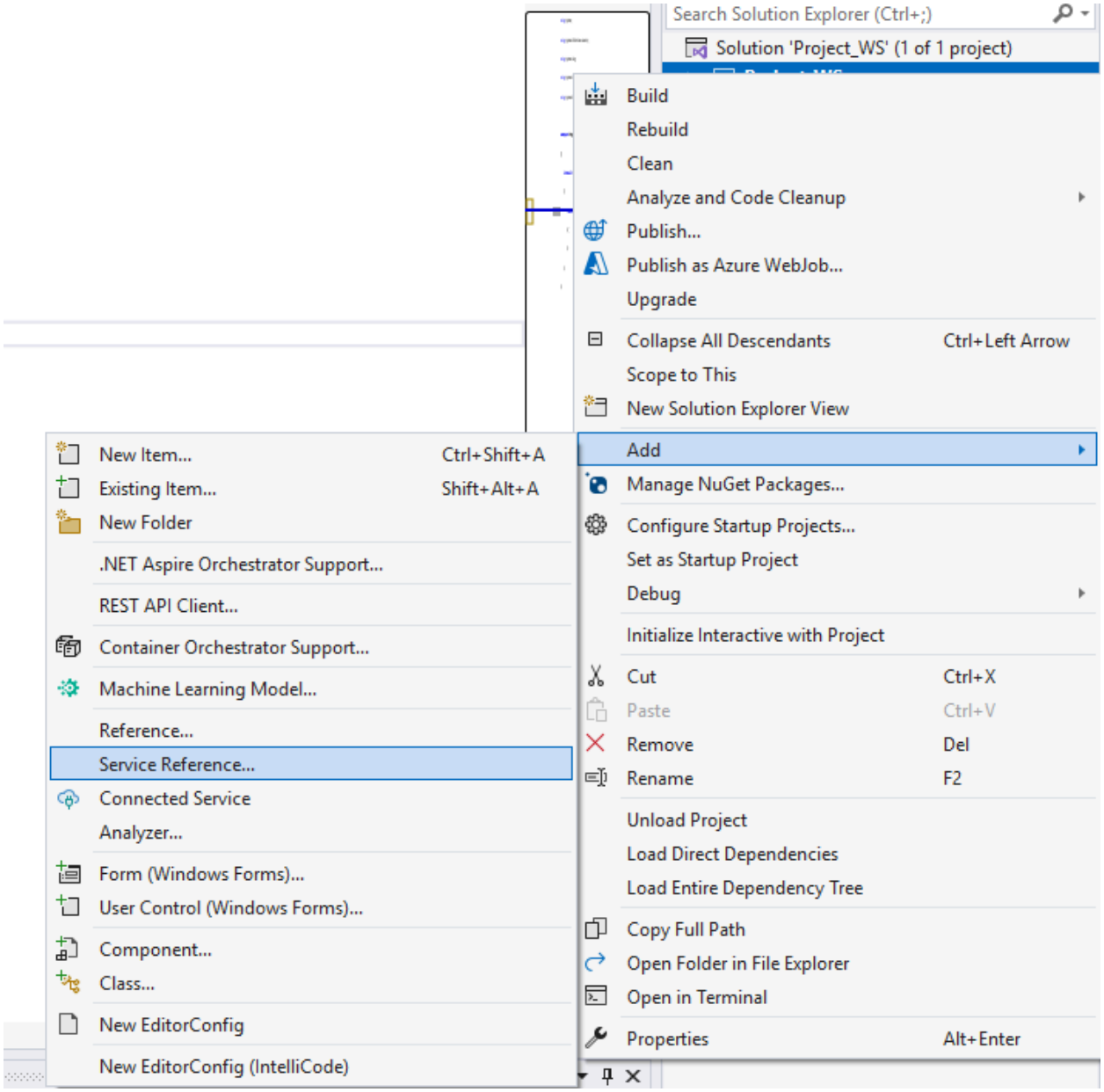

Get started
Clone a repository
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
Create a new project
Choose a project template with code scaffolding to get started
console app
Clear all
C#
Windows
Web
Run code in a Windows terminal. Prints "Hello World" by default.
C++
Windows
Console
Console App (.NET Framework)
A project for creating a command-line application
C#
Windows
Console



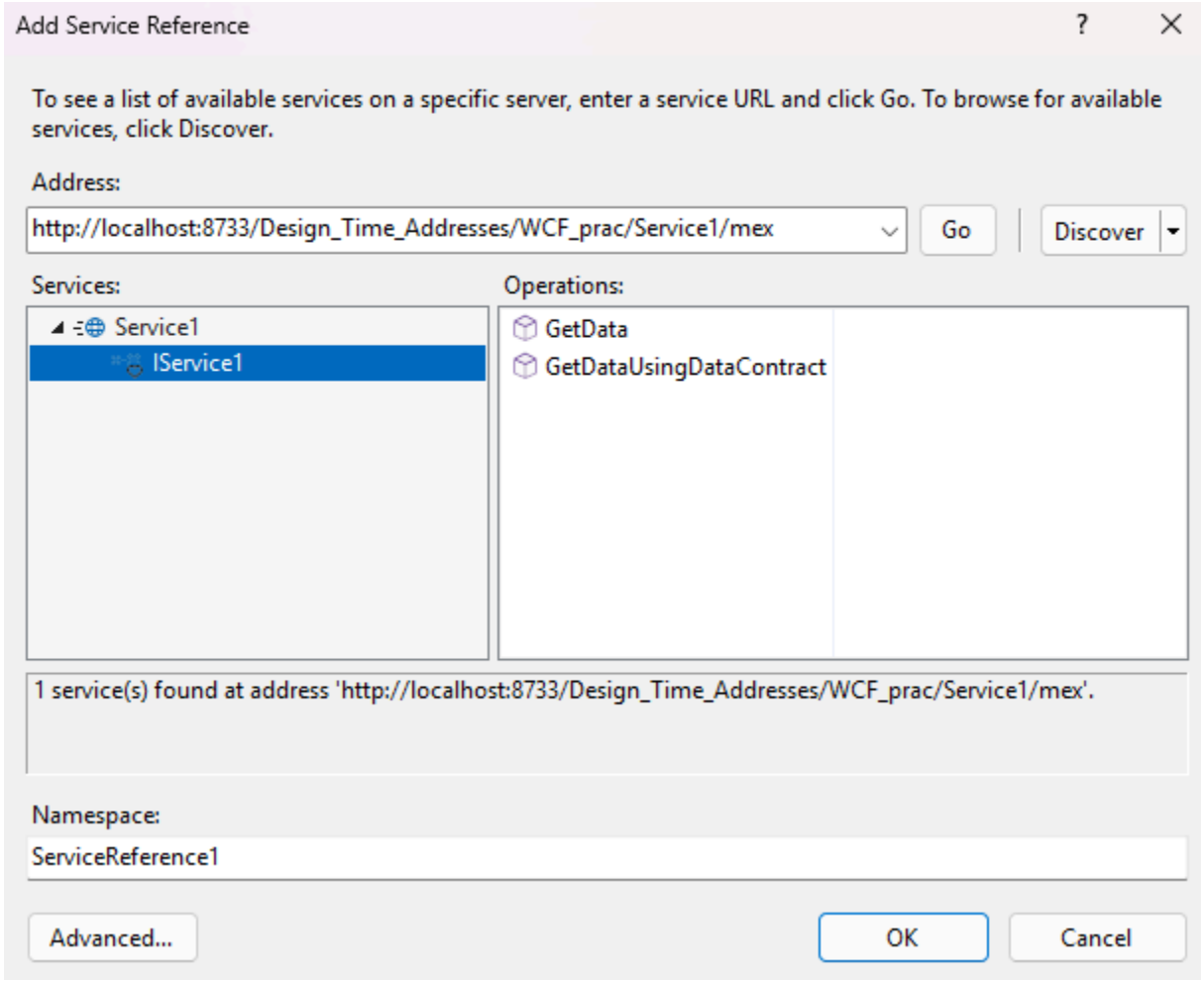
Copy the address by running the previous



Right click on project and add service reference

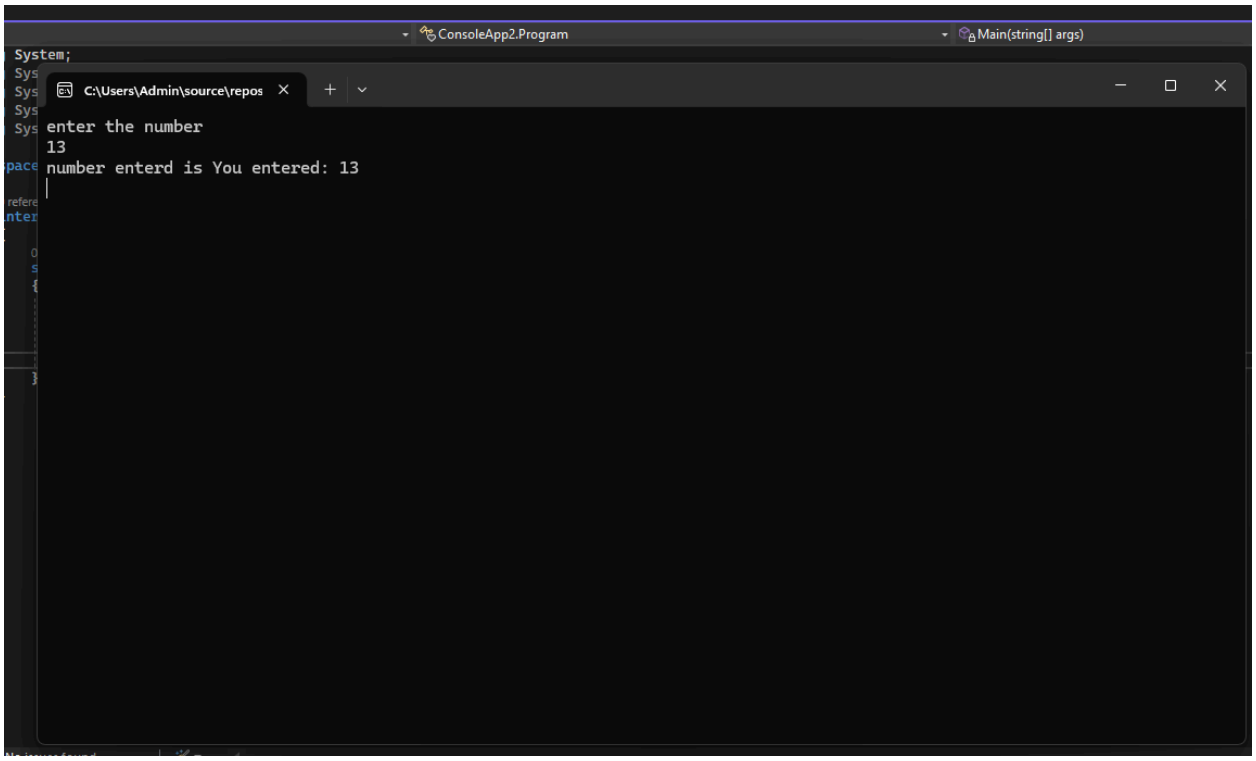

Search Solution Explorer (Ctrl+;)
Solution 'Project_WS' (1 of 1 project)
Build
Rebuild
Clean
Analyze and Code Cleanup
Publish...
Publish as Azure WebJob...
Upgrade
Collapse All Descendants Ctrl+Left Arrow
Scope to This
New Solution Explorer View
Add
Manage NuGet Packages...
Configure Startup Projects...
Set as Startup Project
Debug
Initialize Interactive with Project
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename F2
Unload Project
Load Direct Dependencies
Load Entire Dependency Tree
Copy Full Path
Open Folder in File Explorer
Open in Terminal
Properties Alt+Enter
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Folder
.NET Aspire Orchestrator Support...
REST API Client...
Container Orchestrator Support...
Machine Learning Model...
Reference...
Service Reference...
Connected Service
Analyzer...
Form (Windows Forms)...
User Control (Windows Forms)...
Component...
Class...
New EditorConfig
New EditorConfig (IntelliCode)

Click on Ok




```
using System;

namespace Project_WS
{
	internal class Program
	{
	static void Main(string[] args)
	{
	ServiceReference1.Service1Client obj = new ServiceReference1.Service1Client();
	Console.WriteLine("Enter the number");
	int st = Convert.ToInt32(Console.ReadLine());
	Console.WriteLine("The entered number is: " + obj.GetData(st));
	Console.ReadLine();
	}
	}
}
```

Click on Run
Output:



Practical 4

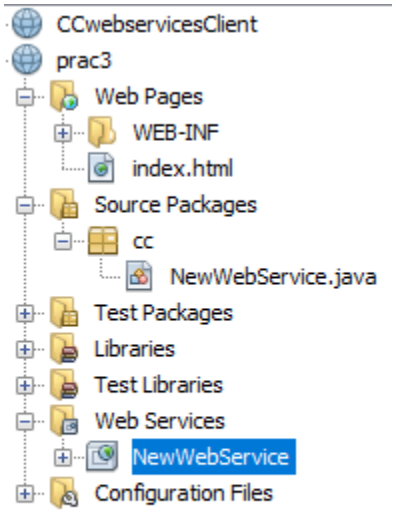
Create an new webservice named as new web services

NewWebService.java

History

```
 change this license header, choose License Headers in Project Properties.
 change this template file, choose Tools | Templates
d open the template in the editor.

ge cc;

t javax.jws.WebService;
t javax.jws.WebMethod;
t javax.jws.WebParam;



uthor Admin

ervice(serviceName = "NewWebService")
c class NewWebService {

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

Test your web services created



```
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.xml.ws.RequestWrapper;
import javax.xml.ws.ResponseWrapper;

/**
 *
 * @author Admin
 */
@WebService(serviceName = "NewWebService")
public class NewWebService {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "DolllarTORup")
    public String DollarToRup(@WebParam(name = "ru") double ru) {
        return "the enter dollar is ="+ru + "is in rupess is :" + (ru*90);
    }
    @WebMethod(operationName = "rupesstodollar")
    public String rupesstodollar(@WebParam(name = "a") double a) {
        return "the enter dollar is ="+a + "is in rupess is :" + (a/90);
    }
```

```
@WebMethod(operationName = "DolllarTORup")
        public String DollarToRup(@WebParam(name = "ru") double ru) {
        return "the enter dollar is ="+ru + "is in rupess is :" + (ru*90);
        }
        @WebMethod(operationName = "rupesstodollar")
        public String rupesstodollar(@WebParam(name = "a") double a) {
        return "the enter dollar is ="+a + "is in rupess is :" + (a/90);
        }
```

NewWebService Web Service Tester

his form will allow you to test your web service implementation (WSDL File)

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

Methods :

ublic abstract java.lang.String cc.NewWebService.rupesstodollar(double)
rupesstodollar (5)

ublic abstract java.lang.String cc.NewWebService.dolllarTORup1(java.lang.String)
dolllarTORup1 ( )

ublic abstract java.lang.String cc.NewWebService.dolllarTORup(double)
dolllarTORup ( )

rupesstodollar Method invocation

Method parameter(s)

Type	Value
double	5

Method returned

java.lang.String : "**the enter dollar is =5.0is in rupess is :0.05555555555555555**"

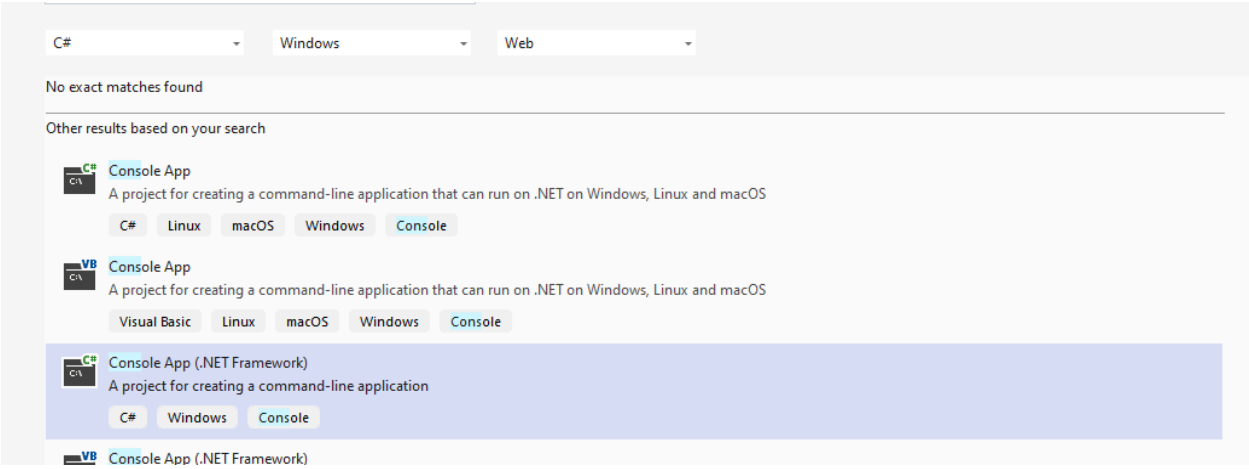
SOAP Request

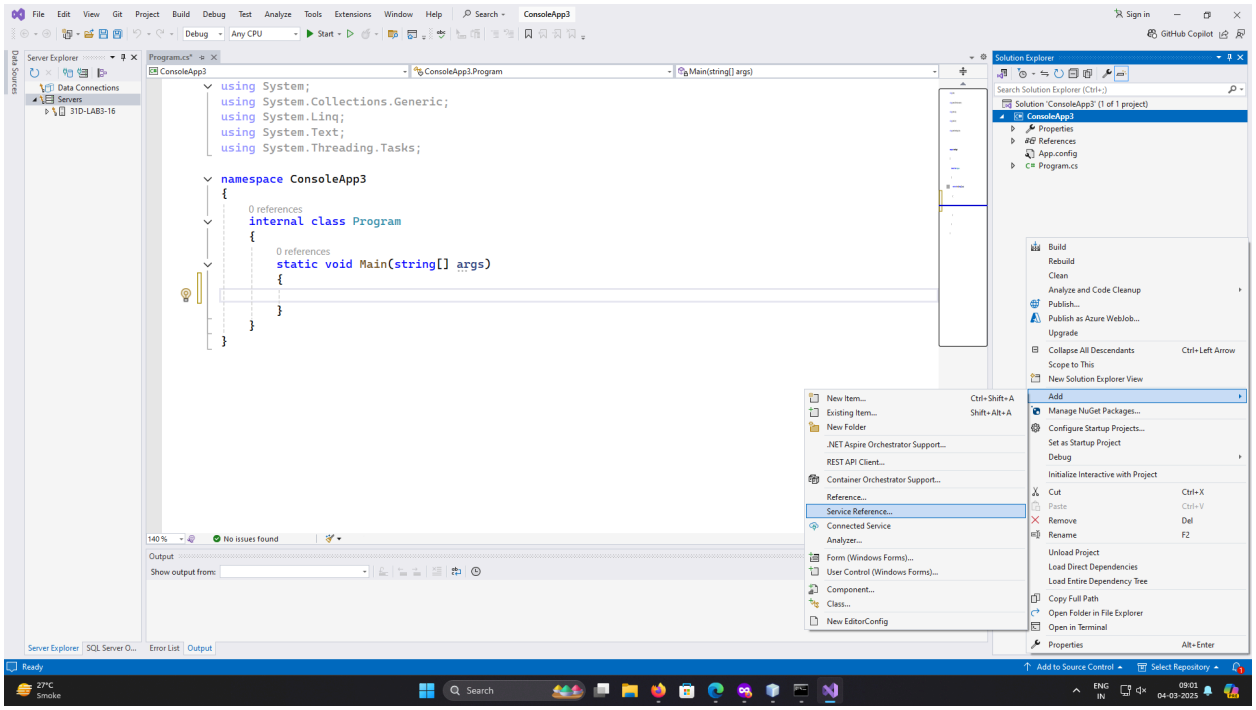
```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoa
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:rupesstodollar xmlns:ns2="http://cc/">
            <a>5.0</a>
        </ns2:rupesstodollar>
    </S:Body>
</S:Envelope>
```

SOAP Response

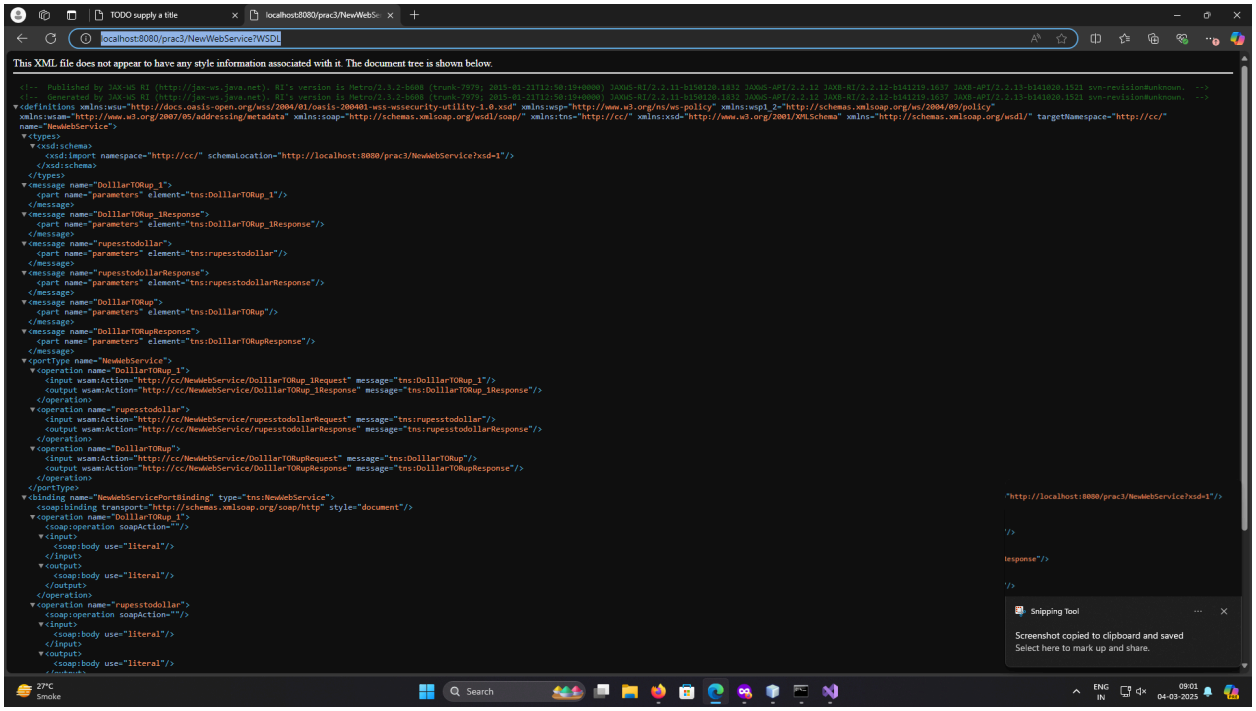
```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoa
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:rupesstodollarResponse xmlns:ns2="http://cc/">
            <return>the enter dollar is =5.0is in rupess is :0.05555555555555555</return>
        </ns2:rupesstodollarResponse>
    </S:Body>
</S:Envelope>
```

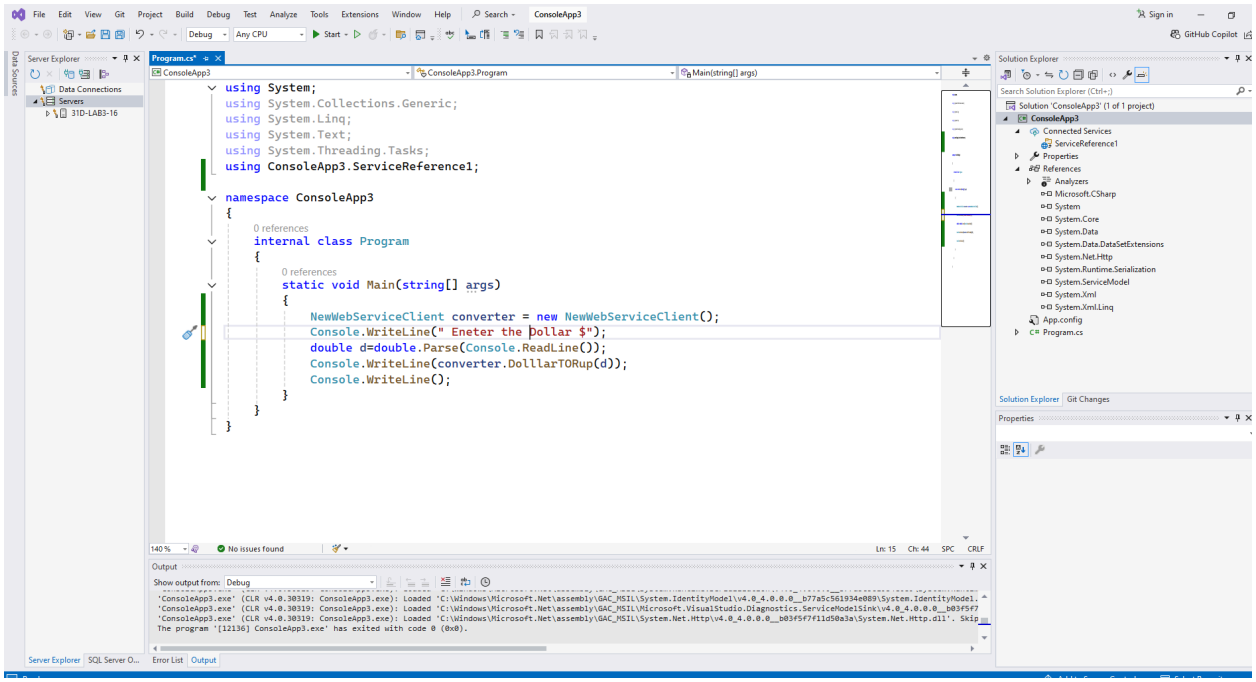
Go to visual copd create na .net framework click on add and then references services copy paste the link of the wsdl file





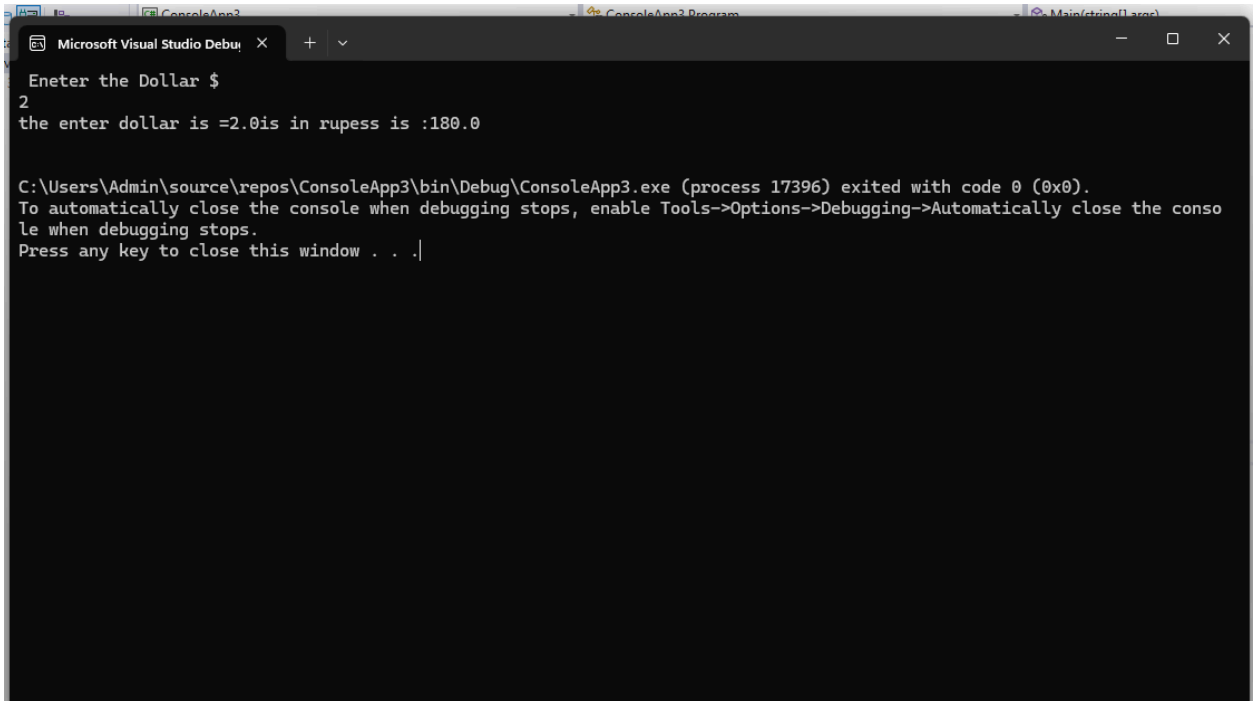
Copy the wsdl link and paste in your visual studio project.



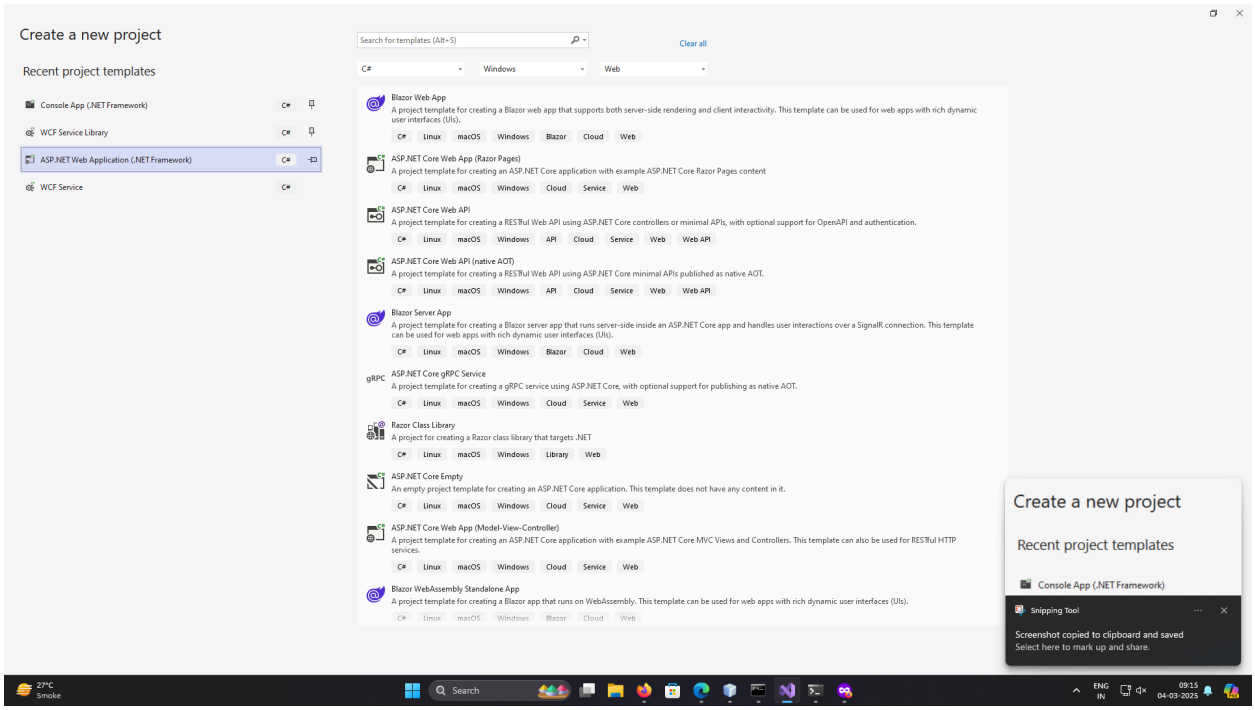


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ConsoleApp3.ServiceReference1;

namespace ConsoleApp3
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            NewWebServiceClient converter = new NewWebServiceClient();
            Console.WriteLine(" Eneter the Dollar $");
            double d=double.Parse(Console.ReadLine());
            Console.WriteLine(converter.DolllarTORup(d));
            Console.WriteLine();
        }
    }
}
```

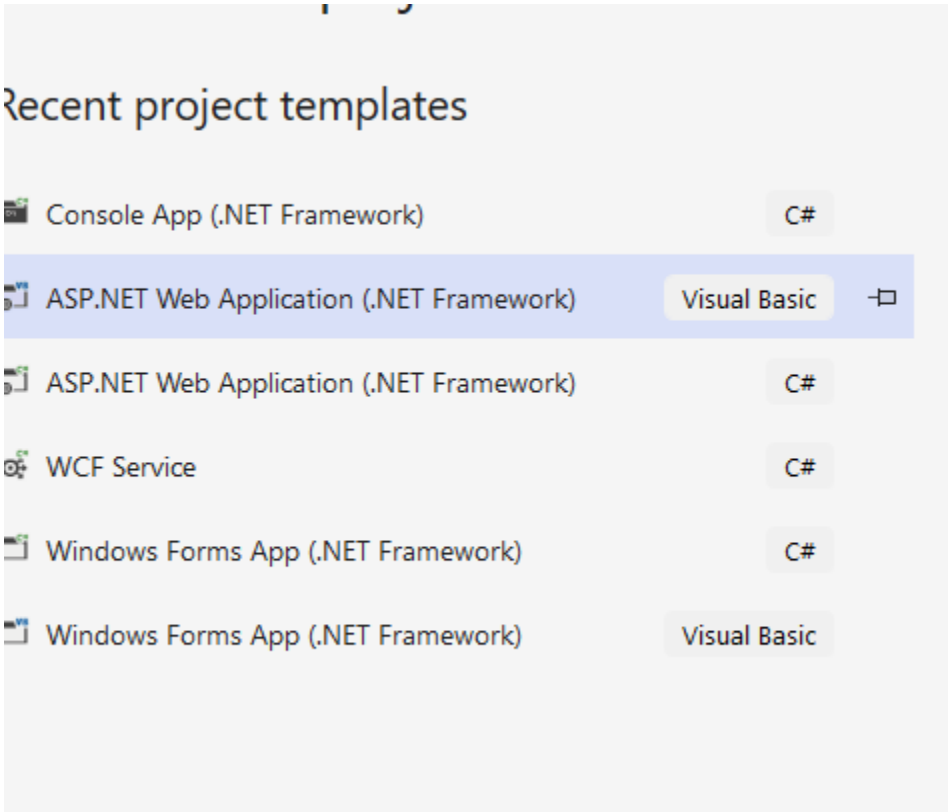
Output :-

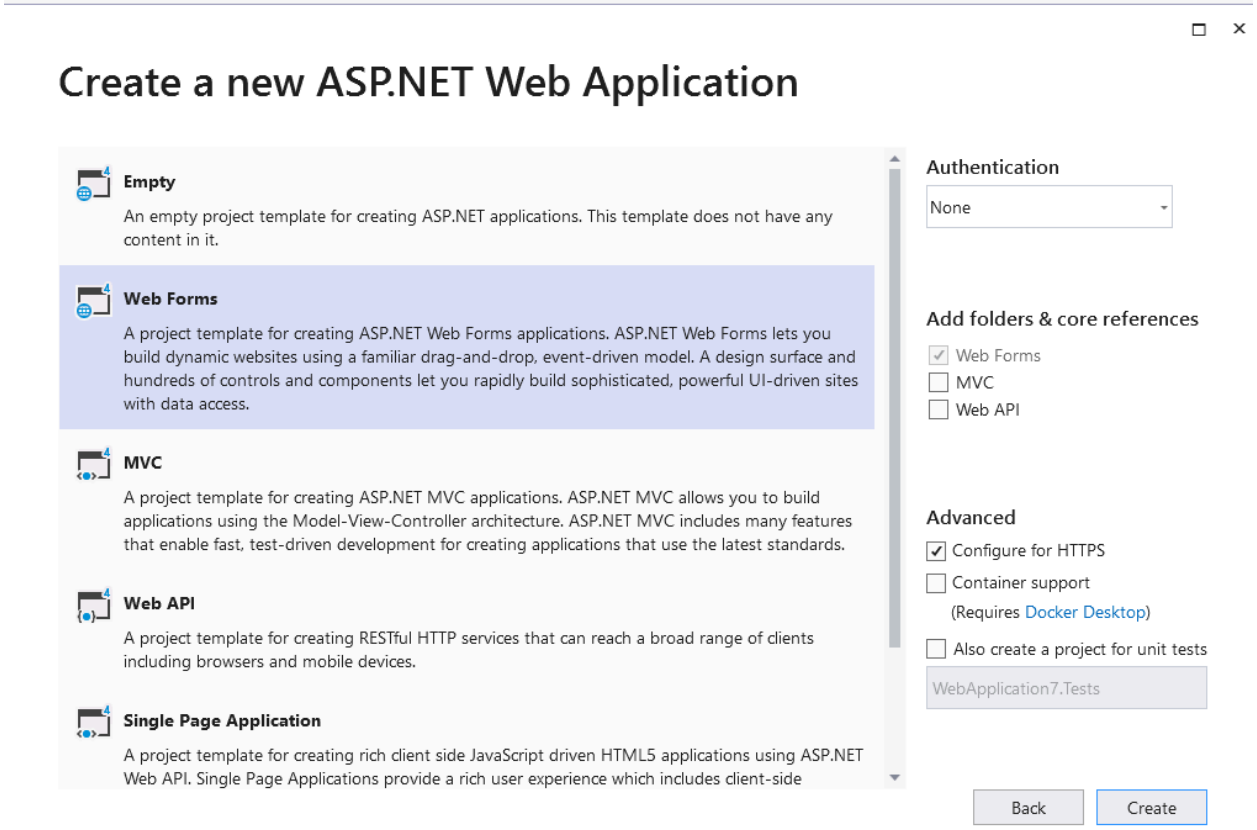


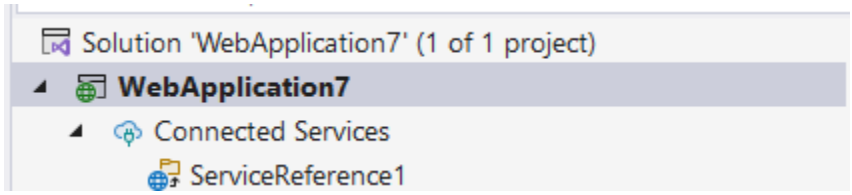
```
Microsoft Visual Studio Debu

 Eneter the Dollar $
2
the enter dollar is =2.0is in rupess is :180.0


C:\Users\Admin\source\repos\ConsoleApp3\bin\Debug\ConsoleApp3.exe (process 17396) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```


Create a new project
Recent project templates
Console App (.NET Framework)
WCF Service Library
ASP.NET Web Application (.NET Framework)
WCF Service
Search for templates (Alt+S)
Clear all
C#
Windows
Web
Blazor Web App
A project template for creating a Blazor web app that supports both server-side rendering and client interactivity. This template can be used for web apps with rich dynamic user interfaces (UIs).
ASP.NET Core Web App (Razor Pages)
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content
ASP.NET Core Web API
A project template for creating a RESTful Web API using ASP.NET Core controllers or minimal APIs, with optional support for OpenAPI and authentication.
ASP.NET Core Web API (native AOT)
A project template for creating a RESTful Web API using ASP.NET Core minimal APIs published as native AOT.
Blazor Server App
A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).
ASP.NET Core gRPC Service
A project template for creating a gRPC service using ASP.NET Core, with optional support for publishing as native AOT.
Razor Class Library
A project for creating a Razor class library that targets .NET
ASP.NET Core Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
Blazor WebAssembly Standalone App
A project template for creating a Blazor app that runs on WebAssembly. This template can be used for web apps with rich dynamic user interfaces (UIs).
Create a new project
Recent project templates
Console App (.NET Framework)
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.
27°C
Smoke
Search
ENG
IN
09:15
04-03-2025

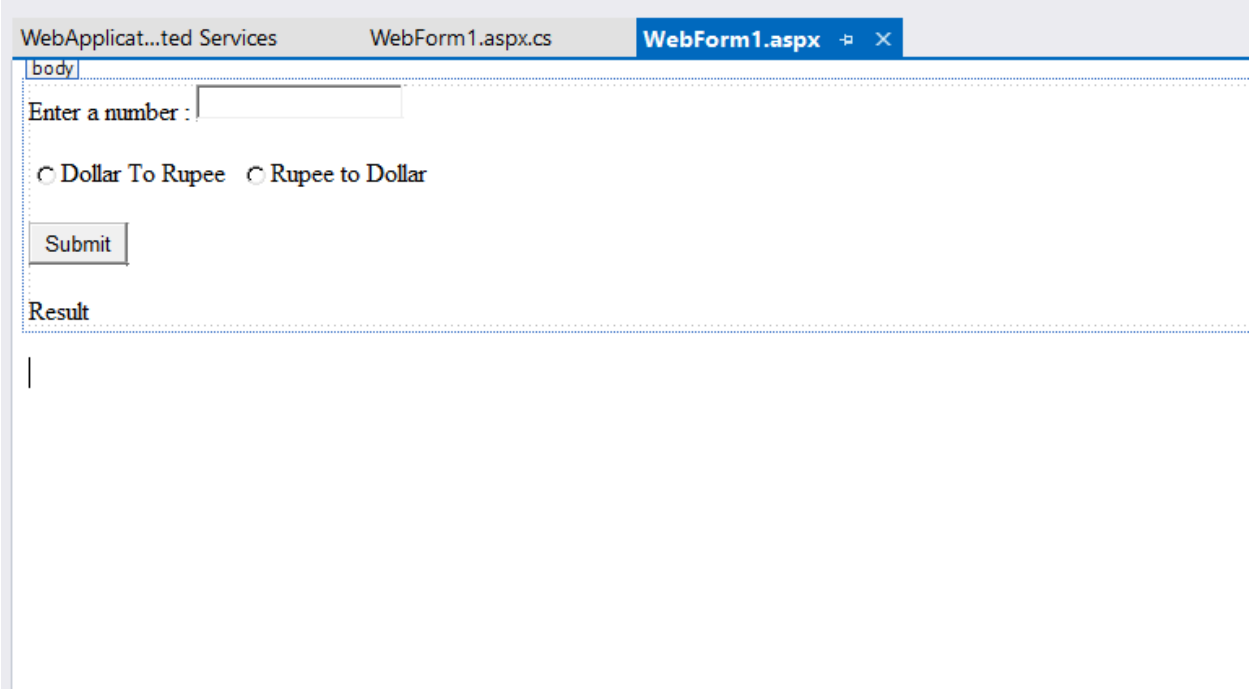
Form :

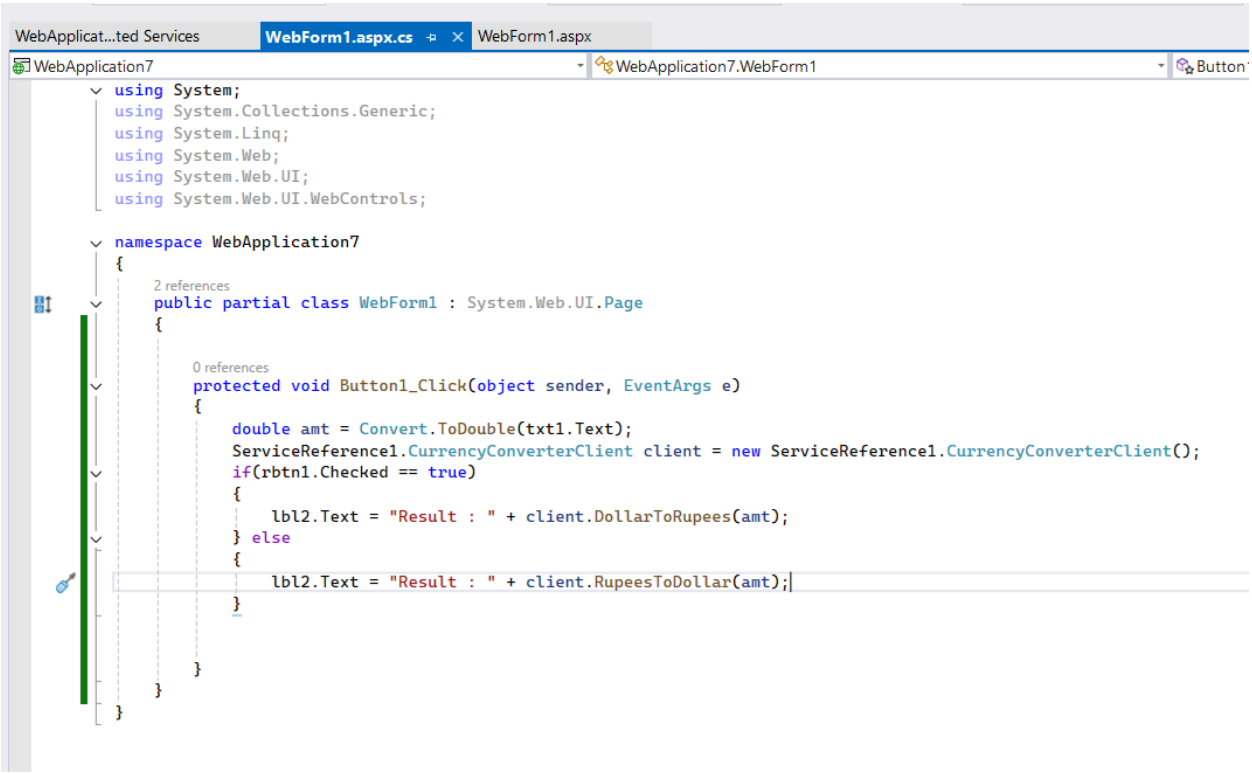

Recent project templates
Console App (.NET Framework)
C#
ASP.NET Web Application (.NET Framework)
Visual Basic
ASP.NET Web Application (.NET Framework)
C#
WCF Service
C#
Windows Forms App (.NET Framework)
C#
Windows Forms App (.NET Framework)
Visual Basic



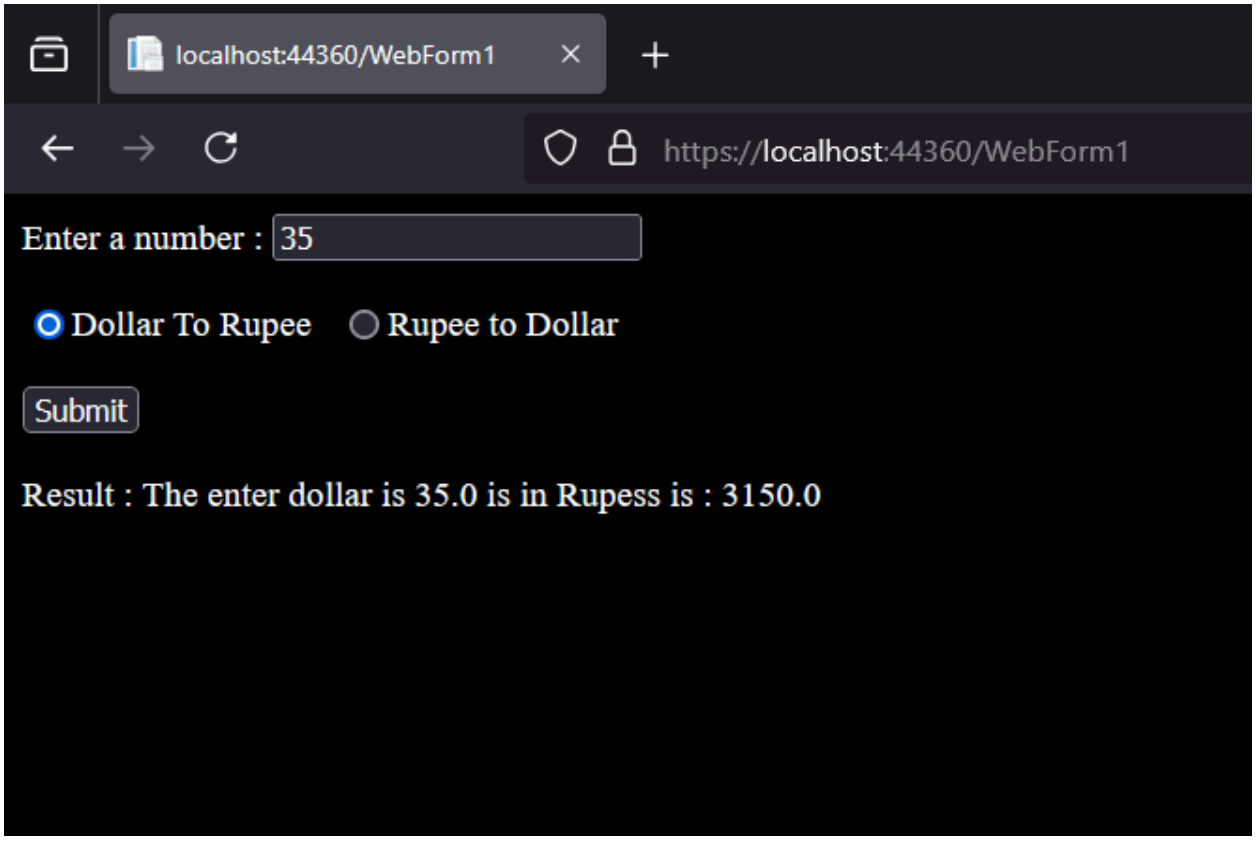


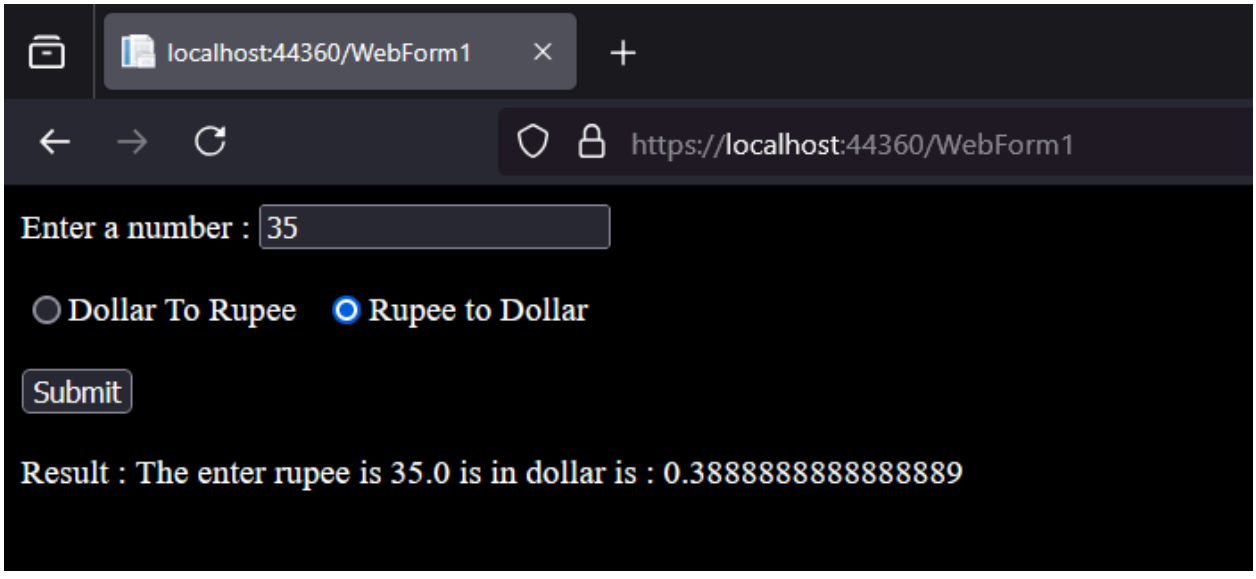
(Add service reference using wsdl as done previously)





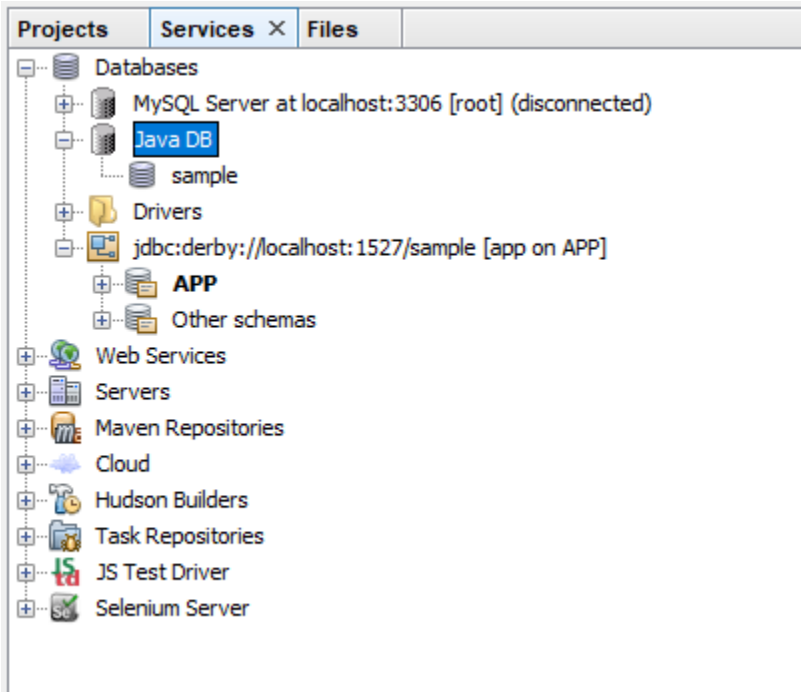
Add a new web form :

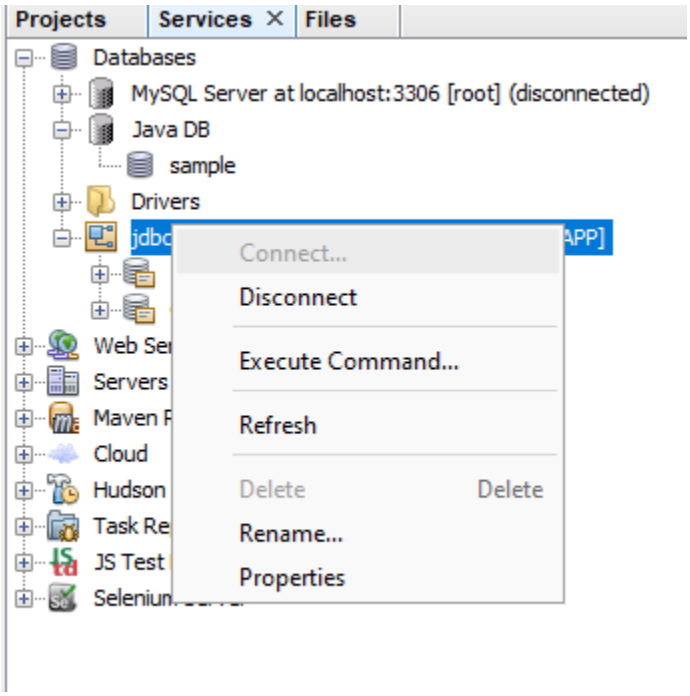

localhost:44360/WebForm1
https://localhost:44360/WebForm1
Enter a number : 35
Dollar To Rupee
Rupee to Dollar
Submit
Result : The enter dollar is 35.0 is in Rupess is : 3150.0


localhost:44360/WebForm1
https://localhost:44360/WebForm1
Enter a number : 35
Dollar To Rupee
Rupee to Dollar
Submit
Result : The enter rupee is 35.0 is in dollar is : 0.3888888888888889

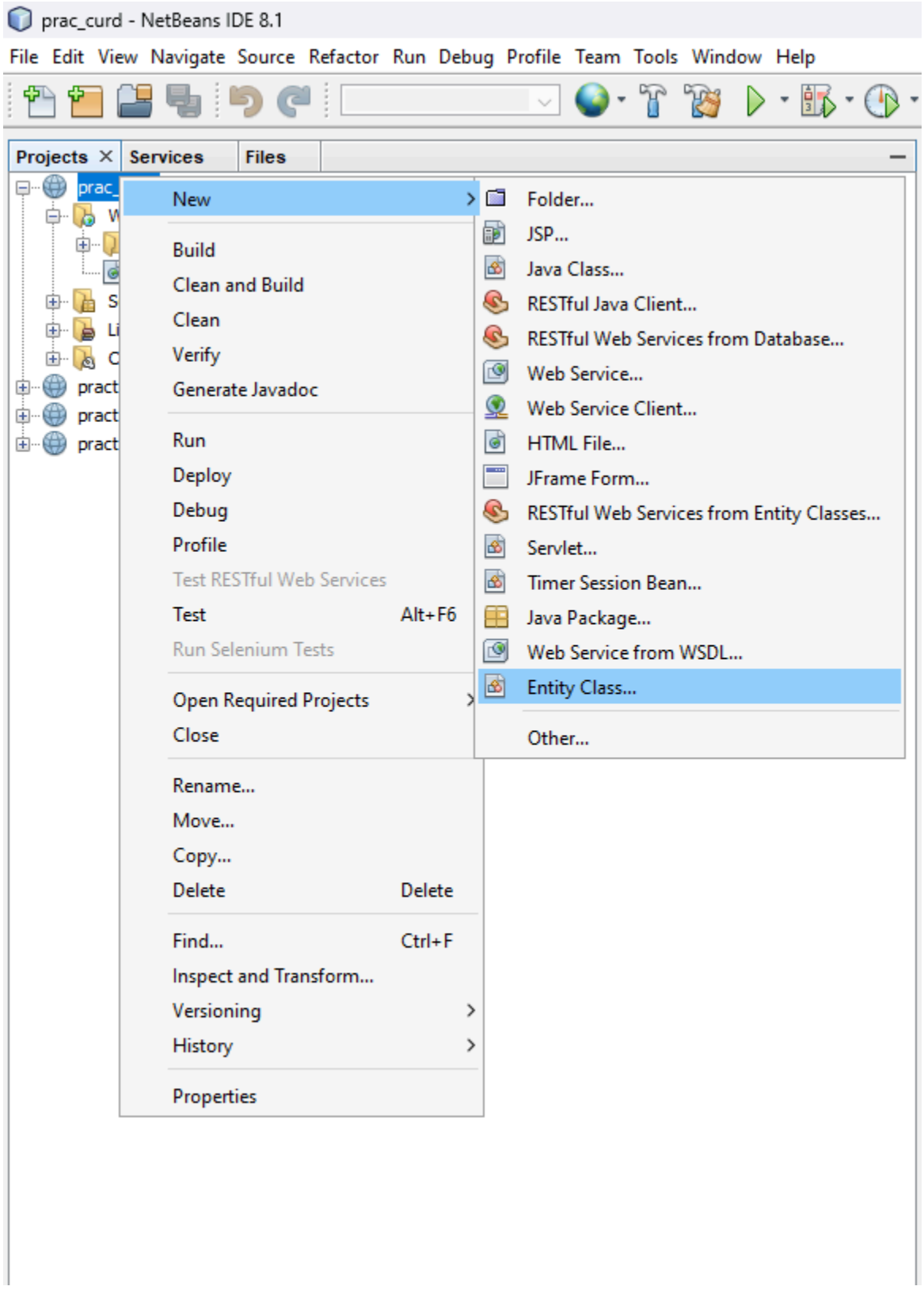
Practical 5

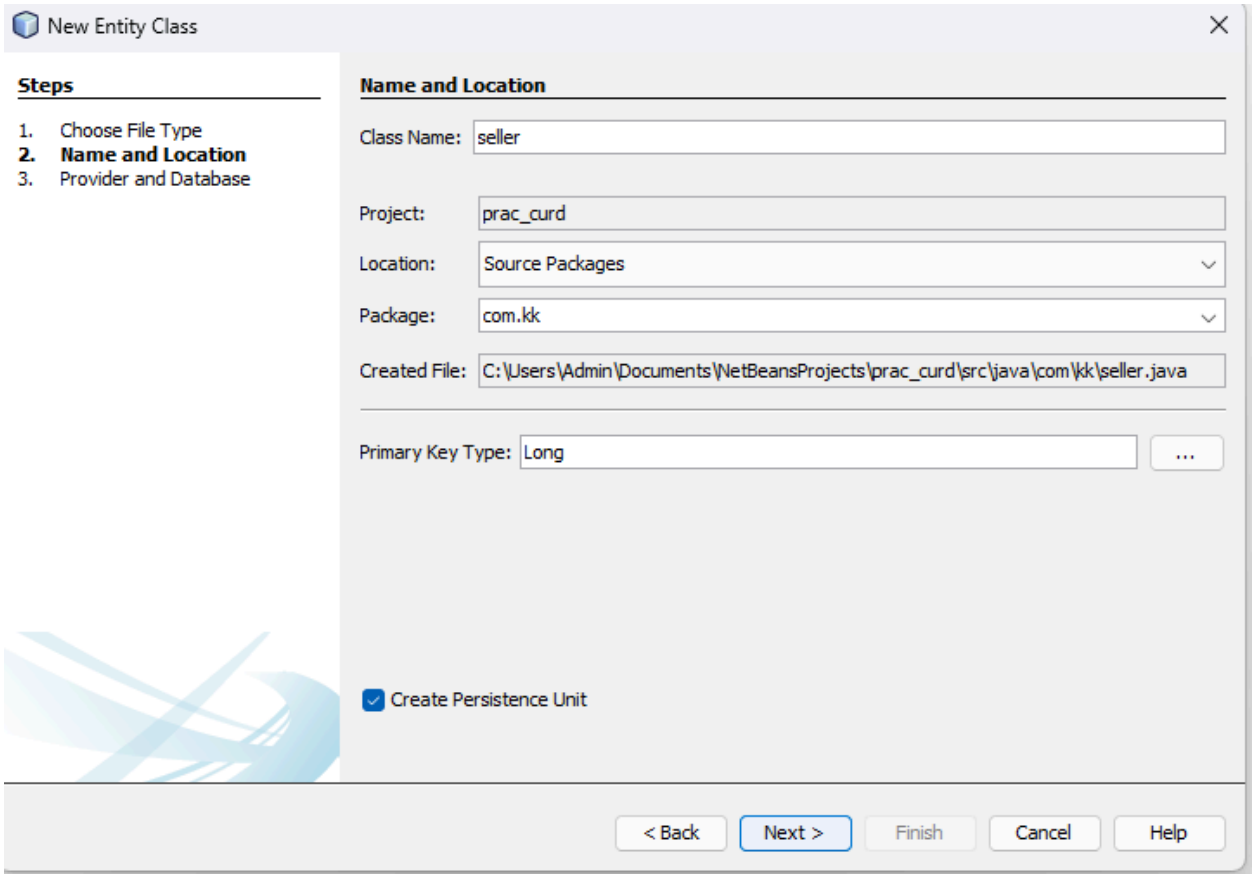
Step 1:- Create an web application curd
Step 2:- connect your db go to services click on Java DB & Connect it



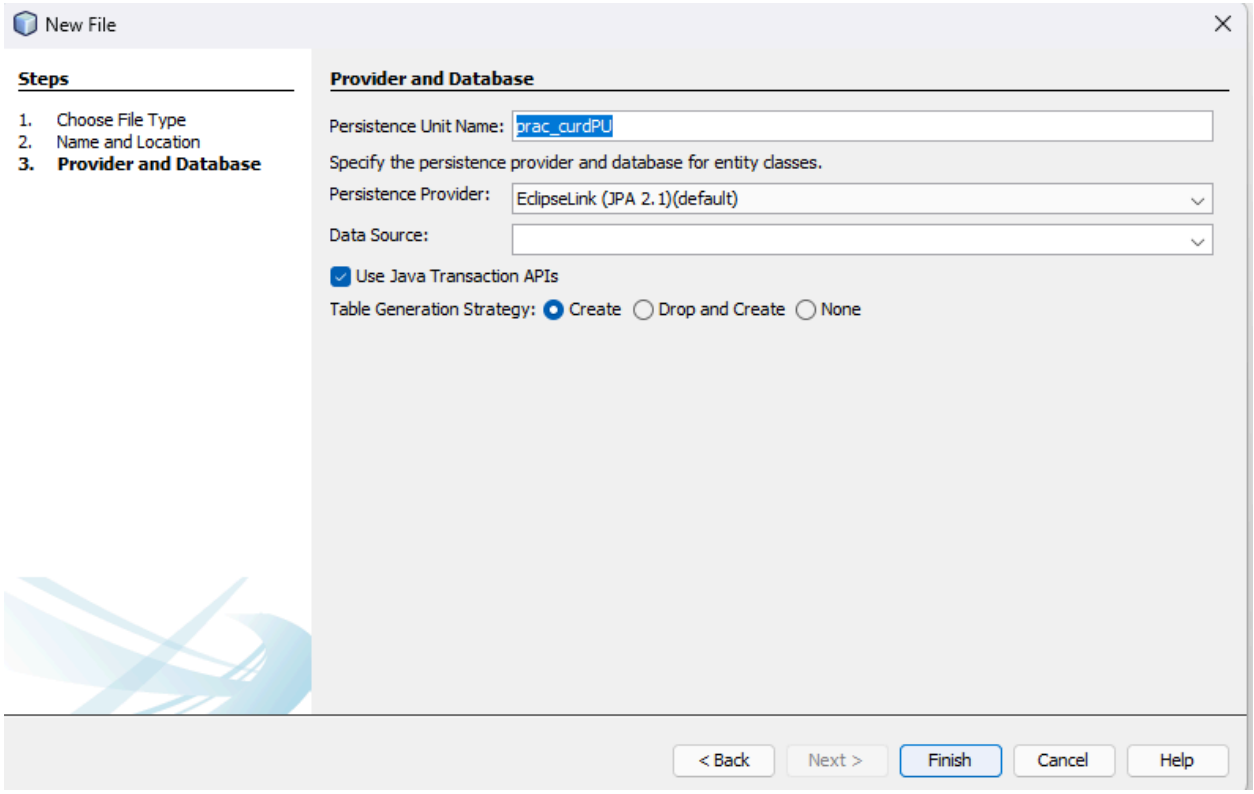


Step 3:- create an entity class

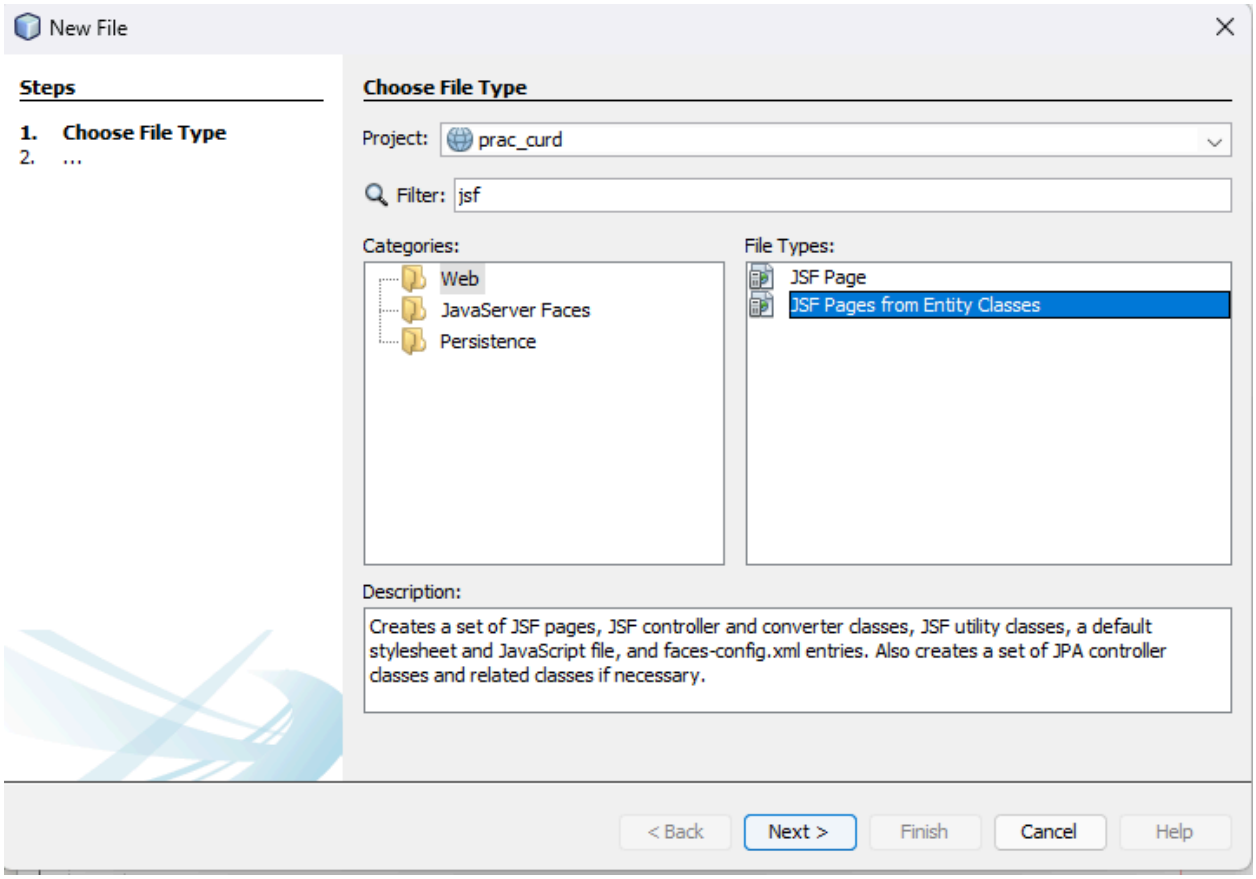




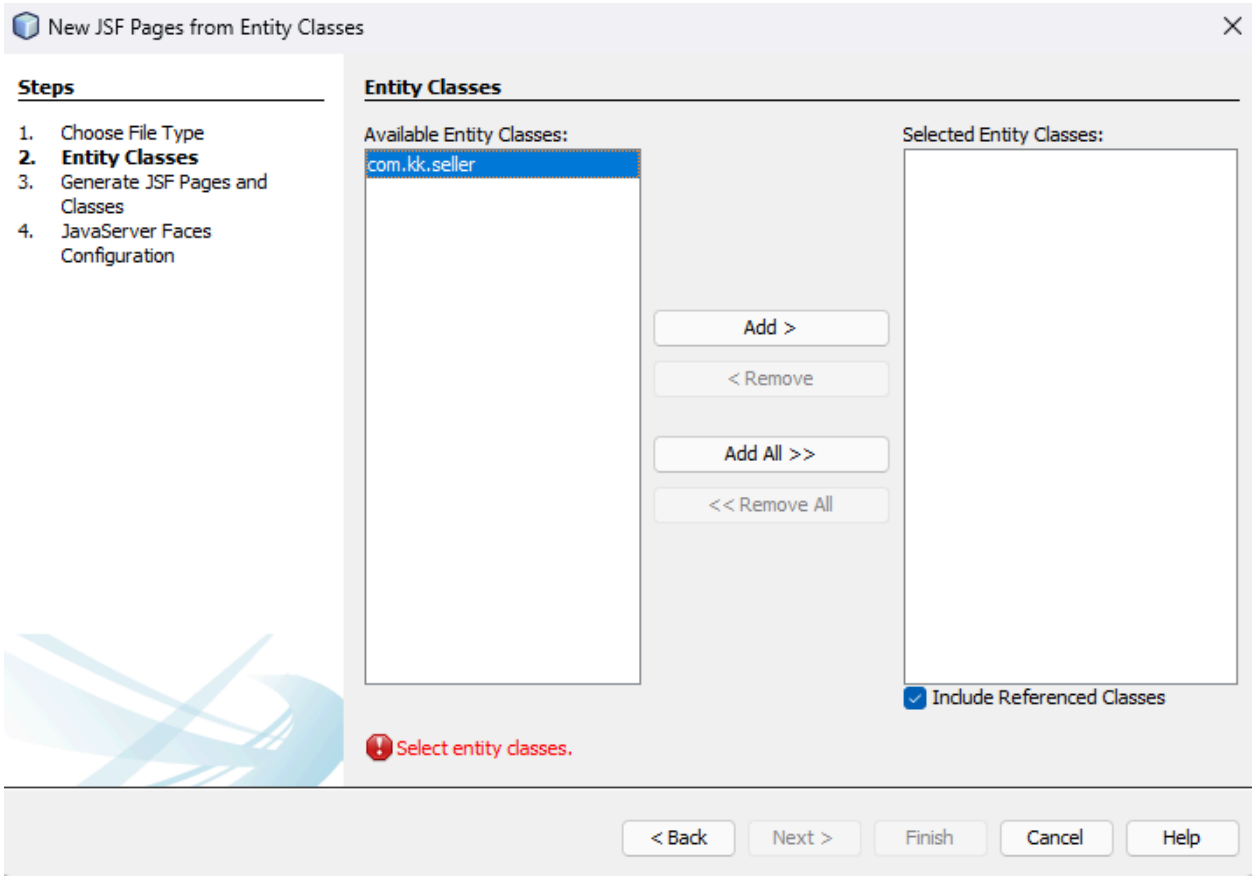
Click on finish

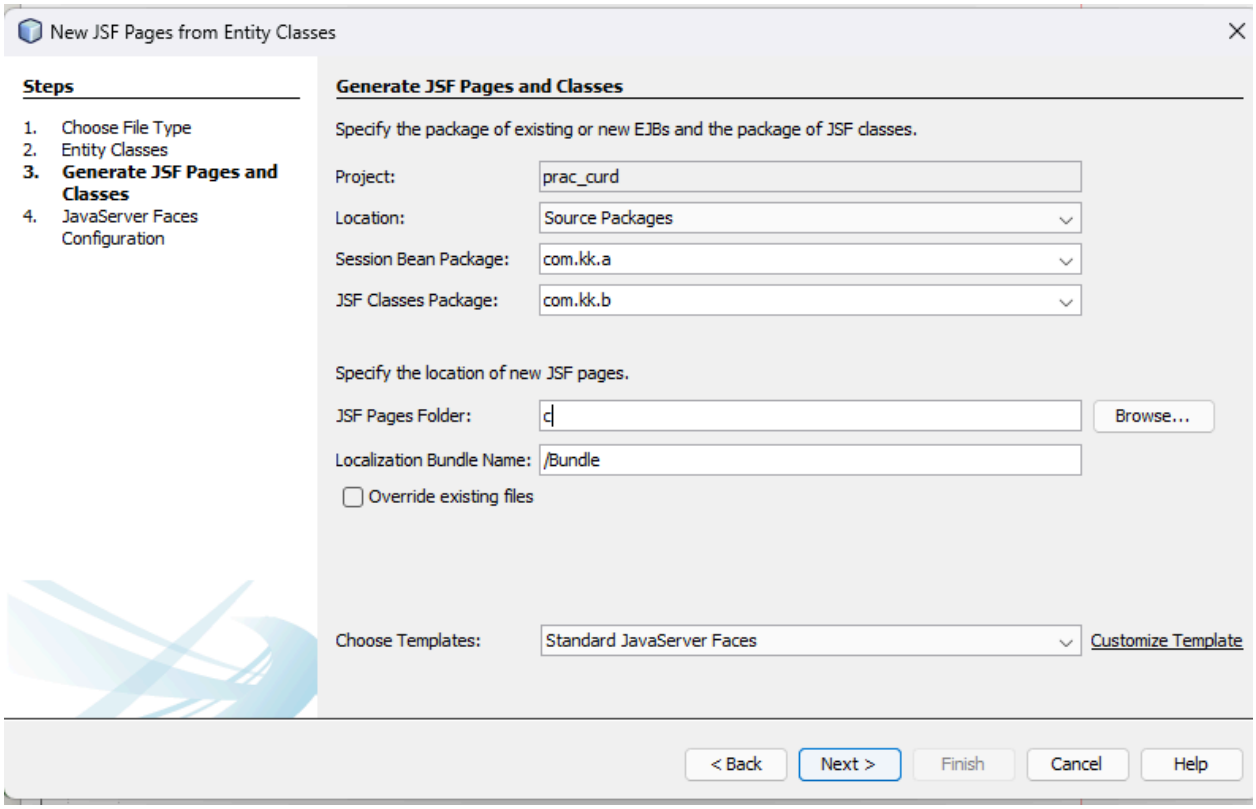


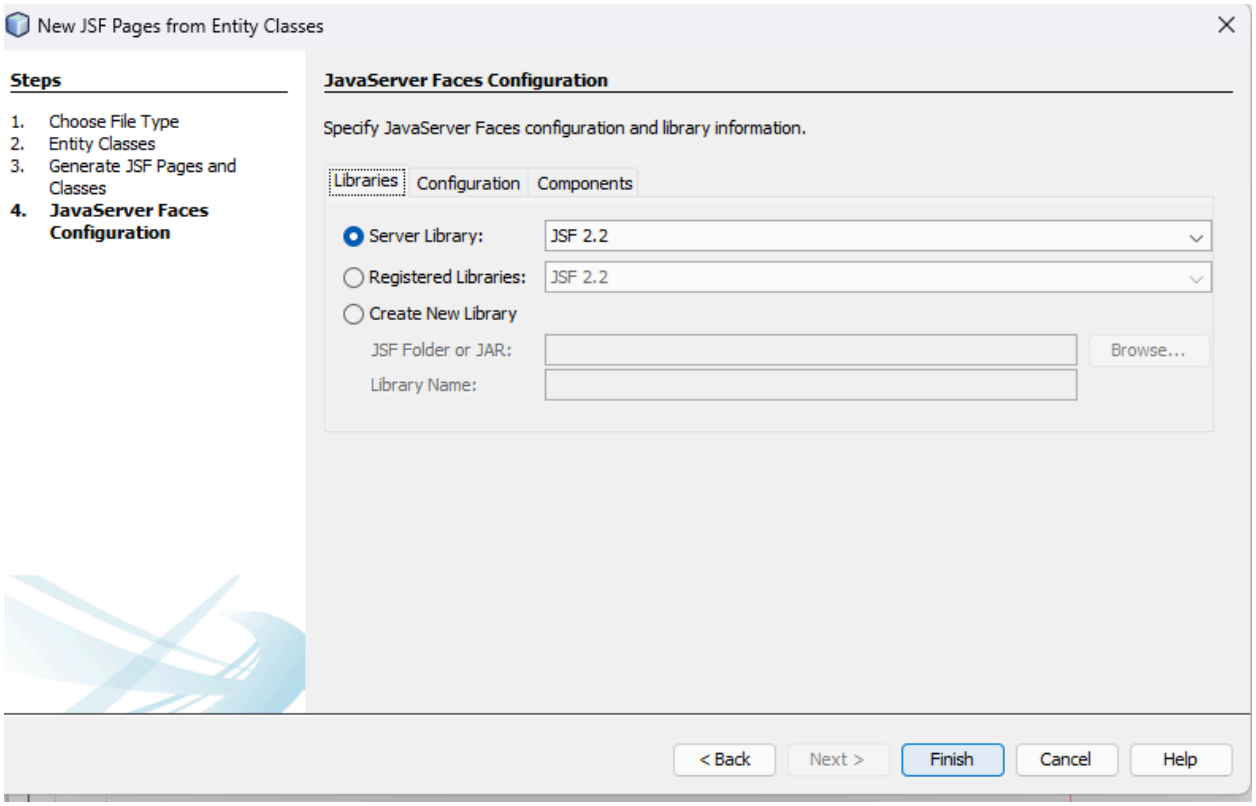
Step :- 4 select the jsf page from entity class.



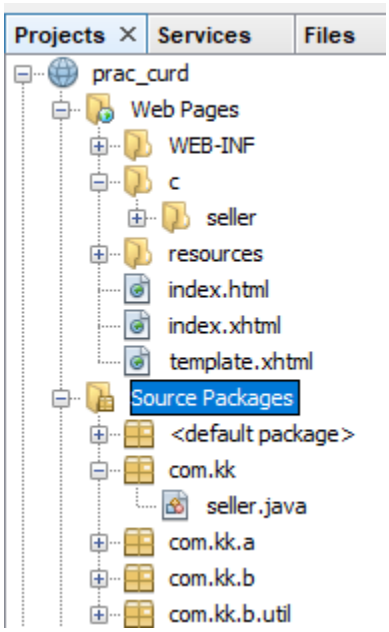
Add the created entity class


New JSF Pages from Entity Classes
Steps
1. Choose File Type
2. Entity Classes
3. Generate JSF Pages and Classes
4. JavaServer Faces Configuration
Entity Classes
Available Entity Classes:
com.kk.seller
Add >
< Remove
Add All >>
<< Remove All
Selected Entity Classes:
Include Referenced Classes
Select entity classes.
< Back
Next >
Finish
Cancel
Help

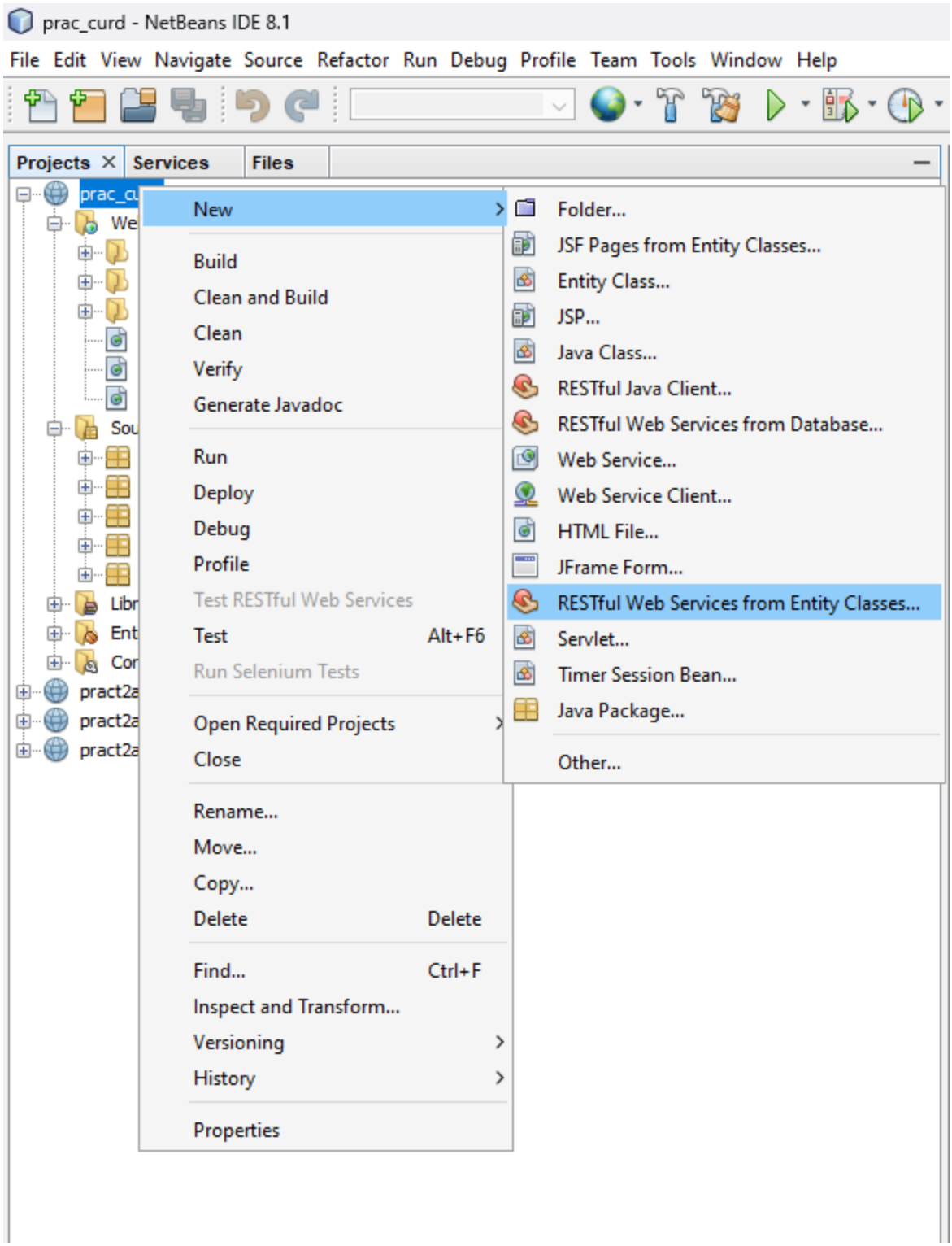

New JSF Pages from Entity Classes
Steps
1. Choose File Type
2. Entity Classes
3. Generate JSF Pages and Classes
4. JavaServer Faces Configuration
Generate JSF Pages and Classes
Specify the package of existing or new EJBs and the package of JSF classes.
Project:
prac_curd
Location:
Source Packages
Session Bean Package:
com.kk.a
JSF Classes Package:
com.kk.b
Specify the location of new JSF pages.
JSF Pages Folder:
c
Browse...
Localization Bundle Name:
/Bundle
Override existing files
Choose Templates:
Standard JavaServer Faces
Customize Template
< Back
Next >
Finish
Cancel
Help

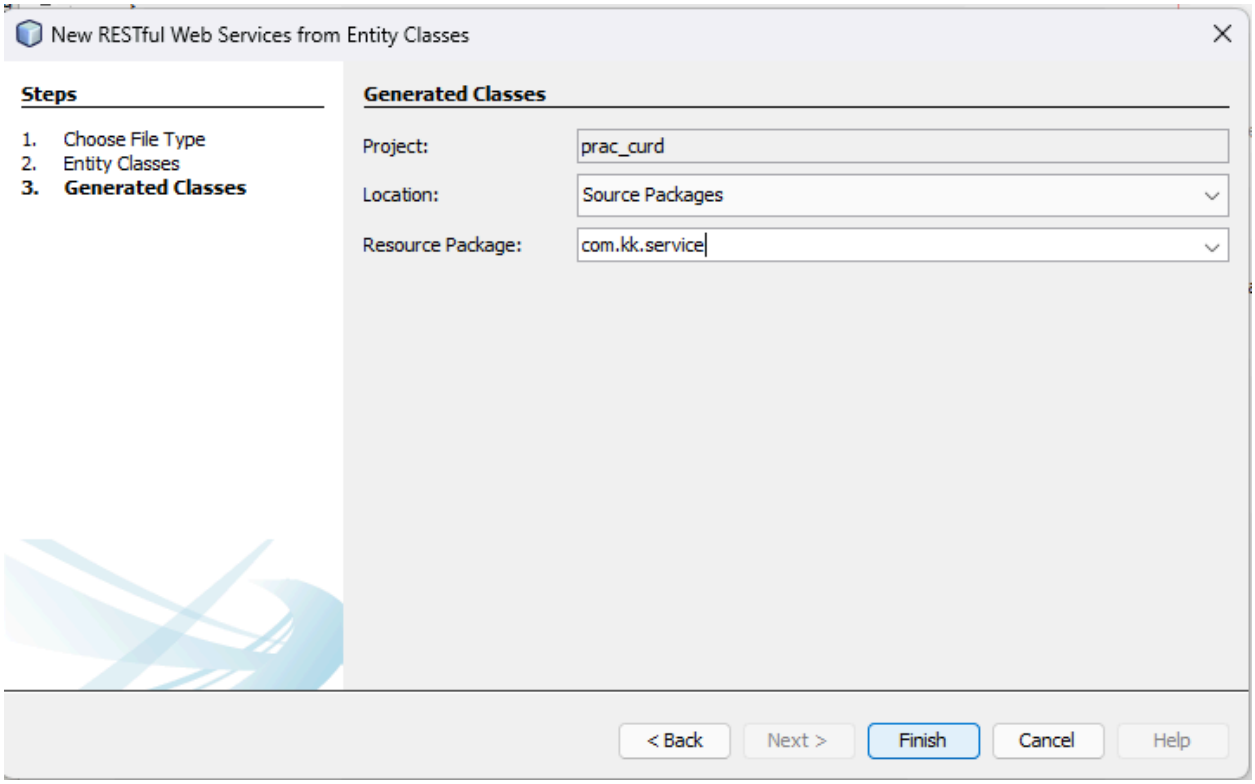


Packages will be created

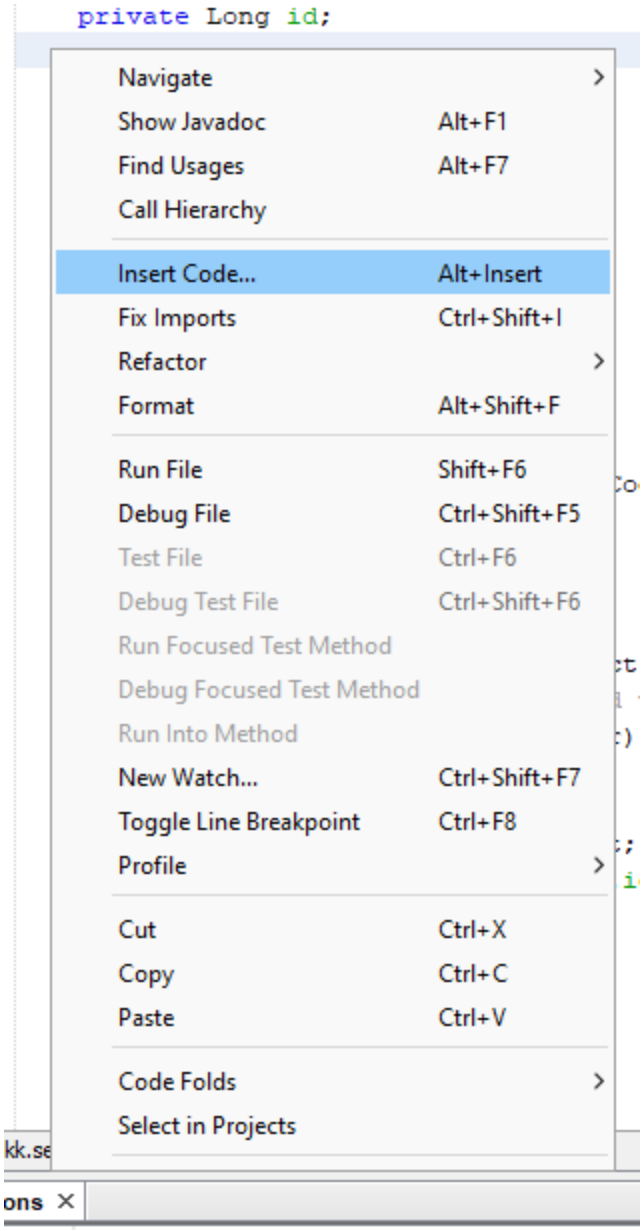


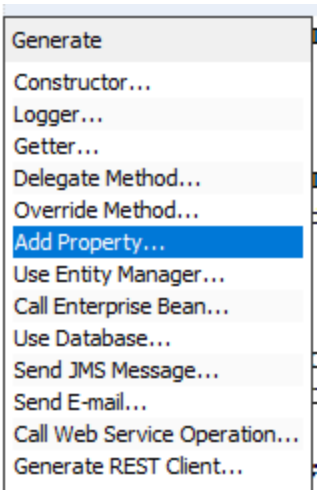
Step 6:- select the restful web services from entity class

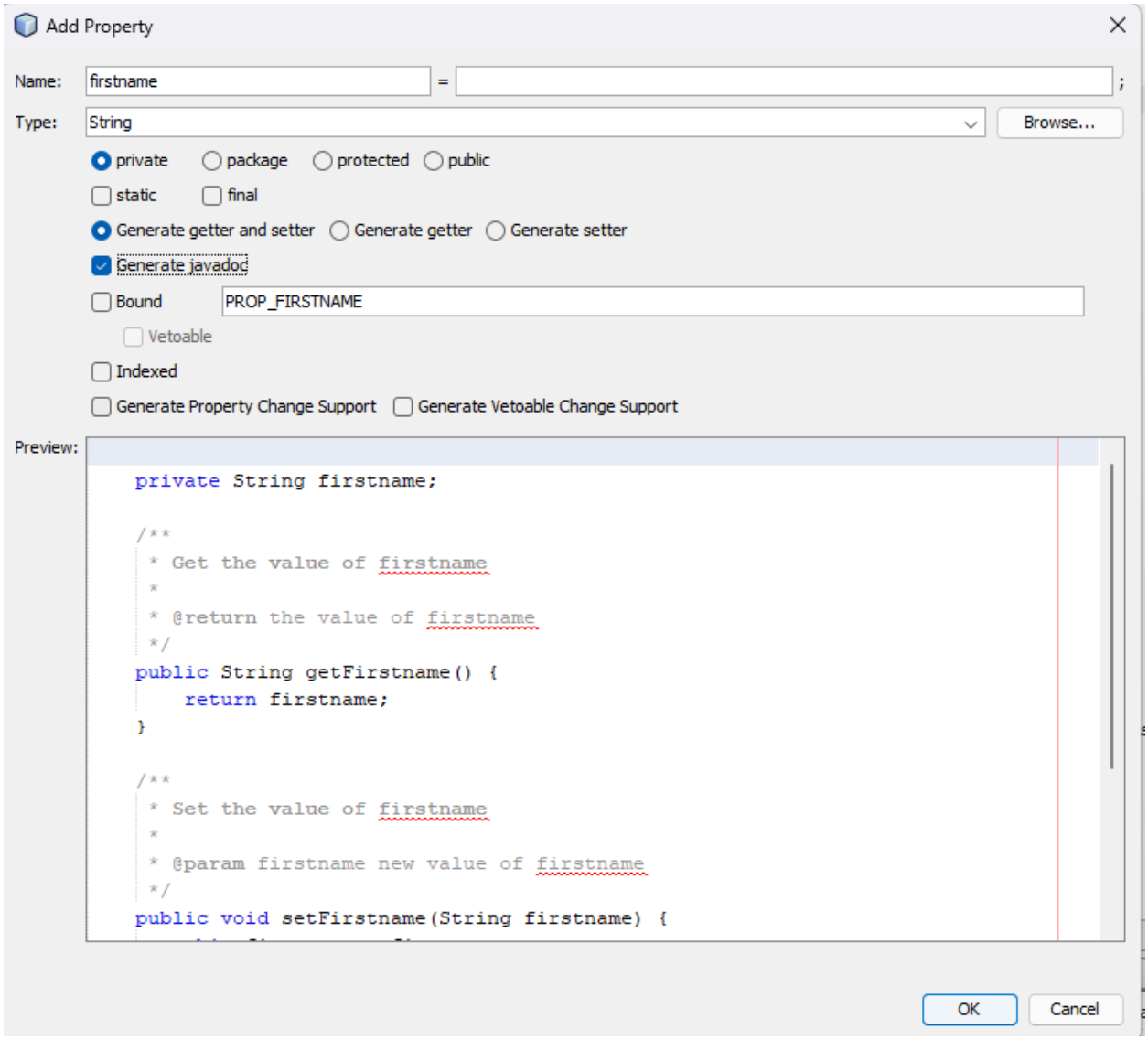

prac_curd - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Services
Files
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
JSF Pages from Entity Classes...
Entity Class...
JSP...
Java Class...
RESTful Java Client...
RESTful Web Services from Database...
Web Service...
Web Service Client...
HTML File...
JFrame Form...
RESTful Web Services from Entity Classes...
Servlet...
Timer Session Bean...
Java Package...
Other...

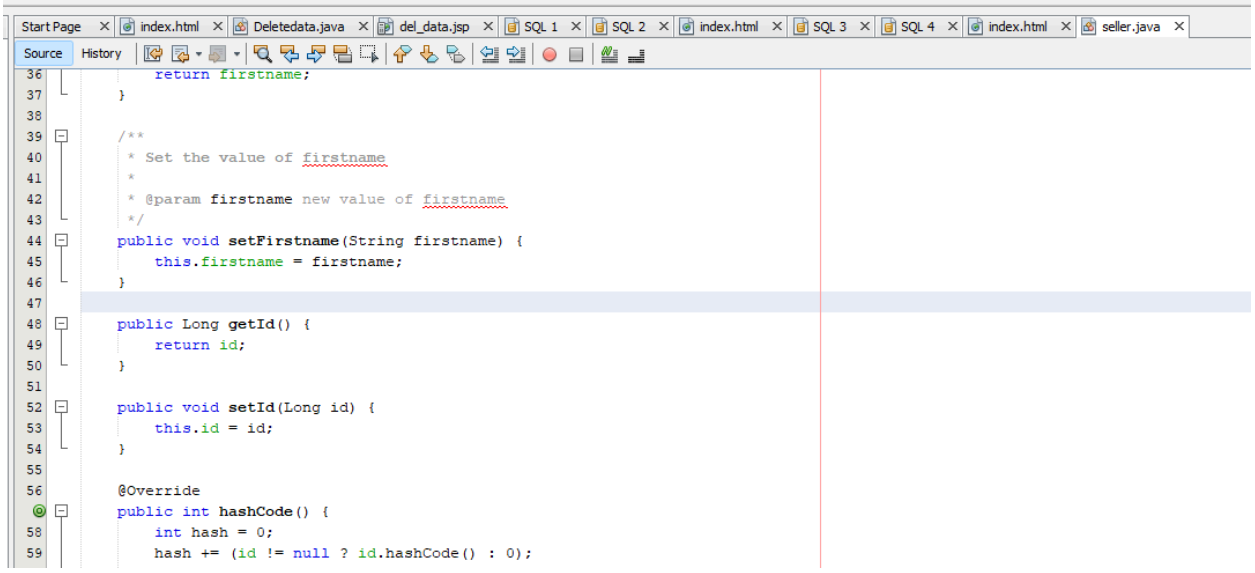


Go to seller.java Right Click on line no 27

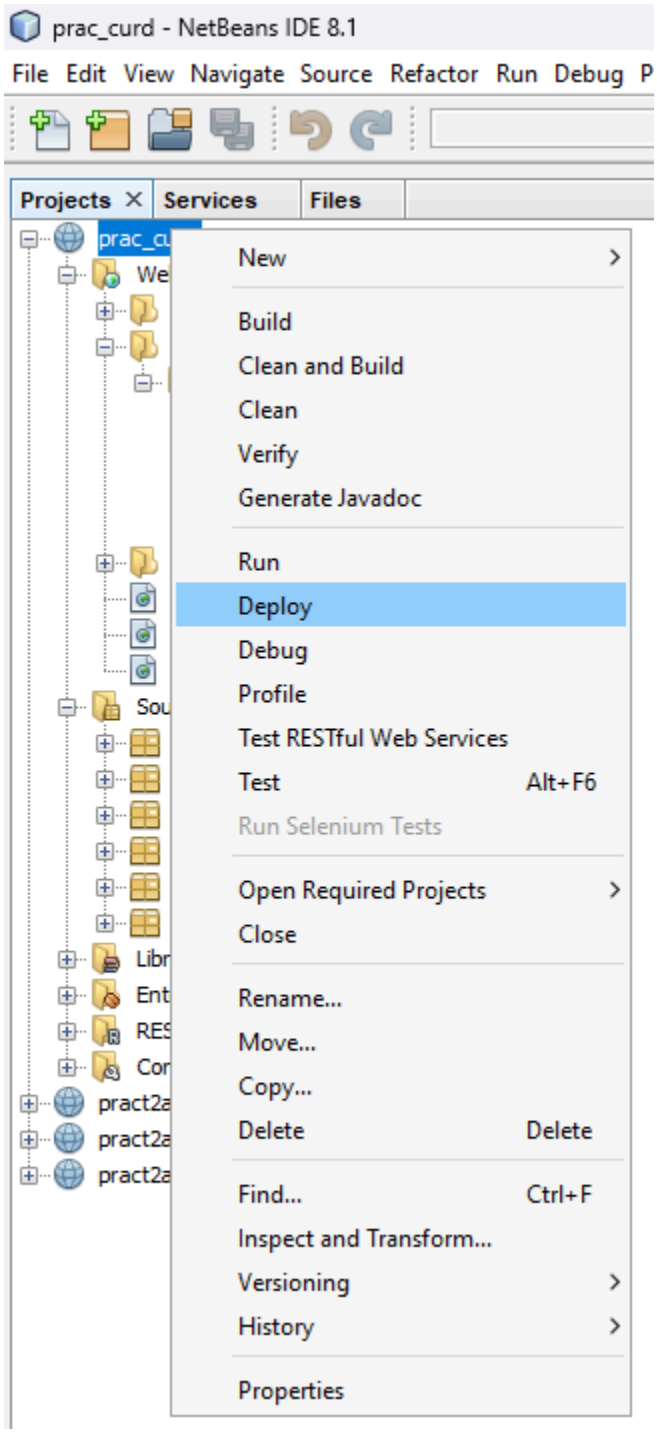

private Long id;
Navigate
Show Javadoc Alt+F1
Find Usages Alt+F7
Call Hierarchy
Insert Code... Alt+Insert
Fix Imports Ctrl+Shift+I
Refactor
Format Alt+Shift+F
Run File Shift+F6
Debug File Ctrl+Shift+F5
Test File Ctrl+F6
Debug Test File Ctrl+Shift+F6
Run Focused Test Method
Debug Focused Test Method
Run Into Method
New Watch... Ctrl+Shift+F7
Toggle Line Breakpoint Ctrl+F8
Profile
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Code Folds
Select in Projects

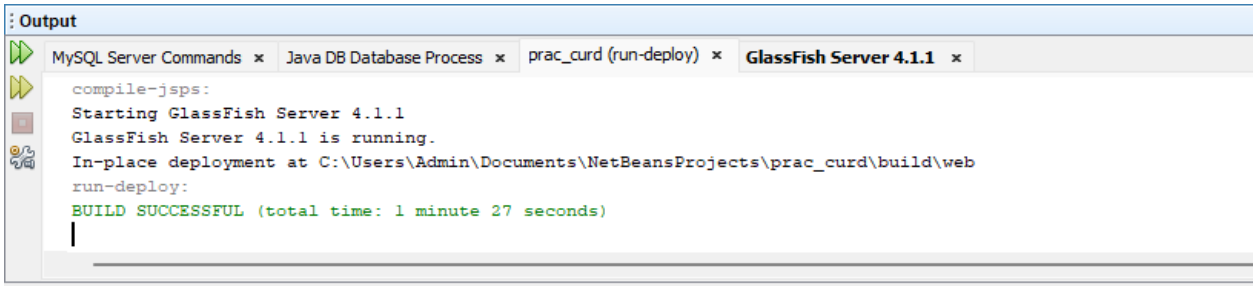

Generate
Constructor...
Logger...
Getter...
Delegate Method...
Override Method...
Add Property...
Use Entity Manager...
Call Enterprise Bean...
Use Database...
Send JMS Message...
Send E-mail...
Call Web Service Operation...
Generate REST Client...


Add Property
Name: firstname = ;
Type: String Browse...
private package protected public
static final
Generate getter and setter Generate getter Generate setter
Generate javadoc
Bound PROP_FIRSTNAME
Vetoable
Indexed
Generate Property Change Support Generate Vetoable Change Support
Preview:
private String firstname;
/**
* Get the value of firstname
*
* @return the value of firstname
*/
public String getFirstname() {
return firstname;
}
/**
* Set the value of firstname
*
* @param firstname new value of firstname
*/
public void setFirstname(String firstname) {
OK
Cancel

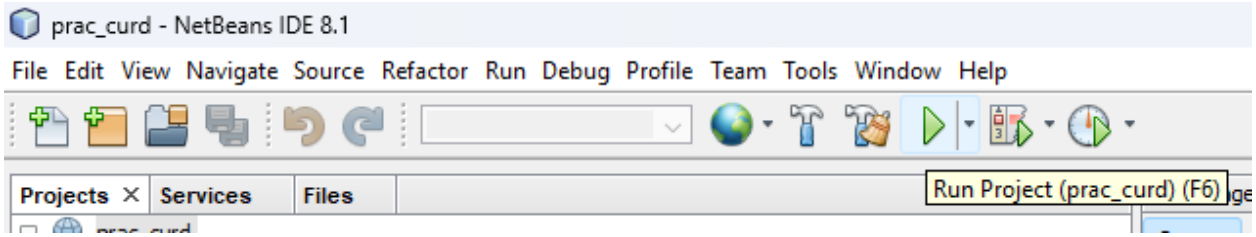

Start Page
index.html
Deletedata.java
del_data.jsp
SQL 1
SQL 2
index.html
SQL 3
SQL 4
index.html
seller.java
Source
History
36 return firstname;
37 }
38
39 /**
40 * Set the value of firstname
41 *
42 * @param firstname new value of firstname
43 */
44 public void setFirstname(String firstname) {
45 this.firstname = firstname;
46 }
47
48 public Long getId() {
49 return id;
50 }
51
52 public void setId(Long id) {
53 this.id = id;
54 }
55
56 @Override
public int hashCode() {
58 int hash = 0;
59 hash += (id != null ? id.hashCode() : 0);

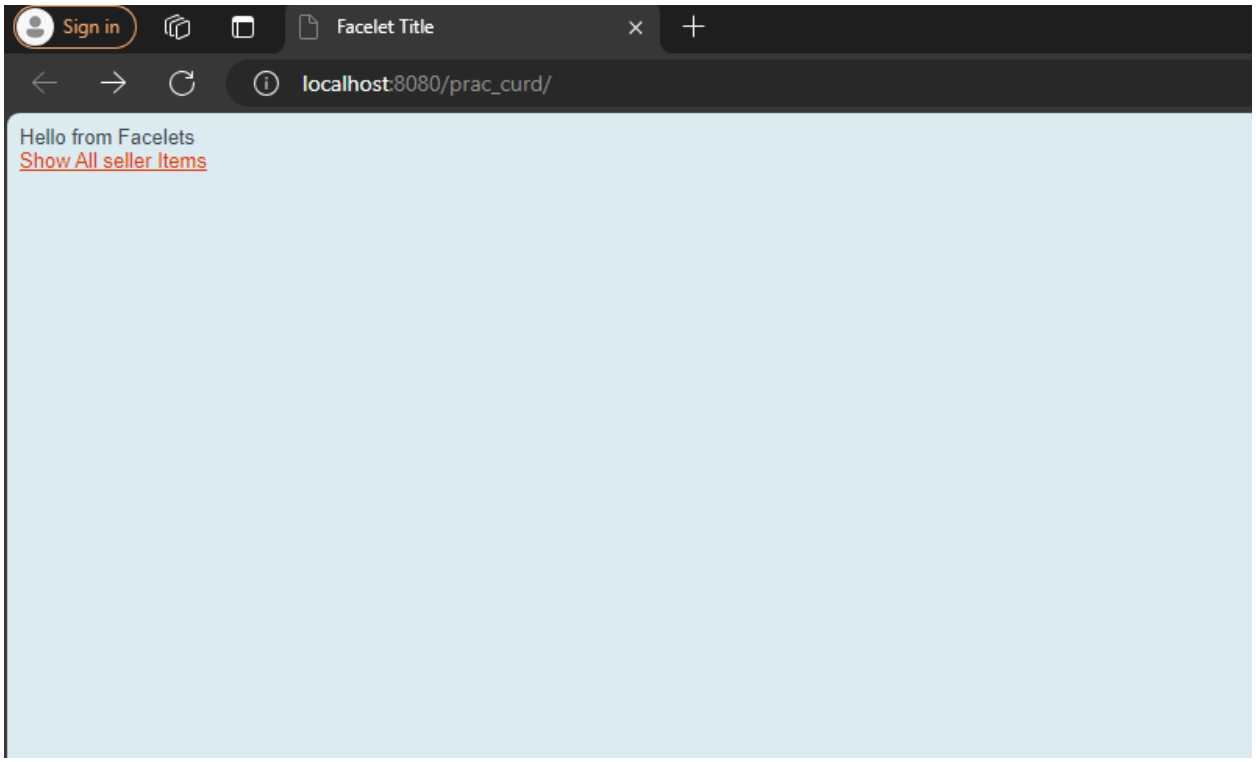
Step 7:- Deploy your project

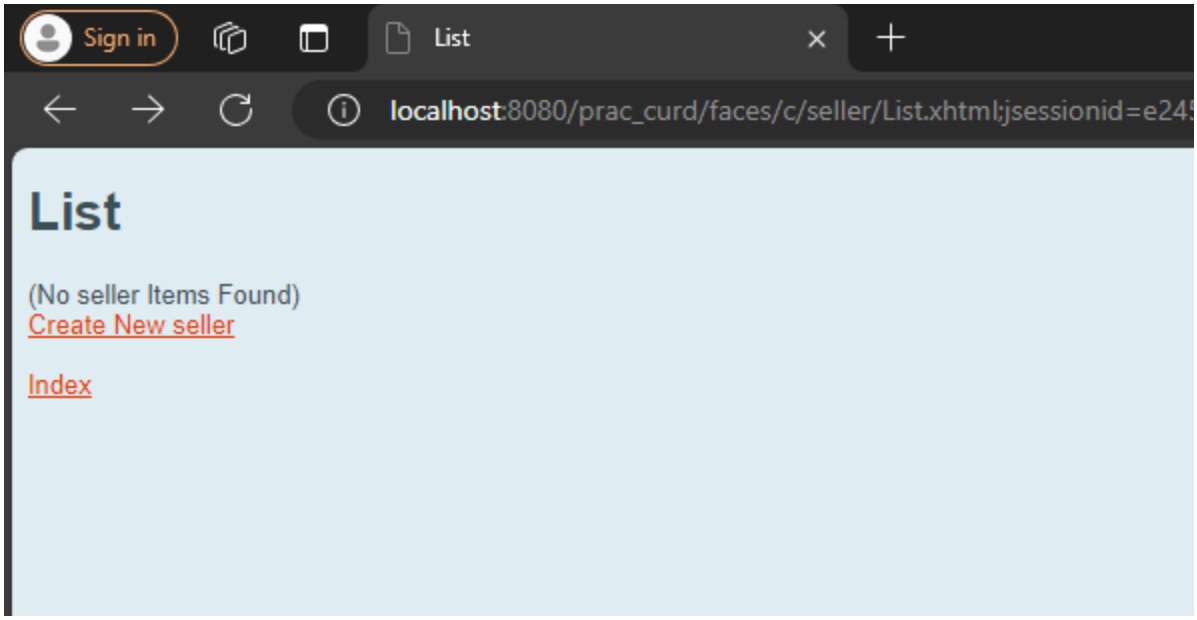




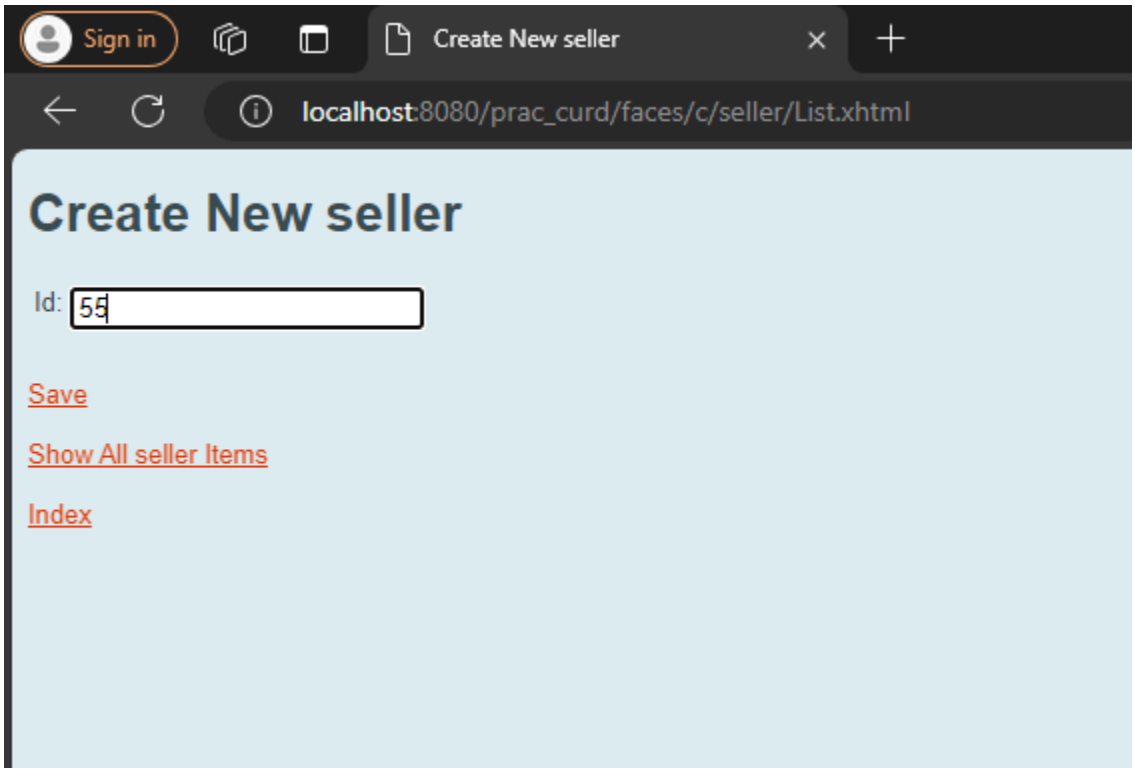
After the deployment run your project

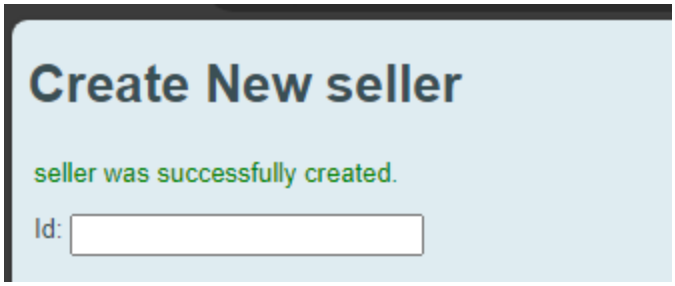




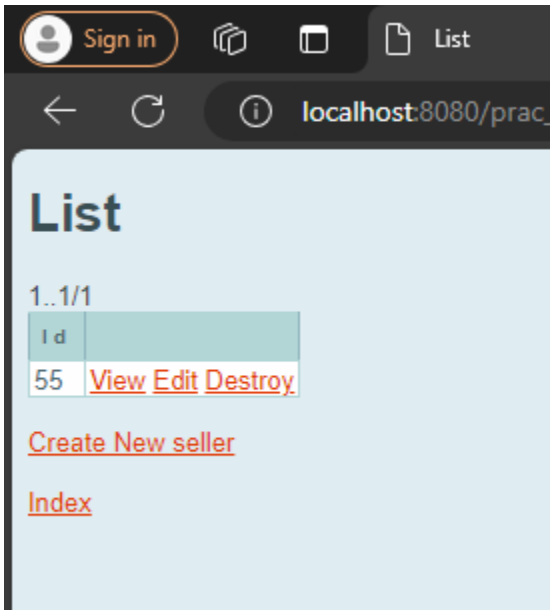


Add the id here and save it .

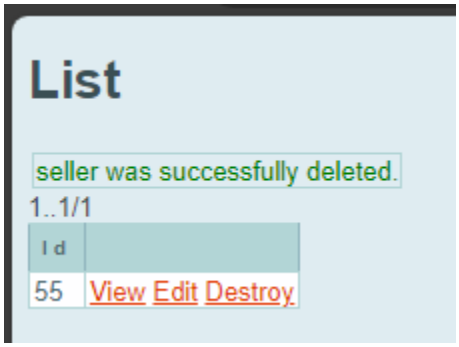




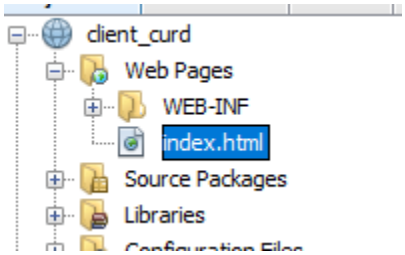
Click on show all seller items



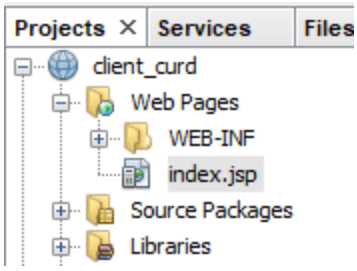
Click on the destory to delete the data created.



Create another project as cloient_Curd

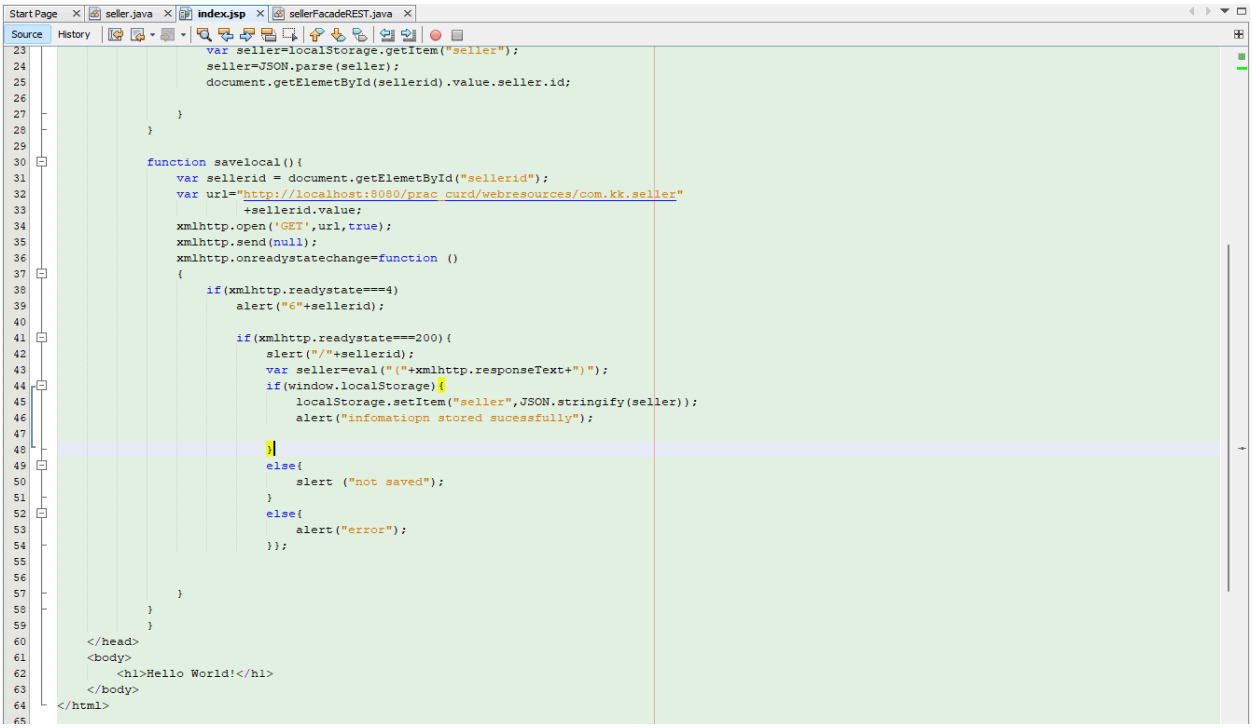


Delete the index .html and create an index .jsp page



Enter the following code in index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <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.getElemetById(sellerid).value.seller.id;

                }
            }

            function savelocal(){
                var sellerid = document.getElemetById("sellerid");
                var url="http://localhost:8080/prac_curd/webresources/com.kk.seller"
            }
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
```



Unset

Working corrected code :-

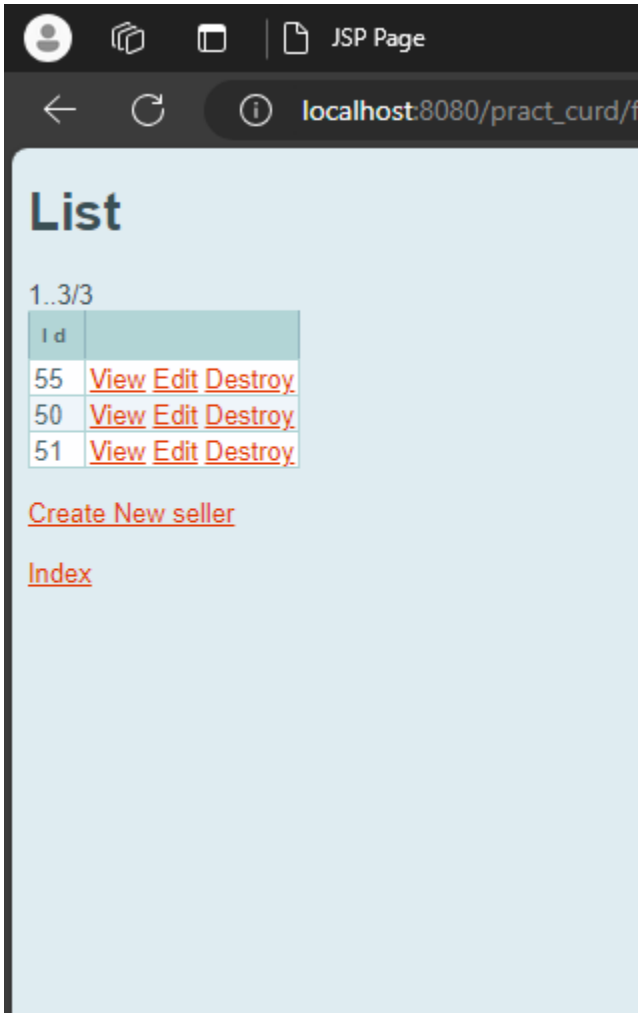
Unset

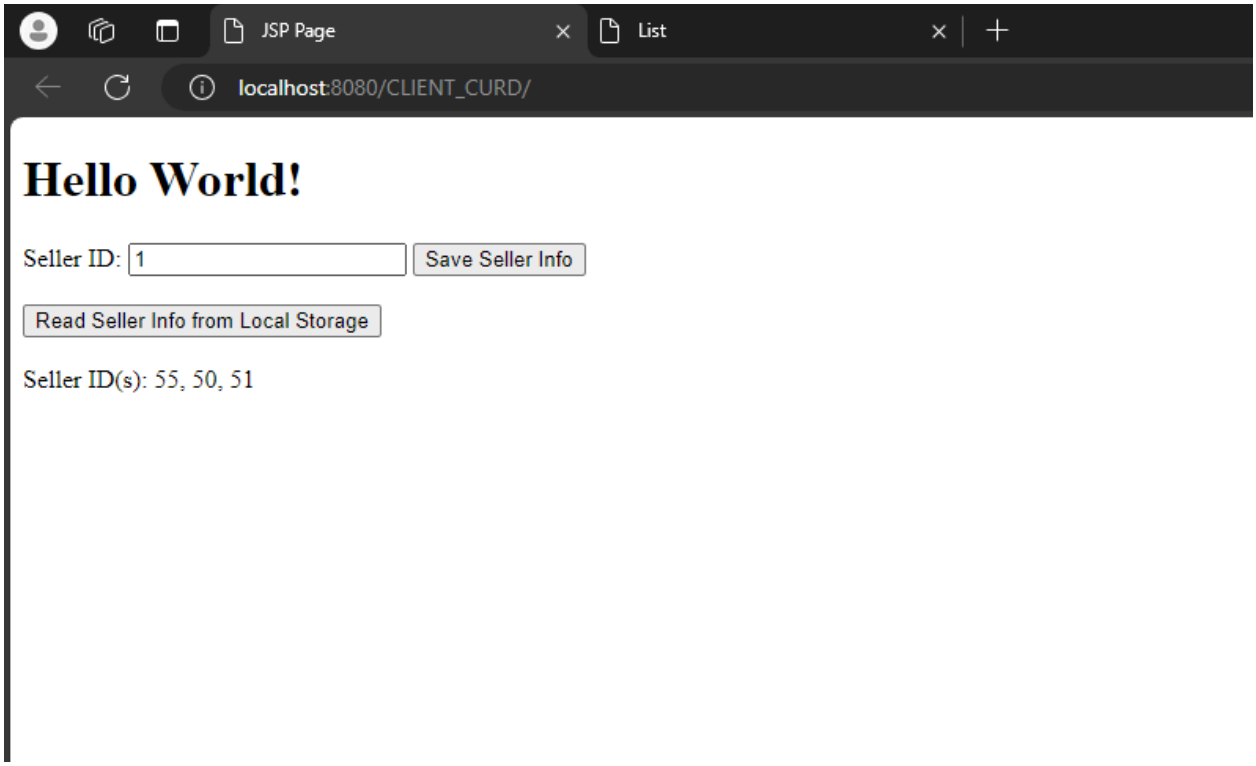
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
	<head>
	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
	<title>JSP Page</title>
	<script language="javascript">
		var xmlhttp;

		function init() {
		xmlhttp = new XMLHttpRequest();
		}

		function readLocal() {
		if (window.localStorage) {
```

```
                    var seller = localStorage.getItem("seller");
                    if (seller) {
                          seller = JSON.parse(seller);
                          document.getElementById("sellerIdLabel").innerText =
"Seller ID(s): " + seller.ids.join(", ");
                    } else {
                          alert("No seller data found in local storage.");
                    }
              } else {
                    alert("Local Storage is not supported.");
              }
              }

              function savelocal() {
              console.log('1');
              var sellerid = document.getElementById("sellerid").value;
              console.log('sellerid', sellerid);
              var url =
"http://localhost:8080/pract_curd/webresources/com.kk.seller/"; // Assuming
this URL now gives multiple sellers
              xmlhttp.open('GET', url, true);
              console.log('outside', url);

              xmlhttp.onreadystatechange = function () {
                    console.log('inside funct');
                    if (xmlhttp.readyState === 4) {
                          console.log('2');

                          // Log the raw server response
                          console.log("Raw server response:",
xmlhttp.responseText);

                          // Parse the XML response
                          var parser = new DOMParser();
                          var xmlDoc =
parser.parseFromString(xmlhttp.responseText, "application/xml");

                          // Extract all seller IDs from the XML
                          var sellerIds = [];
                          var sellers = xmlDoc.getElementsByTagName("seller");
                          console.log('romy',sellers)
                          for (var i = 0; i < sellers.length; i++) {
                          var sellerId =
sellers[i].getElementsByTagName("id")[0].childNodes[0].nodeValue;
```

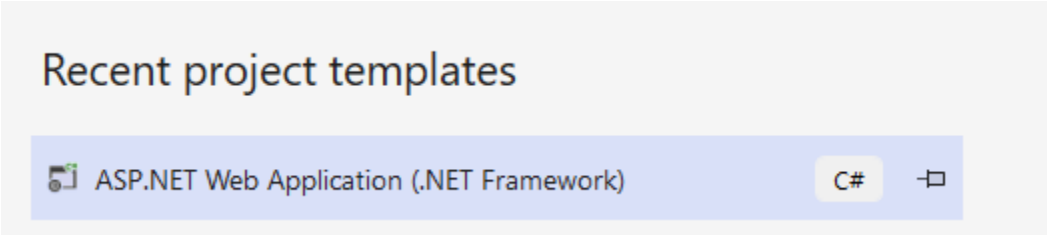
```
                        sellerIds.push(sellerId);
                        }

                        // Log the seller IDs as a JSON object
                        console.log("Parsed Seller IDs as JSON:",
JSON.stringify({ ids: sellerIds }));

                        // Store all seller IDs in localStorage
                        var seller = { ids: sellerIds };
                        if (window.localStorage) {
                        localStorage.setItem("seller",
JSON.stringify(seller));
                        alert("Information stored successfully.");
                        } else {
                        alert("LocalStorage is not supported.");
                        }

                        // Display the seller IDs in the label
                        document.getElementById("sellerIdLabel").innerText =
"Seller ID(s): " + sellerIds.join(", ");
                    }
             };
             xmlhttp.send();
             }
       </script>
       </head>
       <body onload="init()">
       <h1>Hello World!</h1>

       <label for="sellerid">Seller ID:</label>
       <input type="text" id="sellerid" name="sellerid">
       <button onclick="savelocal()">Save Seller Info</button>
       <br><br>

       <button onclick="readLocal()">Read Seller Info from Local
Storage</button>
       <br><br>

       <label id="sellerIdLabel">Seller ID(s): </label><br>

       </body>
</html>
```

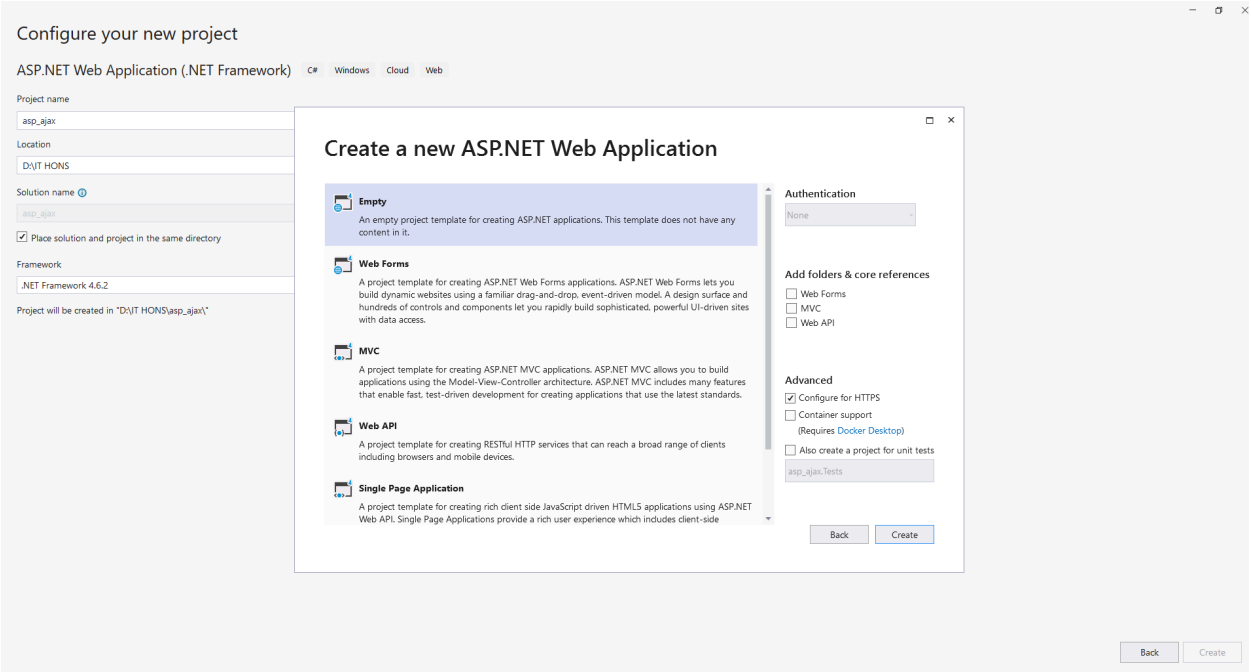
Output:- seller saved in the list are shown in the client curd (index.jsp) page



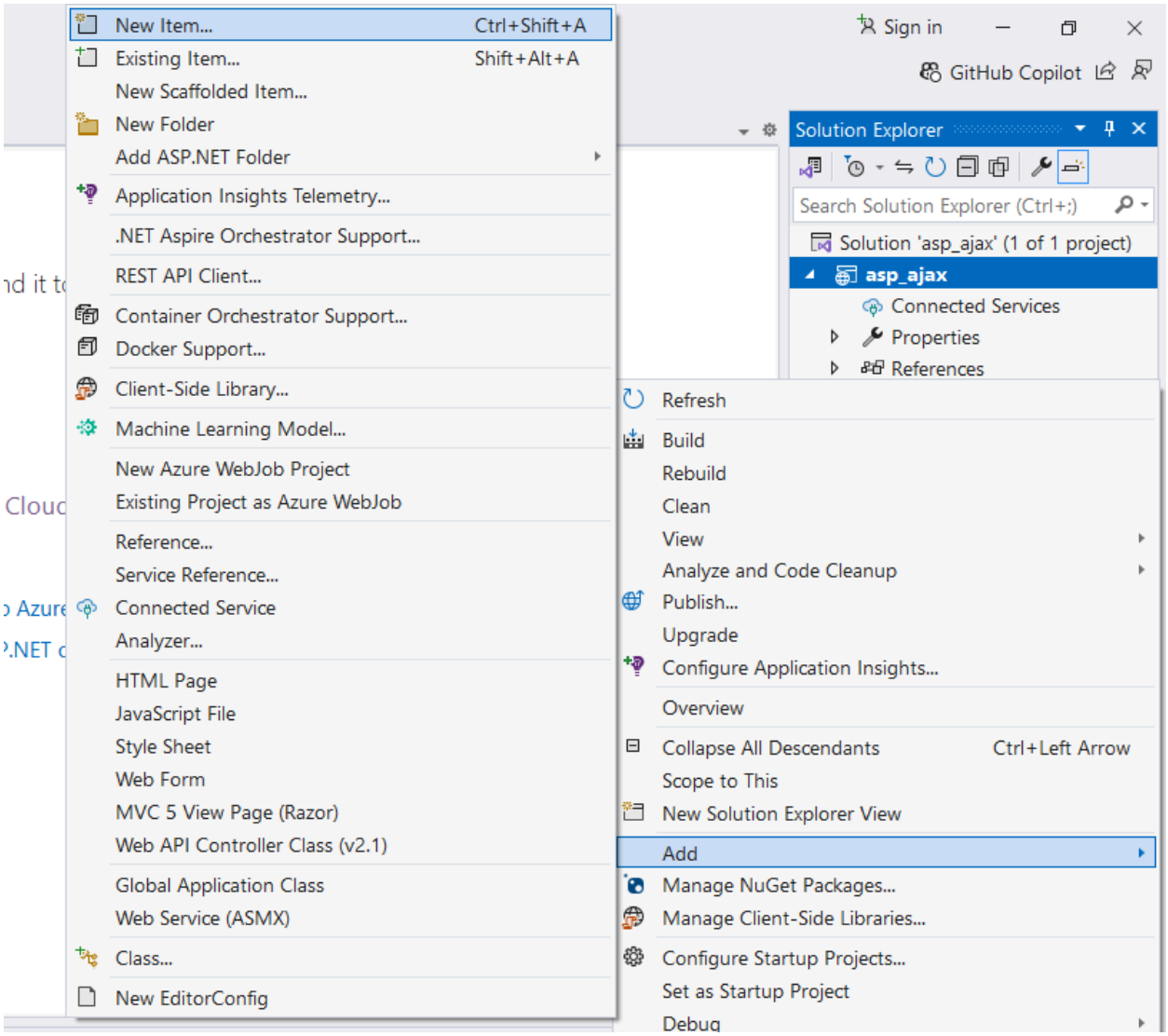

JSP Page
List
localhost:8080/CLIENT_CURD/
Hello World!
Seller ID: 1
Save Seller Info
Read Seller Info from Local Storage
Seller ID(s): 55, 50, 51

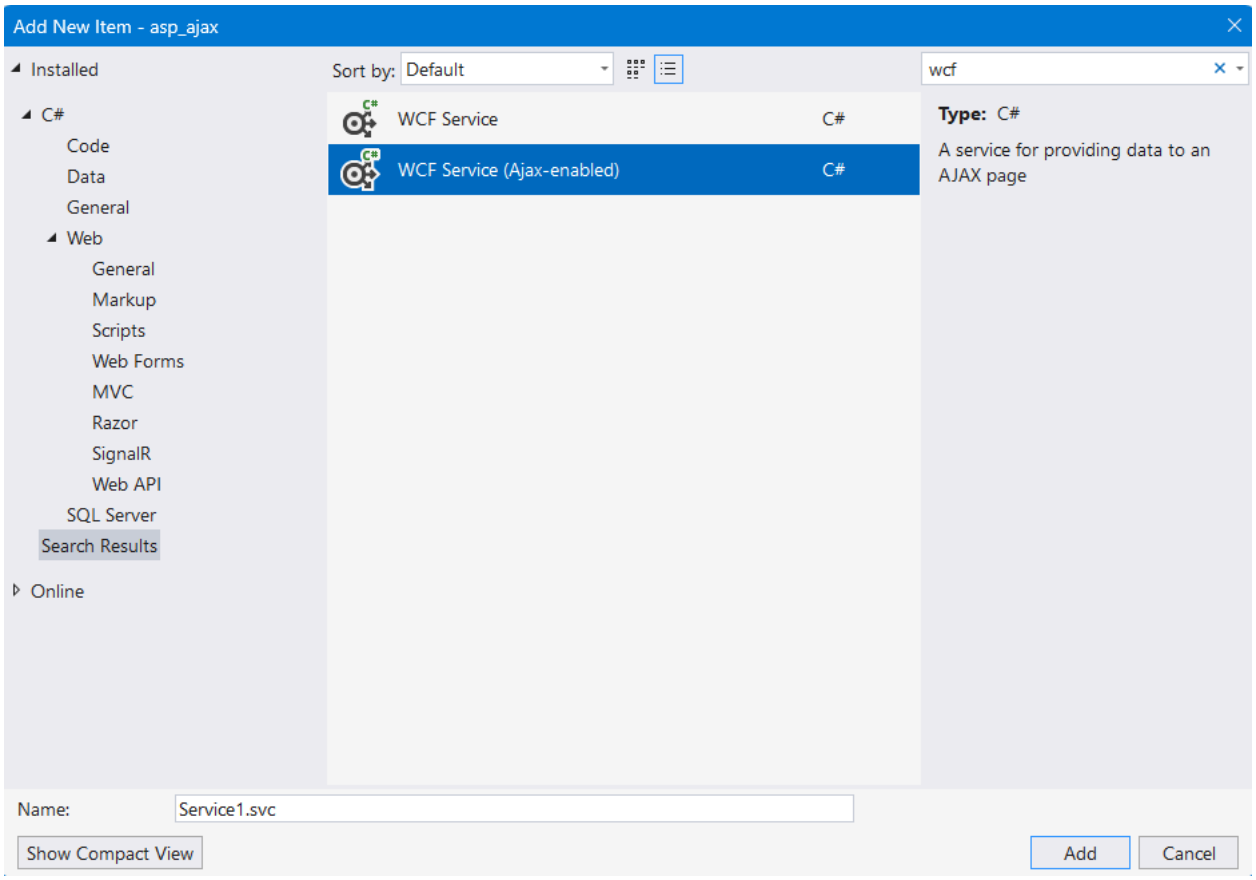
Practical 6

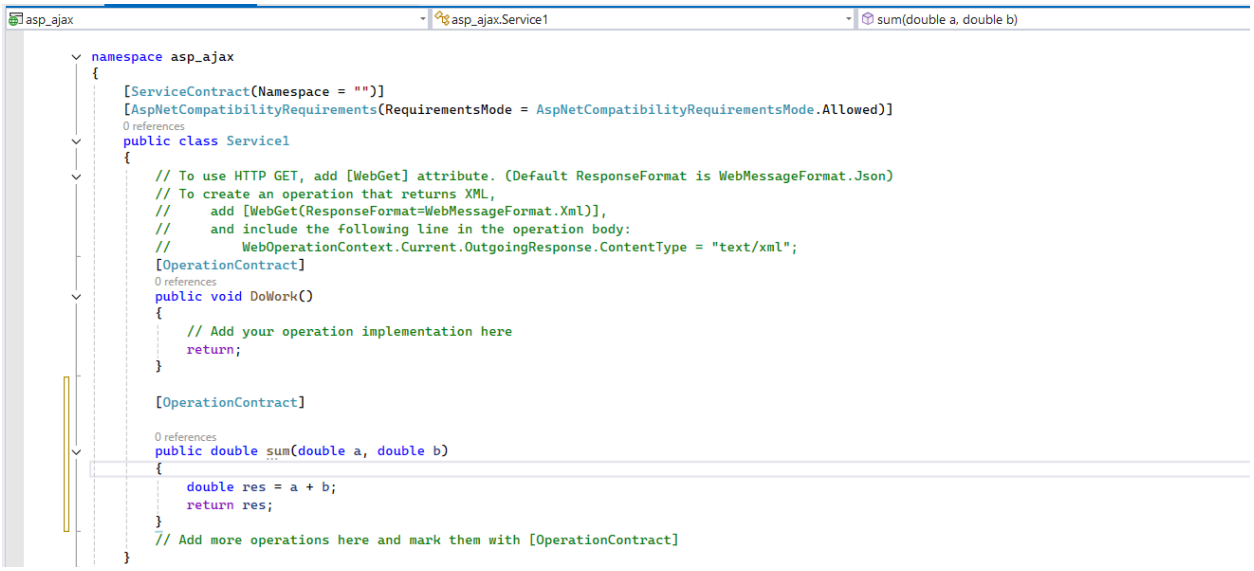




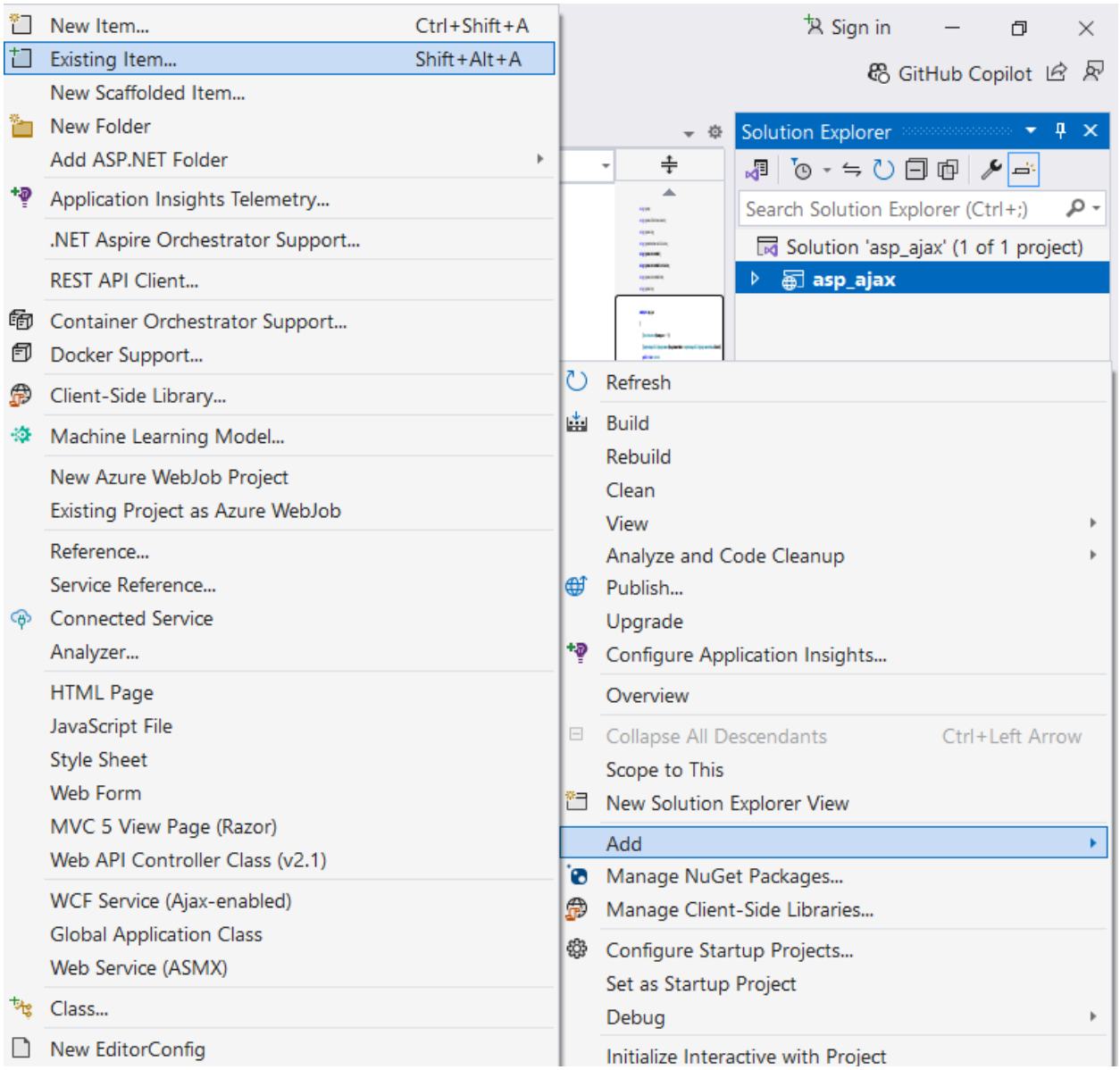
Right click under solution area :


Sign in
GitHub Copilot
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'asp_ajax' (1 of 1 project)
asp_ajax
Connected Services
Properties
References
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Scaffolded Item...
New Folder
Add ASP.NET Folder
Application Insights Telemetry...
.NET Aspire Orchestrator Support...
REST API Client...
Container Orchestrator Support...
Docker Support...
Client-Side Library...
Machine Learning Model...
New Azure WebJob Project
Existing Project as Azure WebJob
Reference...
Service Reference...
Connected Service
Analyzer...
HTML Page
JavaScript File
Style Sheet
Web Form
MVC 5 View Page (Razor)
Web API Controller Class (v2.1)
Global Application Class
Web Service (ASMX)
Class...
New EditorConfig
Refresh
Build
Rebuild
Clean
View
Analyze and Code Cleanup
Publish...
Upgrade
Configure Application Insights...
Overview
Collapse All Descendants Ctrl+Left Arrow
Scope to This
New Solution Explorer View
Add
Manage NuGet Packages...
Manage Client-Side Libraries...
Configure Startup Projects...
Set as Startup Project
Debug

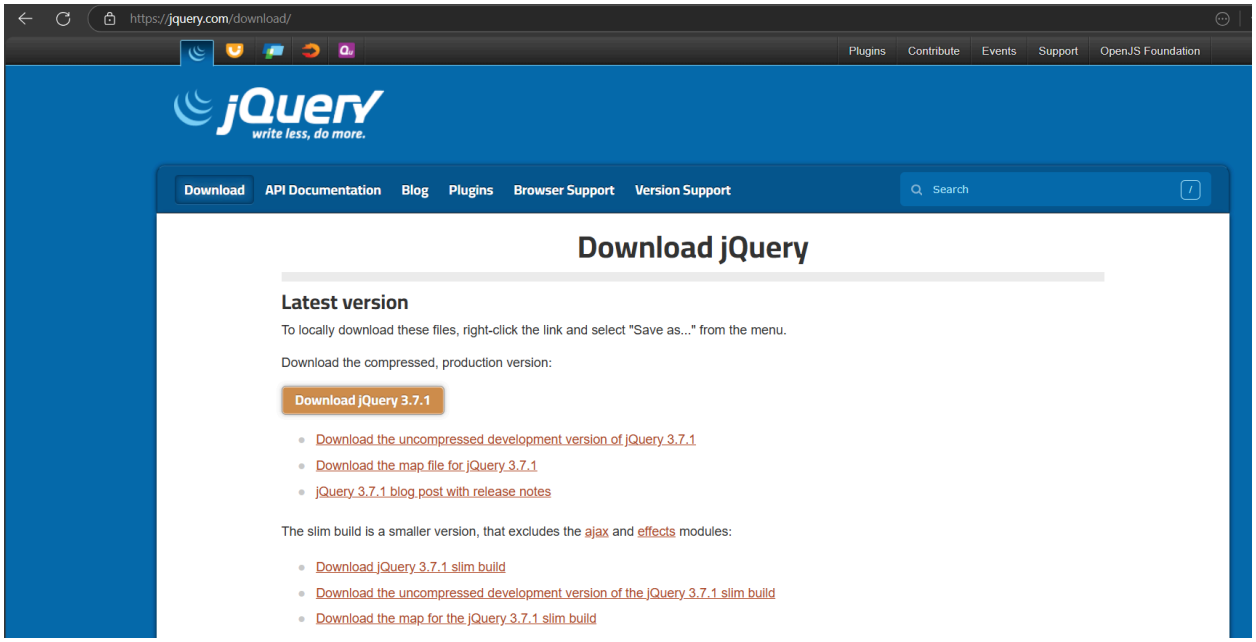


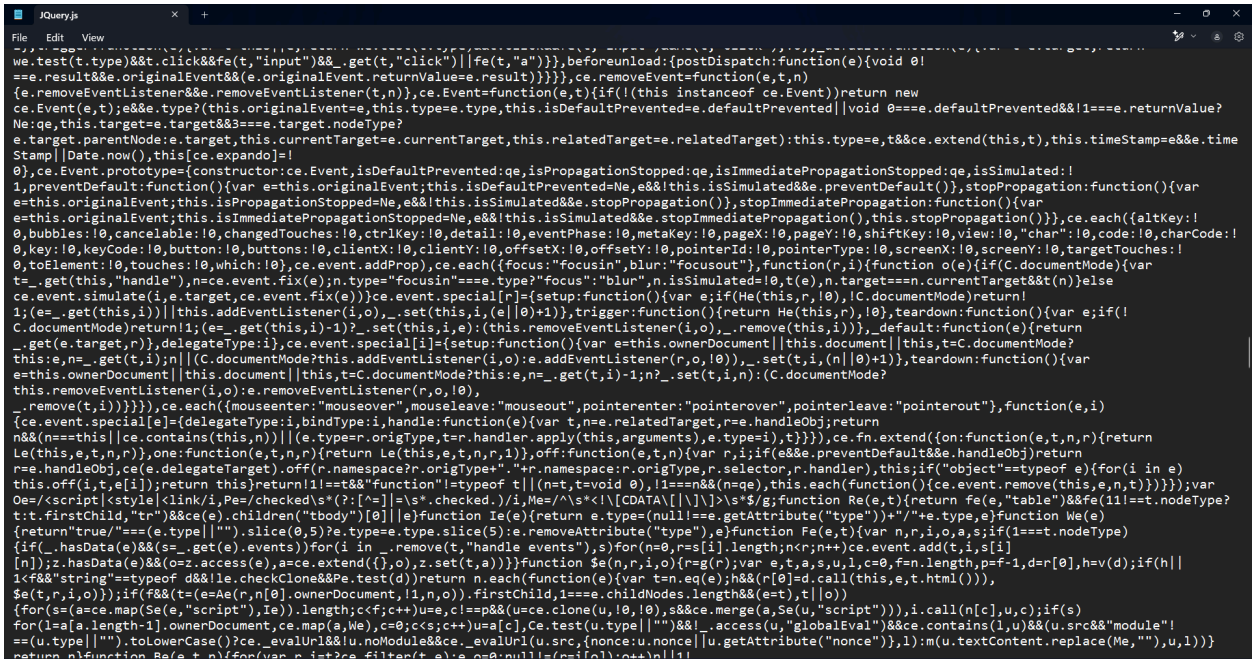


```
asp_ajax                    asp_ajax.Service1                    sum(double a, double b)

namespace asp_ajax
{
    [ServiceContract(Namespace = "")]
    [AspNetCompatibilityRequirements(RequirementsMode = AspNetCompatibilityRequirementsMode.Allowed)]
    0 references
    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]
        0 references
        public void DoWork()
        {
            // Add your operation implementation here
            return;
        }

        [OperationContract]

        0 references
        public double sum(double a, double b)
        {
            double res = a + b;
            return res;
        }
        // Add more operations here and mark them with [OperationContract]
    }
```

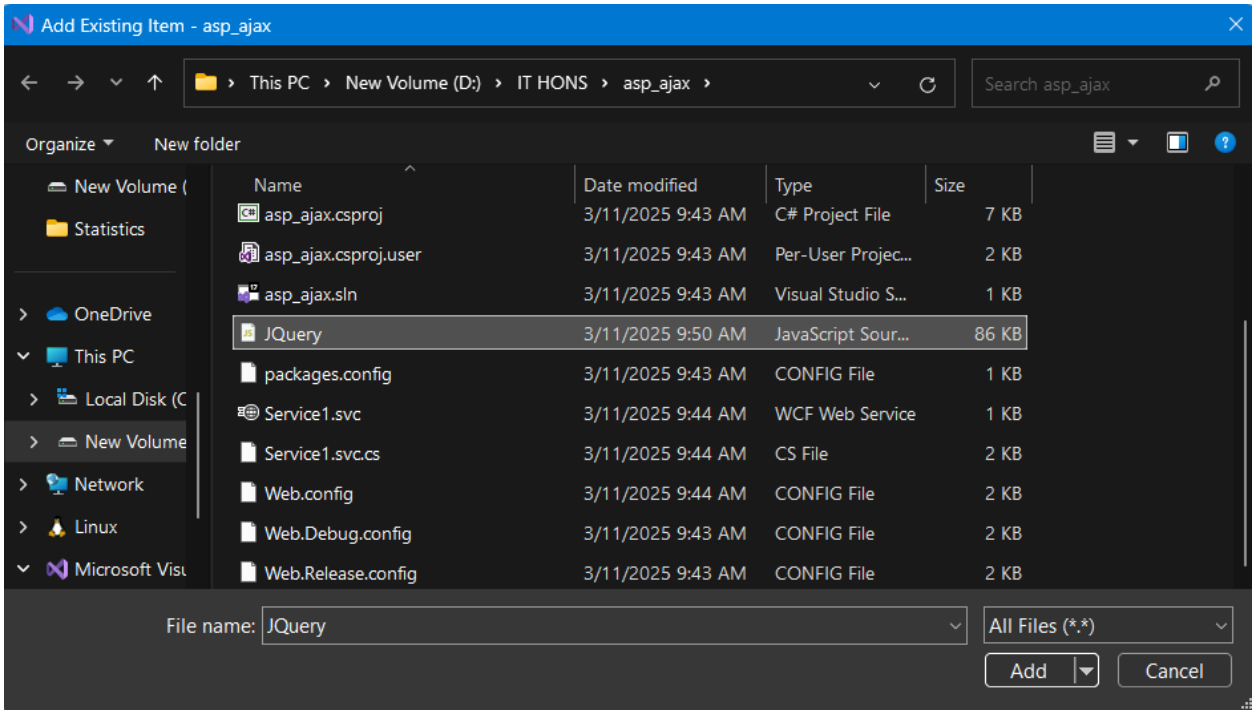


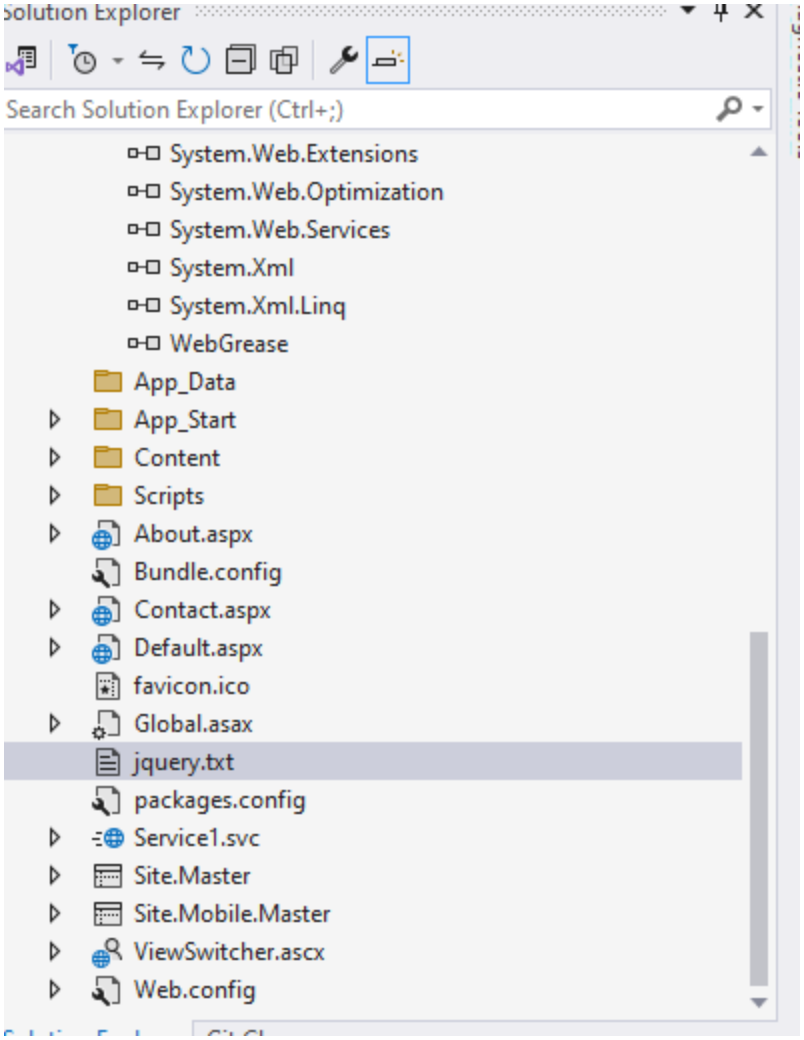
Save jQuery from notepad

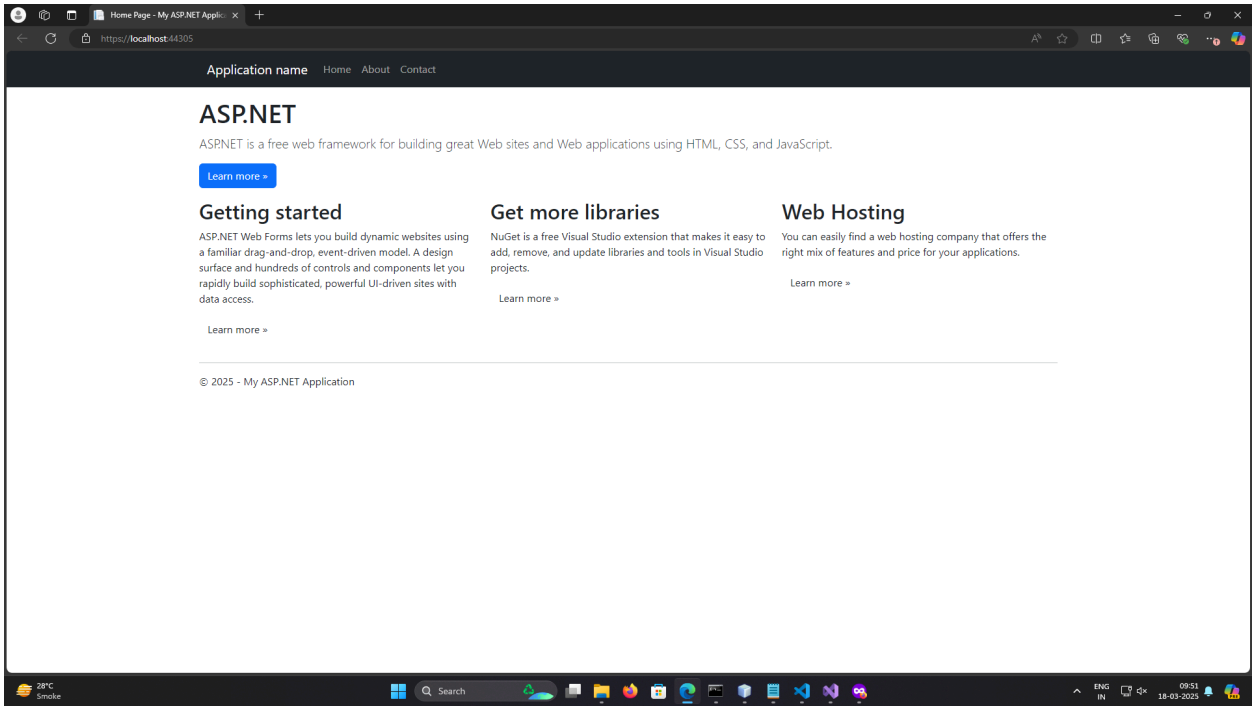

https://jquery.com/download/
Plugins
Contribute
Events
Support
OpenJS Foundation
jQuery
write less, do more.
Download
API Documentation
Blog
Plugins
Browser Support
Version Support
Search
Download jQuery
Latest version
To locally download these files, right-click the link and select "Save as..." from the menu.
Download the compressed, production version:
Download jQuery 3.7.1
Download the uncompressed development version of jQuery 3.7.1
Download the map file for jQuery 3.7.1
jQuery 3.7.1 blog post with release notes
The slim build is a smaller version, that excludes the ajax and effects modules:
Download jQuery 3.7.1 slim build
Download the uncompressed development version of the jQuery 3.7.1 slim build
Download the map for the jQuery 3.7.1 slim build


JQuery.js
File
Edit
View

Then load it :

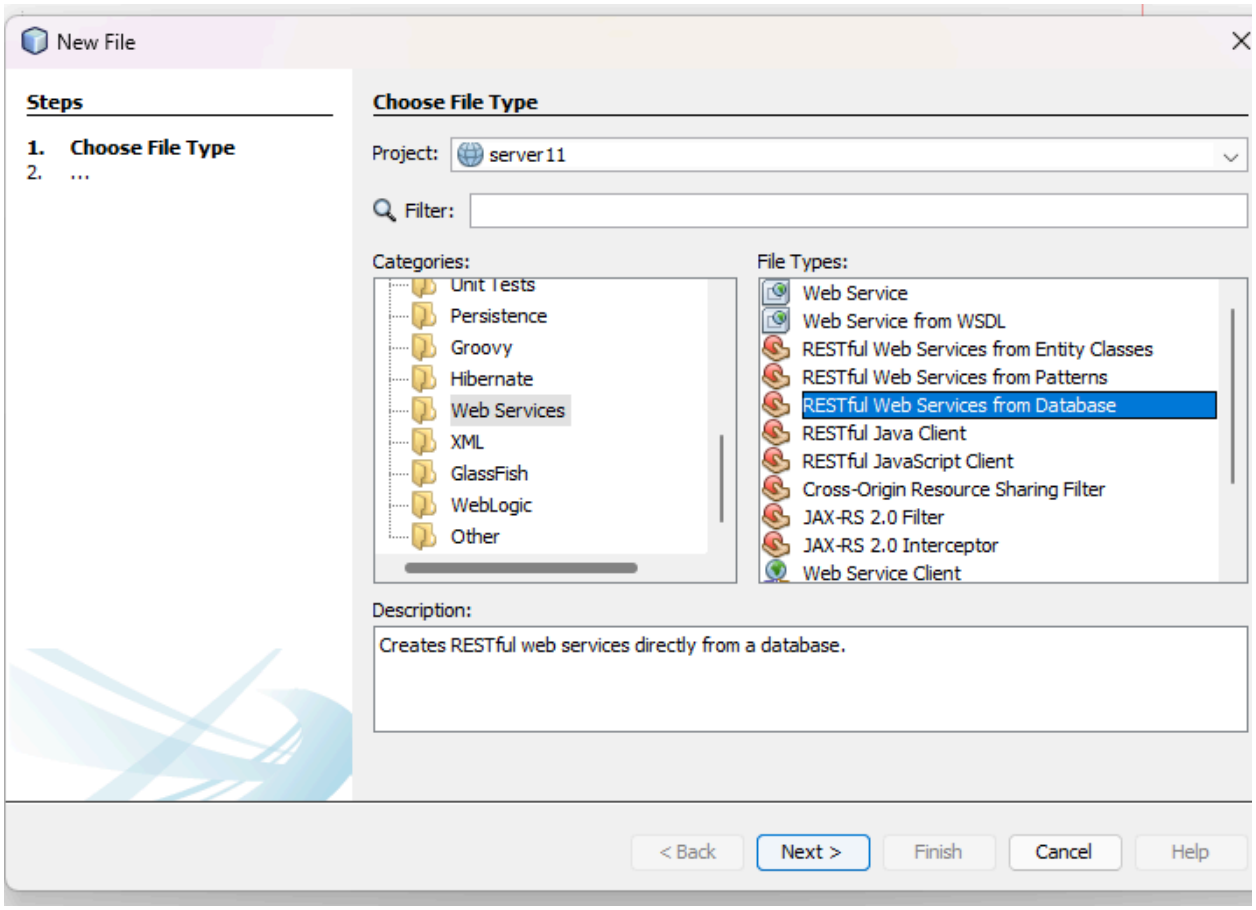

Add Existing Item - asp_ajax
This PC › New Volume (D:) › IT HONS › asp_ajax ›
Search asp_ajax
Organize
New folder
New Volume (
Statistics
OneDrive
This PC
Local Disk (C
New Volume
Network
Linux
Microsoft Visu
Name
Date modified
Type
Size
asp_ajax.csproj 3/11/2025 9:43 AM C# Project File 7 KB
asp_ajax.csproj.user 3/11/2025 9:43 AM Per-User Projec... 2 KB
asp_ajax.sln 3/11/2025 9:43 AM Visual Studio S... 1 KB
JQuery 3/11/2025 9:50 AM JavaScript Sour... 86 KB
packages.config 3/11/2025 9:43 AM CONFIG File 1 KB
Service1.svc 3/11/2025 9:44 AM WCF Web Service 1 KB
Service1.svc.cs 3/11/2025 9:44 AM CS File 2 KB
Web.config 3/11/2025 9:44 AM CONFIG File 2 KB
Web.Debug.config 3/11/2025 9:43 AM CONFIG File 2 KB
Web.Release.config 3/11/2025 9:43 AM CONFIG File 2 KB
File name: JQuery
All Files (*.*)
Add
Cancel

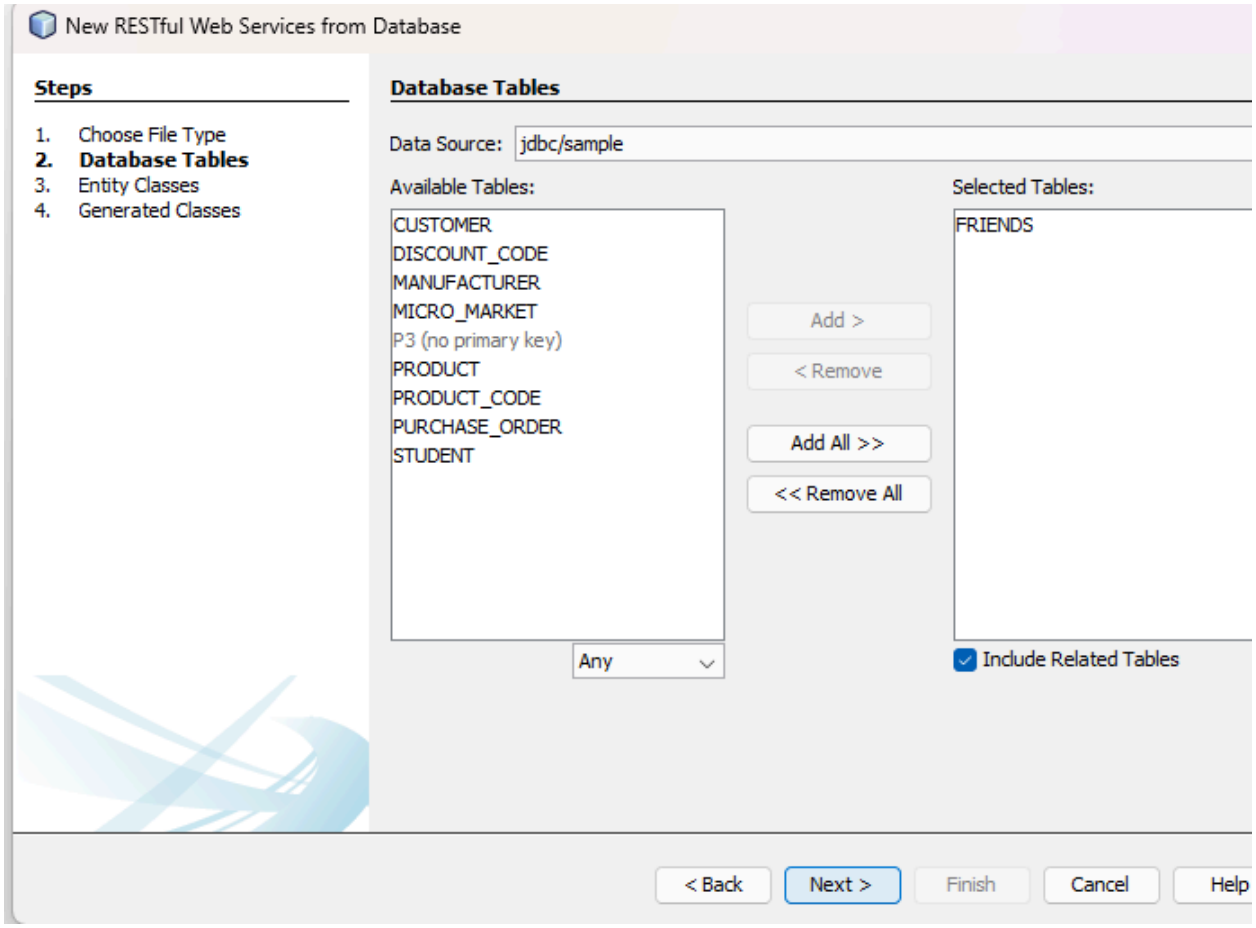

Solution Explorer
Search Solution Explorer (Ctrl+;)
System.Web.Extensions
System.Web.Optimization
System.Web.Services
System.Xml
System.Xml.Linq
WebGrease
App_Data
App_Start
Content
Scripts
About.aspx
Bundle.config
Contact.aspx
Default.aspx
favicon.ico
Global.asax
jquery.txt
packages.config
Service1.svc
Site.Master
Site.Mobile.Master
ViewSwitcher.ascx
Web.config

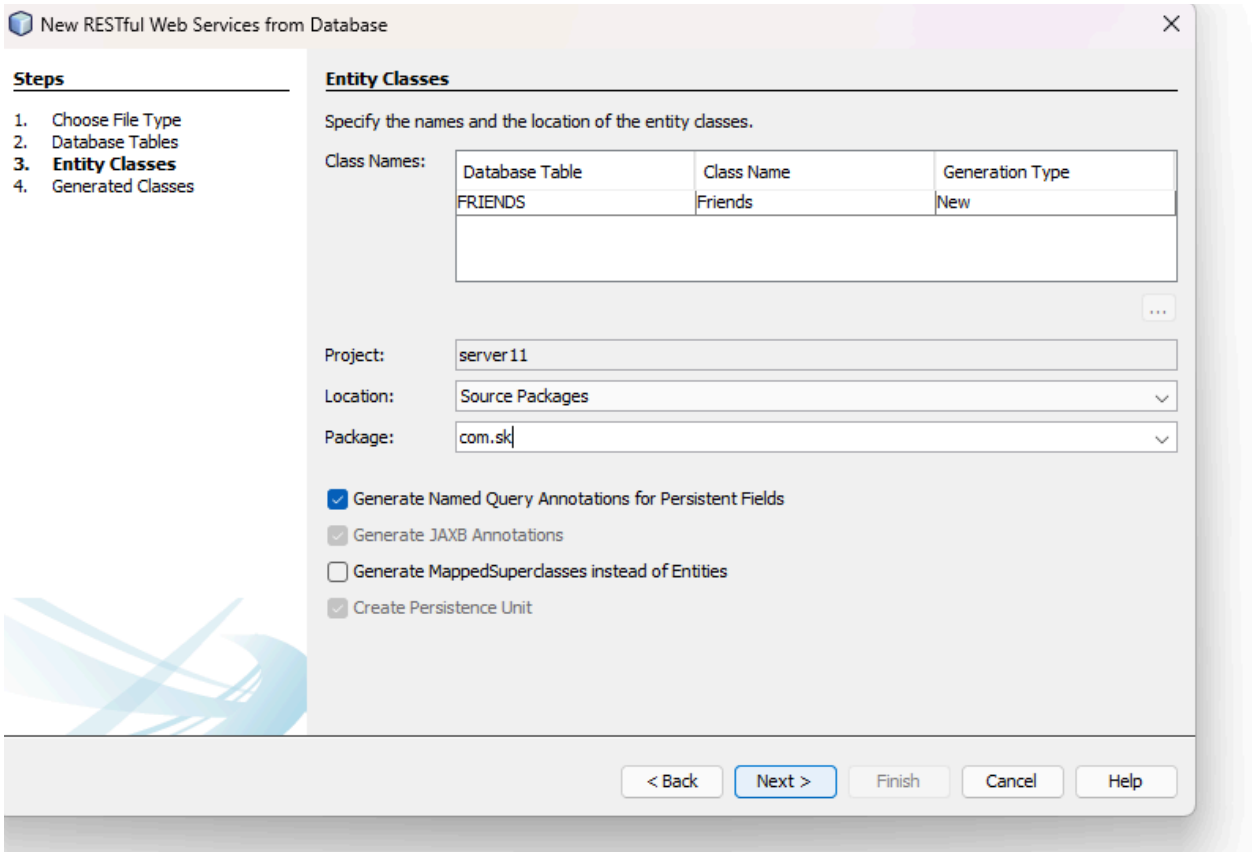


Practical 7

Use the friends table form the database
Create a new project then select the restful web services for database
Then import the table
Create another project named client11


New File
Steps
1. Choose File Type
2. ...
Choose File Type
Project:
server11
Filter:
Categories:
Unit Tests
Persistence
Groovy
Hibernate
Web Services
XML
GlassFish
WebLogic
Other
File Types:
Web Service
Web Service from WSDL
RESTful Web Services from Entity Classes
RESTful Web Services from Patterns
RESTful Web Services from Database
RESTful Java Client
RESTful JavaScript Client
Cross-Origin Resource Sharing Filter
JAX-RS 2.0 Filter
JAX-RS 2.0 Interceptor
Web Service Client
Description:
Creates RESTful web services directly from a database.
< Back
Next >
Finish
Cancel
Help


New RESTful Web Services from Database
Steps
1. Choose File Type
2. Database Tables
3. Entity Classes
4. Generated Classes
Database Tables
Data Source: jdbc/sample
Available Tables:
CUSTOMER
DISCOUNT_CODE
MANUFACTURER
MICRO_MARKET
P3 (no primary key)
PRODUCT
PRODUCT_CODE
PURCHASE_ORDER
STUDENT
Add >
< Remove
Add All >>
<< Remove All
Selected Tables:
FRIENDS
Any
Include Related Tables
< Back
Next >
Finish
Cancel
Help



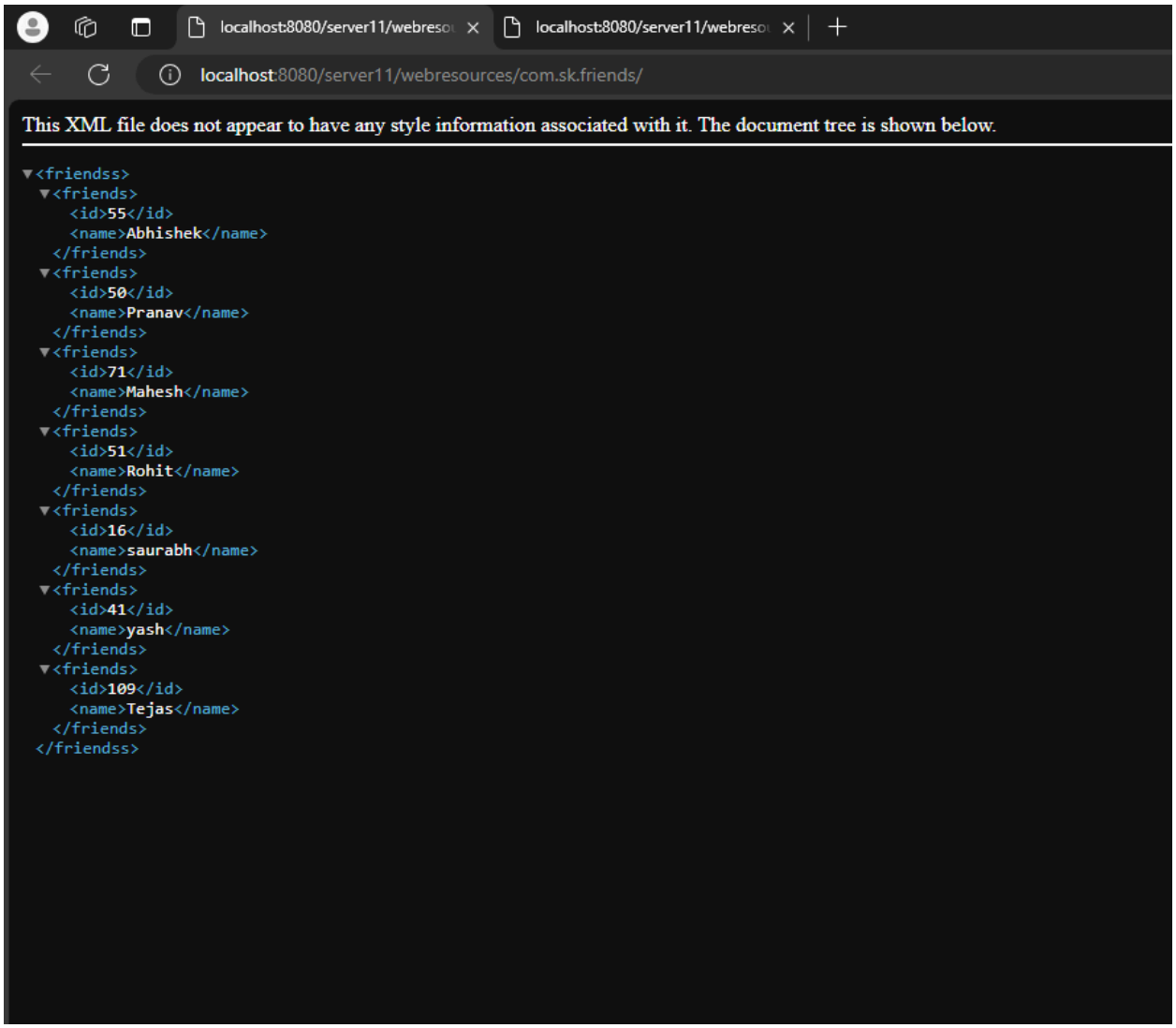
Code is given in the classroom paste it in the client11 index page and run it

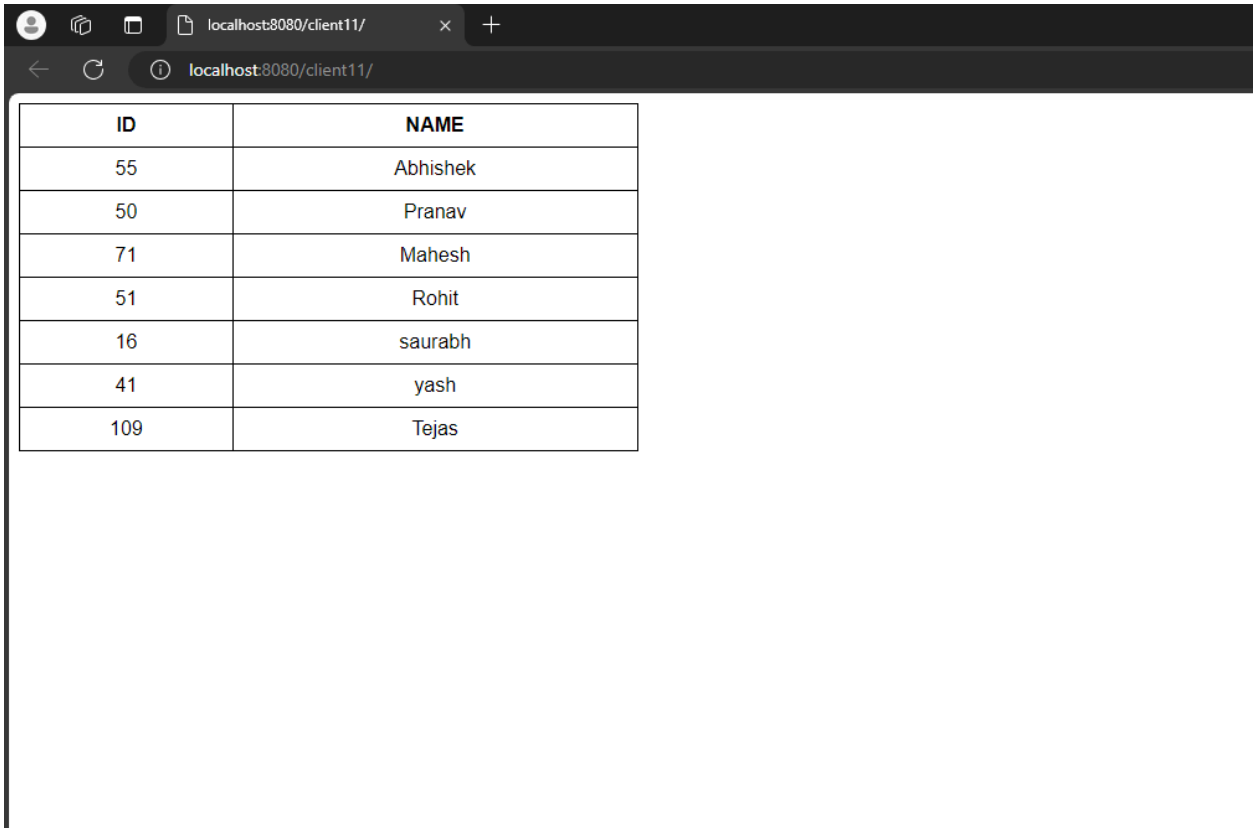
```
<!DOCTYPE html>
<html>
<head>
        <style>
        table {
        font-family: Arial, sans-serif;
        border-collapse: collapse;
        width: 50%;
        }
        td, th {
        border: 1px solid #000000;
        text-align: center;
        padding: 8px;
        }
        </style>
        <script>
        console.log('hello')

        document.addEventListener("DOMContentLoaded", function () {
        fetch('http://localhost:8080/server11/webresources/com.sk.friends/')
        .then(response => {
```

```
        //console.log("1: Fetch initiated");

        if (!response.ok) {
        throw new Error('HTTP error! Status: ' + response.status);
        }
        return response.text();  // Read as text first to debug
    })
    .then(text => {
        console.log("Raw Response:", text);  // Debugging response

        let parser = new DOMParser();
        let xmlDoc = parser.parseFromString(text, "text/xml");

        // Convert the XML data to a JSON-like structure
        let friends = Array.from(xmlDoc.getElementsByTagName("friends")).map(friend
=> {
        return {
        id: friend.getElementsByTagName("id")[0].textContent,
        name: friend.getElementsByTagName("name")[0].textContent
        };
        });

        console.log("Converted JSON Data:", friends); // Debugging JSON output

        // Proceed with your previous code for rendering the table
        let table = document.getElementById("myTable");

        // Clear previous rows (except headers)
        table.innerHTML = "<tr><th>ID</th><th>NAME</th></tr>";

        if (Array.isArray(friends)) {
        friends.forEach(friend => {
        let row = table.insertRow();
        let cell1 = row.insertCell(0);
        let cell2 = row.insertCell(1);

        cell1.innerHTML = friend.id;
        cell2.innerHTML = friend.name;
        });
        } else {
        console.error("Unexpected response format. Expected an array.");
        }
    })
    .catch(error => console.error("Fetch error:", error));
```

```
        });
        </script>
</head>
<body>
        <table id="myTable">
        <tr>
        <th>ID</th>
        <th>NAME</th>
        </tr>
        </table>
</body>
</html>
```

Output:-





ID	NAME
55	Abhishek
50	Pranav
71	Mahesh
51	Rohit
16	saurabh
41	yash
109	Tejas