Practical-1

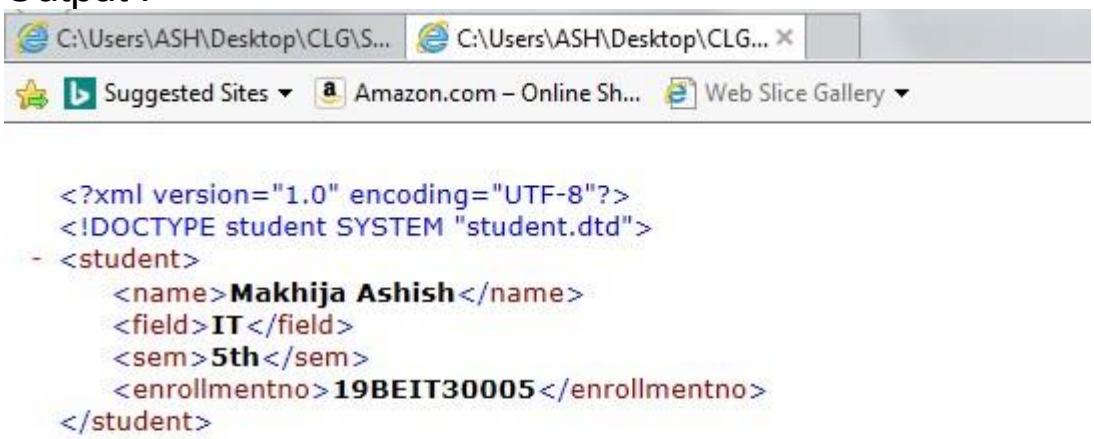
Practical-1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE student SYSTEM "student.dtd" >
<student>
 <name>Makhija Ashish</name>
 <field>IT</field>
 <sem>5th</sem>
 <enrollmentno>19BEIT30005</enrollmentno>  </student>
```

student.dtd

```
<!DOCTYPE student[
 <!ELEMENT student (name,field,sem,enrollmentno) >

<!ELEMENT name (#PCDATA)>

 <!ELEMENT field (#PCDATA)>
 <!ELEMENT sem (#PCDATA)>
 <!ELEMENT enrollmentno (#PCDATA)> ]>
```

Output :



Practical 2

Aim : Create XSD file for student information and create a valid well formed XML document to store student information against this XSD file.

P2.xml

```
<?xml version = "1.0"?>
<Student>
xmlns:xsi = "http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation = "P2.xsd"
<Name>Ashish</Name>
<First-Name>Makhija</First-Name>
<En.No.>19BEIT30005</En.No.>
<Class>BE-IT</Class>
<Sem>5th</Sem>
<Collage>LDRP-ITR</Collage>
<Email>ashish123@gmail.com</Email>
<Mobile-no.>123456789</Mobile-no.>
</Student>
```

P2.XSD

```
<?xml version = "1.0"?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"> <xs:element
name = "Student">
<xs:complexType>
<xs:sequence>
<xs:element name = "Name" type = "xs:string"/>
<xs:element name = "First-Name" type = "xs:string"/>
<xs:element name = "En.No." type = "xs:varchar"/>
<xs:element name = "Class" type = "xs:string"/>
<xs:element name = "Sem" type = "xs:varchar"/>
<xs:element name = "Collage" type = "xs:string"/>
<xs:element name = "Email" type = "xs:varchar"/>
```

```
<xs:element name = "Mobile-no." type = "xs:integer"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

OUTPUT

Practical 3

XML File:

```
<?xml version="1.0"
encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="table(p3).xsl"?>

<data>

<StudentData>
<name>Patel Deep</name>
<enroll>19BEIT30010</enroll>
<class>5th IT</class>
<email>deep123@gmail.com</email>
</StudentData>
<StudentData>
<name>Dhruv Patel</name>
<enroll>19BEIT30014</enroll>
<class>5th IT</class>
<email>dhruvp1@gmail.com</email>
</StudentData>
<StudentData>

<name>Patel Deep</name>
<enroll>19SBEIT30012</enroll>
<class>5th IT</class>
<email>dp12p@gmail.com</email>
</StudentData>
<StudentData>
<name>Makhija Ashish</name>
<enroll>19BEIT30005</enroll>
<class>5th IT</class>
<email>ashish123@gmail.com</email>
</StudentData>
</data>
```

table(p3).xsl

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>
<body>
<h3>Student Information</h3>
<table border="1">
<tr bgcolor="YELLOW">

<th style="text-align:center">Name</th>
<th style="text-align:center">Enrollment Number</th>

<th style="text-align:center">Class</th>

<th style="text-align:center">Email</th>
</tr>
<xsl:for-each select ="data/StudentData">

<tr>

<td><xsl:value-of select="name"/></td>

<td><xsl:value-of select="enroll"/></td>
<td><xsl:value-of select="class"/></td>
<td><xsl:value-of select="email"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

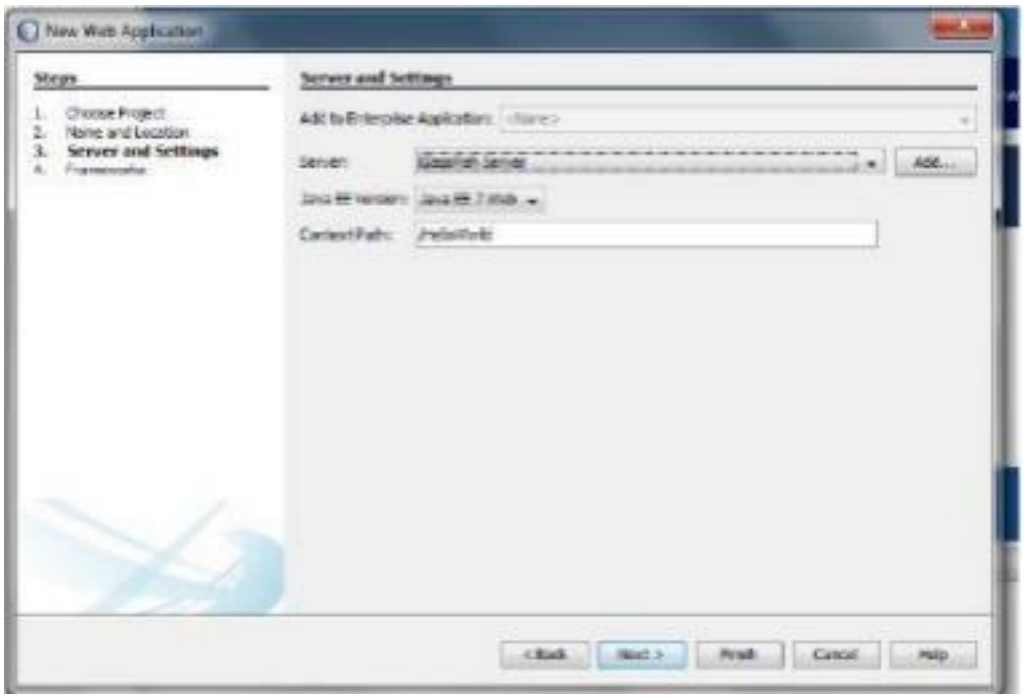
OUTPUT :

Student Information

Name	Enrollment Number	Class	Email
Patel Deep	19BEIT30010	5th IT	deep123@gmail.com
Dhruv Patel	19BEIT30014	5th IT	dhruvp1@gmail.com
Patel Deep	19SBEIT30012	5th IT	dp12p@gmail.com
Makhija Ashish	19BEIT30005	5th IT	ashish123@gmail.com

PRACTICAL 3

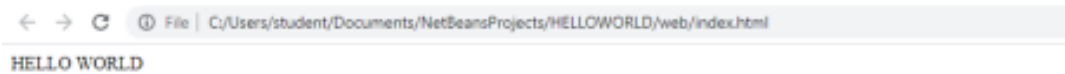
Aim **:** Create a hello world application using netbeans.



CODE:

```
<html>
<head>
<title>Hello World Title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0"> </head>
<body>
<div>HELLO WORLD</div>
</body>
</html>
```

OUTPUT:



HELLO WORLD

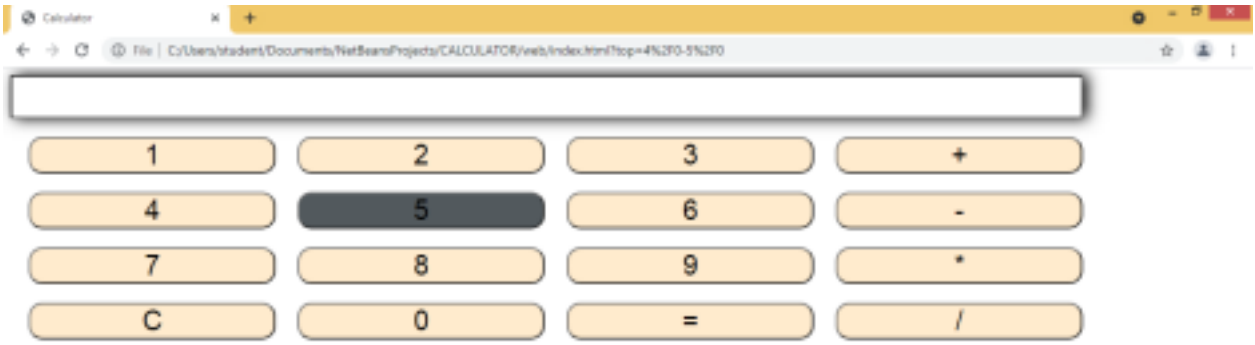
PRACTICAL 5

CODE:

```
<html>
        <head>
        <title>Calculator</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
        input[type=text]
         {
        position:relative;
        width:85%;
        margin:top;
        font-size:20px;
        padding:10px;
        box-shadow:5px 0px 15px black;

         }
        input[type=button]{

        background-color: blanchedalmond;
        width:20%;
        font-size: 30px;
        font-weight: 500;
        border-radius: 15px;
        margin-left: 20px;
        margin-top:20px;
         }
        input[type=button]:hover{
        background-color: #52595D;
        box-shadow: yellow;


        }
        </style>
        </head>
        <body>
        <form name="Calc">
        <input type="text" name="top"/>
        <br>
        <input type="button" value="1" onclick="Calc.top.value+='1'"/>
        <input type="button" value="2" onclick="Calc.top.value+='2'"/>
        <input type="button" value="3" onclick="Calc.top.value+='3'"/>
```

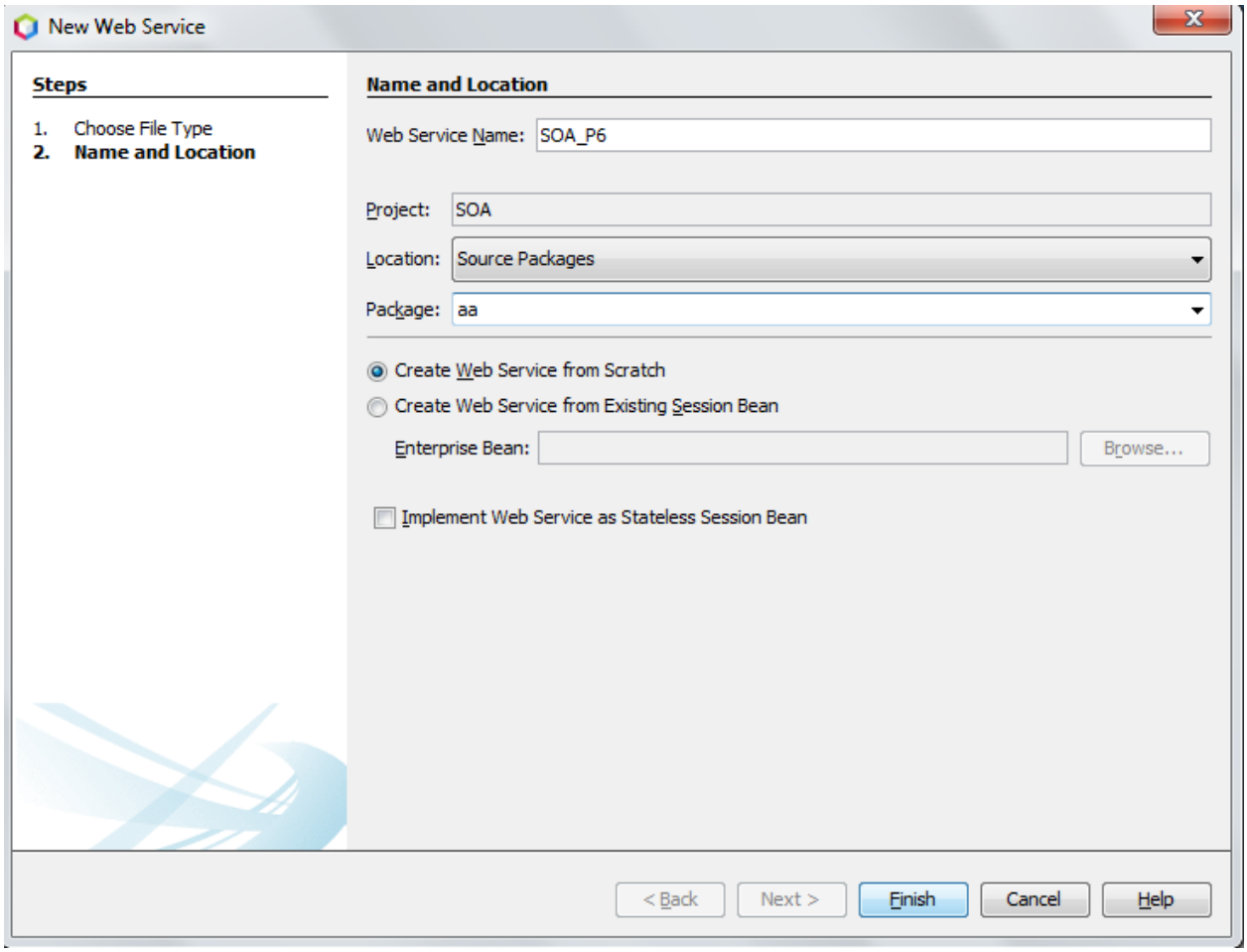
```
<input type="button" value="+" onclick="Calc.top.value+='+'"/>
<br>
<input type="button" value="4" onclick="Calc.top.value+='4'"/>
<input type="button" value="5" onclick="Calc.top.value+='5'"/>
<input type="button" value="6" onclick="Calc.top.value+='6'"/>
<input type="button" value="-" onclick="Calc.top.value+='-'"/>
<br>
<input type="button" value="7" onclick="Calc.top.value+='7'"/>
<input type="button" value="8" onclick="Calc.top.value+='8'"/>
<input type="button" value="9" onclick="Calc.top.value+='9'"/>
<input type="button" value="*" onclick="Calc.top.value+='*'"/>
<br>

<input type="button" value="C" onclick="Calc.top.value=''"/>
<input type="button" value="0" onclick="Calc.top.value+='0'"/>
<input type="button" value="=" onclick="Calc.top.value=eval(Calc.top.value)"/>
<input type="button" value="/" onclick="Calc.top.value+='/'"/>
<br>
</form>
</body>
</html>
```

Output:



Practical – 6

Aim : Create a Hello Web Services using netbeans:



☐ Code for Hello Web Services:

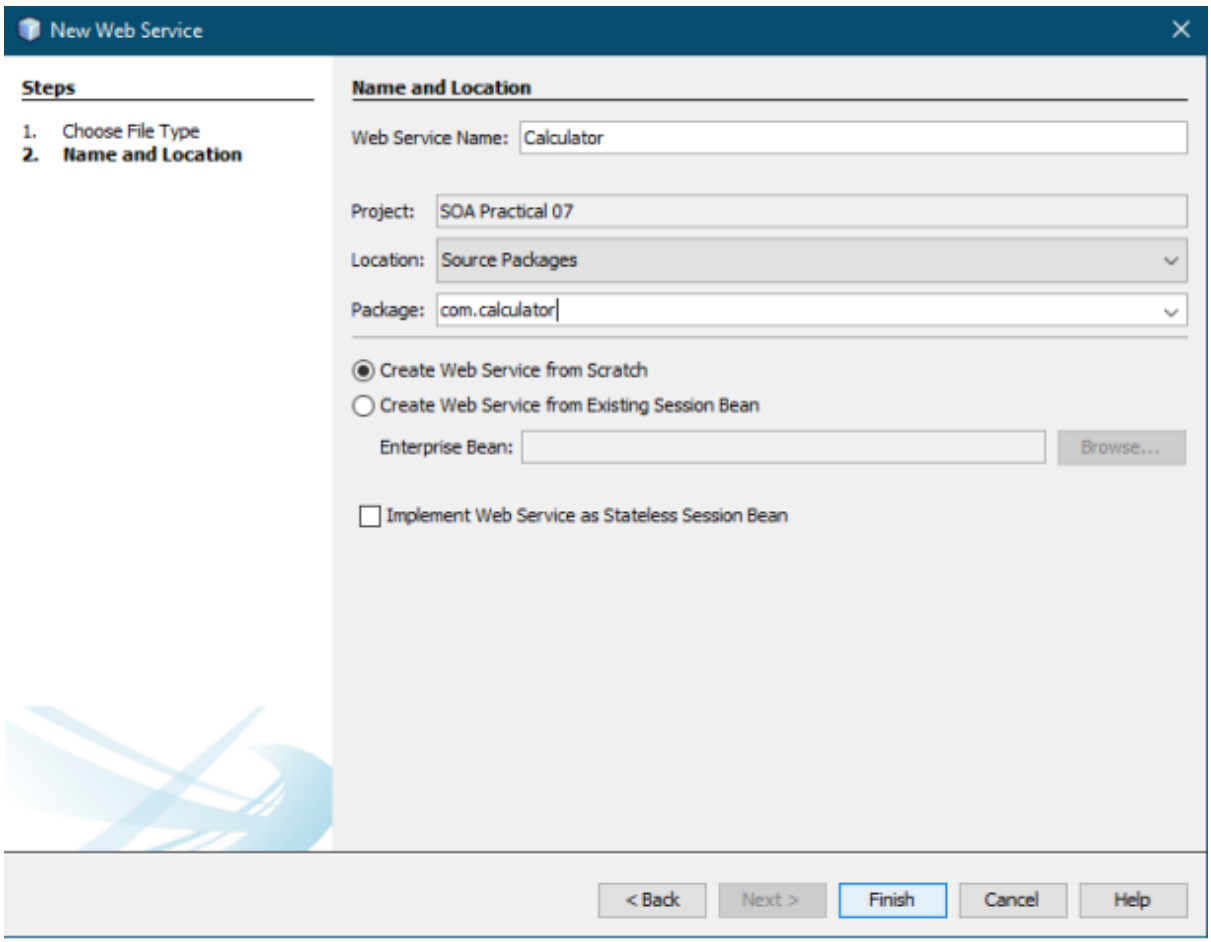
```
package aa;

import javax.jws.WebService;

import javax.jws.WebMethod;
```

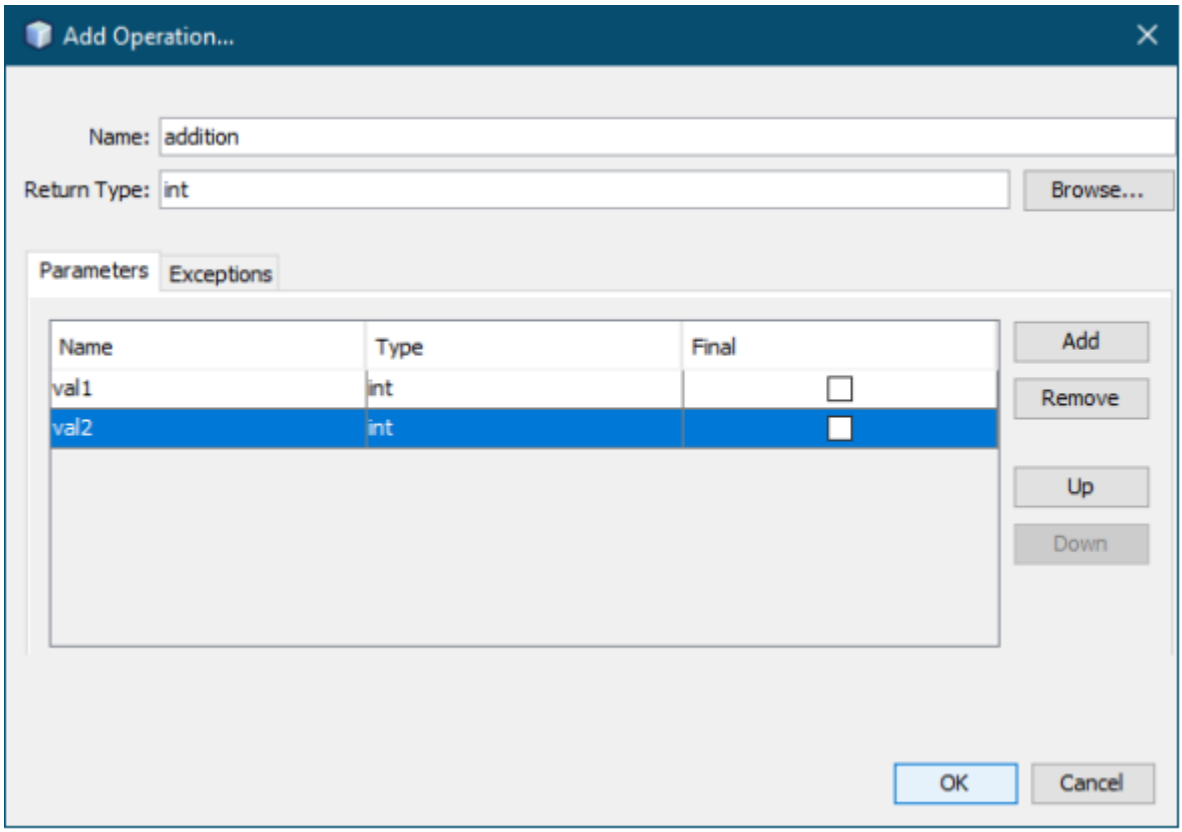
```
import javax.jws.WebParam;

@WebService(serviceName = "SOA_P6")

public class SOA_P6 {



    @WebMethod(operationName = "hello")

    public String hello(@WebParam(name = "name") String txt) {

        return "Hello " + txt + " !";

    }
```

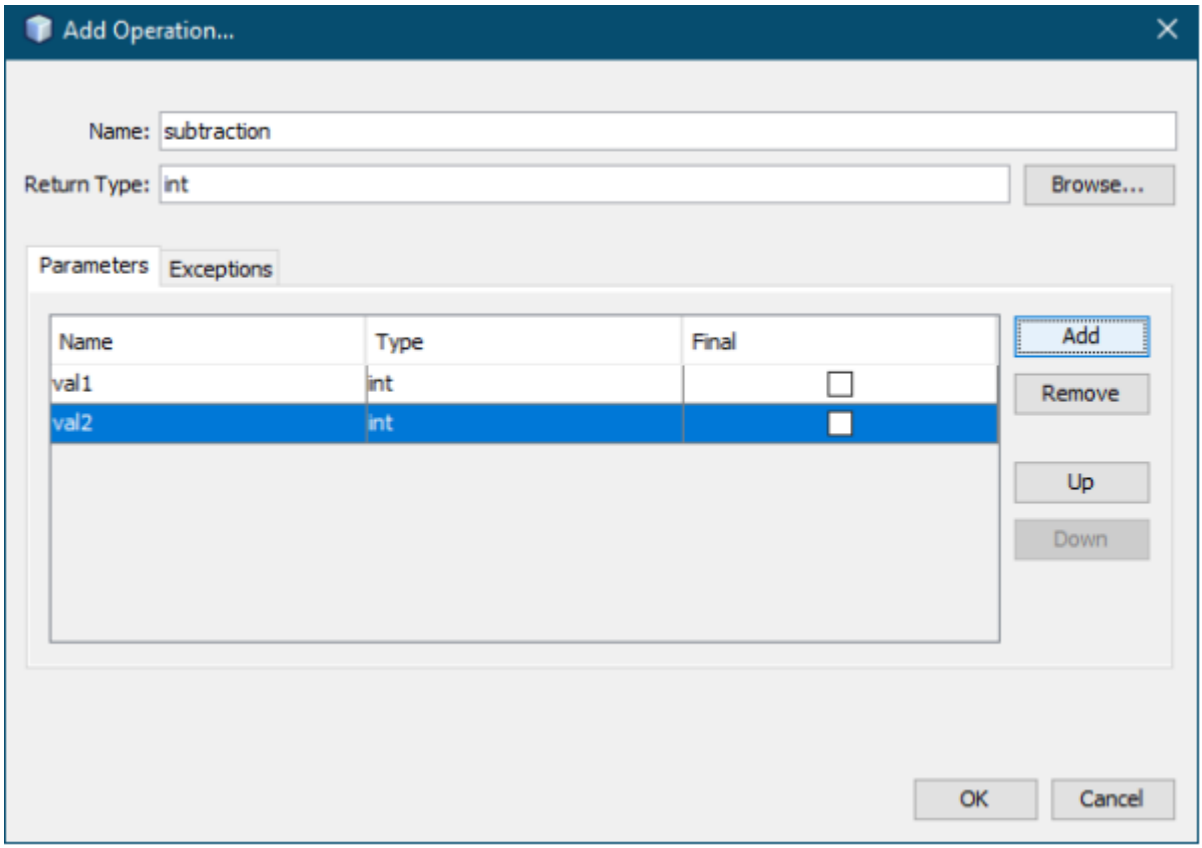
} ***Output***:

Practical -7

Aim : Create a calculator web service using netbeans





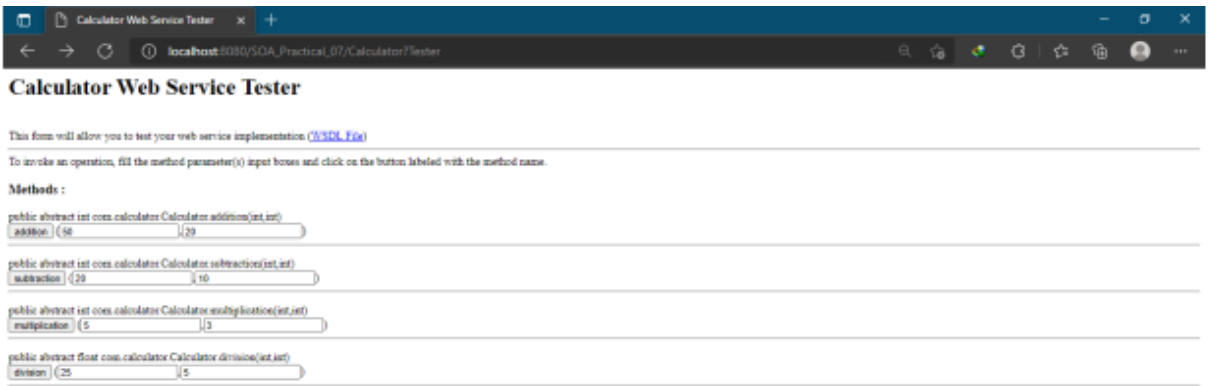


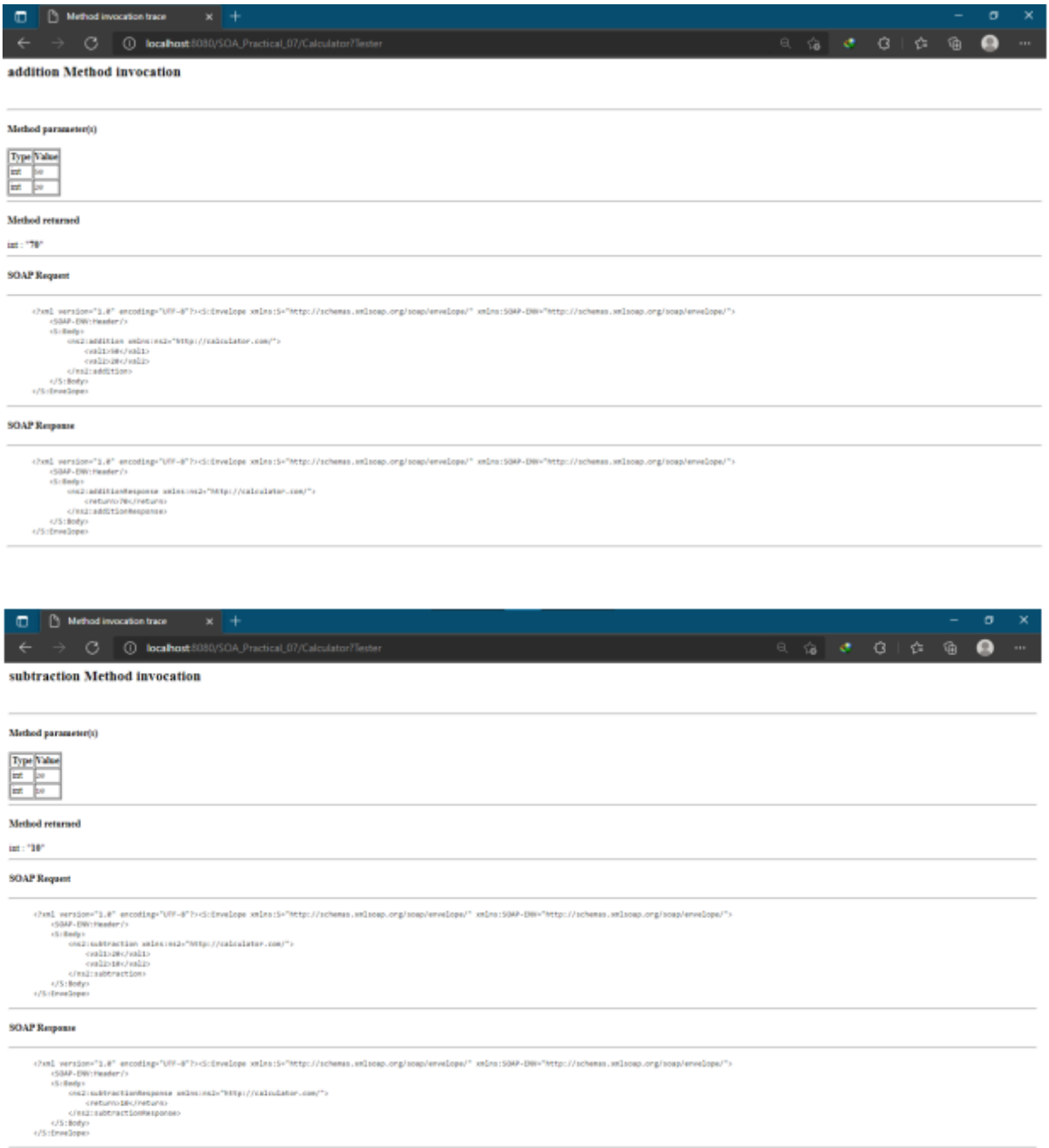
Calculator Web Services

```
package Calculator;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.ejb.Stateless;
@WebService(serviceName = "Calculator")
@Stateless()
public class Calculator {
 /**
 * This is a sample web service operation
 */
 /*@WebMethod(operationName = "hello")
 public String hello(@WebParam(name = "name") String txt) {  return
"Hello " + txt + " !";

 }*/
 @WebMethod(operationName = "ADDITION")
 public int ADDITION(@WebParam(name = "Value1") int Value1, @WebParam(name =  "Value2")
int Value2) {
 //TODO write your implementation code here:
 return (Value1 + Value2);
 }
 /**
 * Web service operation
 */
 @WebMethod(operationName = "SUBTRACTION")
 public int SUBTRACTION(@WebParam(name = "Value1") int Value1,
@WebParam(name = "Value2") int Value2) {
 //TODO write your implementation code here:
 return (Value1 - Value2);
 }
 /**
 * Web service operation
 */
 @WebMethod(operationName = "MULTIPLICATION")
 public int MULTIPLICATION(@WebParam(name = "Value1") int Value1,
@WebParam(name = "Value2") int Value2) {
 //TODO write your implementation code here:
 return (Value1 * Value2);
```

```
}



/**
* Web service operation
*/
@WebMethod(operationName = "DIVISION")
public int DIVISION(@WebParam(name = "Value1") int Value1, @WebParam(name =  "Value2") int Value2) {
//TODO write your implementation code here:
return (Value1 / Value2);
}
/**
* Web service operation
*/

}
```

OUTPUT




Method invocation trace
localhost:8080/SOA_Practical_07/Calculator?Tester
addition Method invocation
Method parameter(s)
Type Value
int 50
int 20
Method returned
int : "70"
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:addition xmlns:ns2="http://calculator.com/">
<val1>50</val1>
<val2>20</val2>
</ns2:addition>
</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:additionResponse xmlns:ns2="http://calculator.com/">
<return>70</return>
</ns2:additionResponse>
</S:Body>
</S:Envelope>

Method invocation trace
localhost:8080/SOA_Practical_07/Calculator?Tester
subtraction Method invocation
Method parameter(s)
Type Value
int 30
int 20
Method returned
int : "10"
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:subtraction xmlns:ns2="http://calculator.com/">
<val1>20</val1>
<val2>10</val2>
</ns2:subtraction>
</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:subtractionResponse xmlns:ns2="http://calculator.com/">
<return>10</return>
</ns2:subtractionResponse>
</S:Body>
</S:Envelope>

Method invocation trace

localhost:8080/SOA_Practical_07/Calculator?tester

multiplication Method invocation

Method parameter(s)

Type	Value
int	5
int	3

Method returned

int : "15"

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:multiplication xmlns:ns2="http://calculator.com/">
            <val1>5</val1>
            <val2>3</val2>
        </ns2:multiplication>
    </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:multiplicationResponse xmlns:ns2="http://calculator.com/">
            <return>15</return>
        </ns2:multiplicationResponse>
    </S:Body>
</S:Envelope>
```


Method invocation trace

localhost:8080/SOA_Practical_07/Calculator?tester

division Method invocation

Method parameter(s)

Type	Value
int	25
int	5

Method returned

float : "5.0"

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:division xmlns:ns2="http://calculator.com/">
            <val1>25</val1>
            <val2>5</val2>
        </ns2:division>
    </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:divisionResponse xmlns:ns2="http://calculator.com/">
            <return>5.0</return>
        </ns2:divisionResponse>
    </S:Body>
</S:Envelope>
```

Practical – 8

Aim : To create Restful Web Services Code for creating Restful Web service

index.html

```
<html>
 <head>
 <title></title>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width,  initial-scale=1.0">
 </head>
 <body>
 <div></div>
 </body>
</html>
```

ApplicationConfig.java

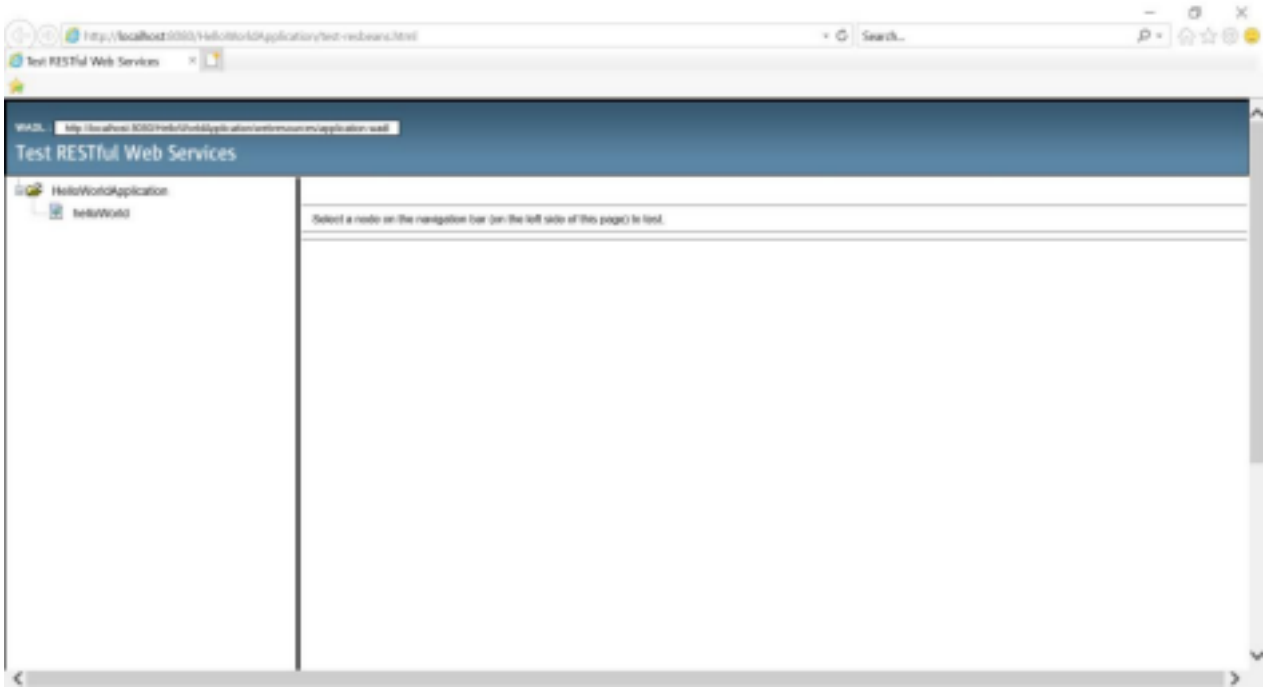
```
package helloWorld;
import java.util.Set;
import javax.ws.rs.core.Application;
@javax.ws.rs.ApplicationPath("webresources")  public class ApplicationConfig
extends Application {   @Override
 public Set<Class<?>> getClasses() {
 Set<Class<?>> resources = new java.util.HashSet<>();
addRestResourceClasses(resources);
 return resources;
 }
 private void addRestResourceClasses(Set<Class<?>>  resources) {
 resources.add(helloWorld.HelloWorld.class);   }
}
```

HelloWorld.java

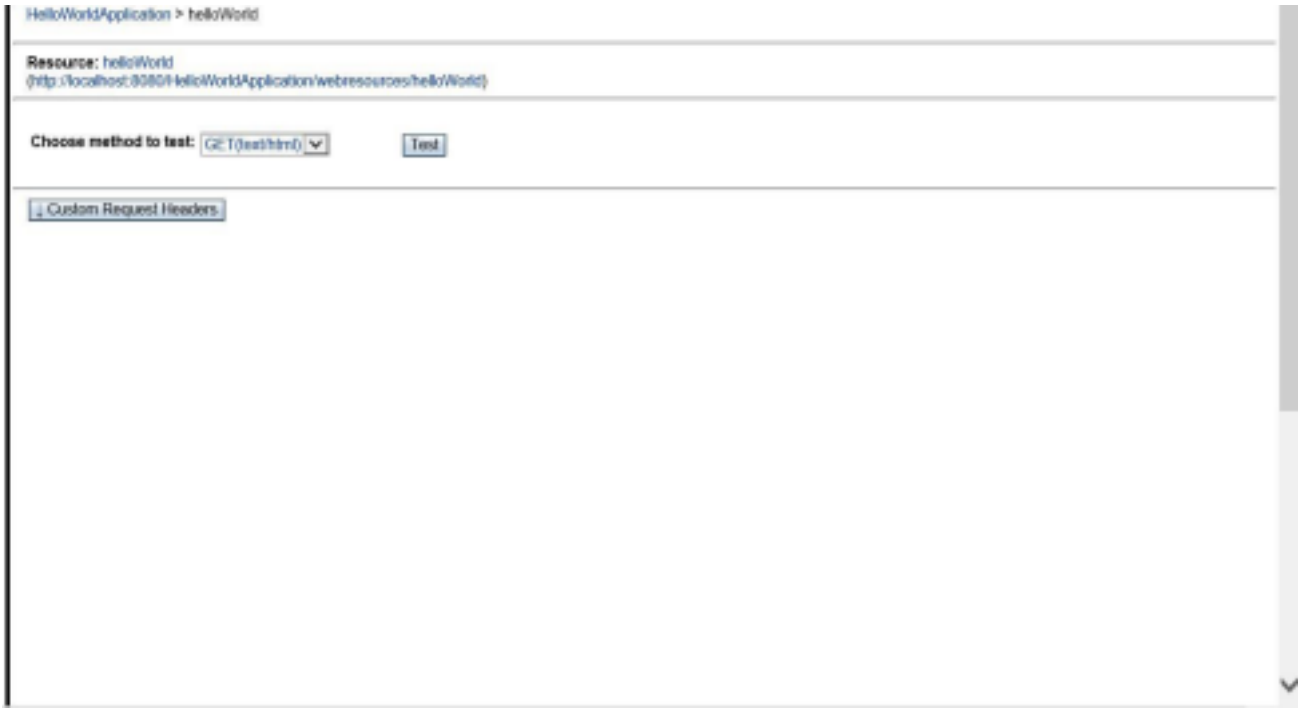
```
package helloWorld;
import javax.ws.rs.core.Context;
```

```
 import javax.ws.rs.core.UriInfo;
import javax.ws.rs.Produces;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PUT;
import javax.ws.rs.core.MediaType;
@Path("helloworld")
public class HelloWorld {
 @Context
 private UriInfo context;
 public HelloWorld() {
 }
@GET
 @Produces(MediaType.TEXT_HTML)   public String getHtml() {
  return "<html><body><h1>Hello  World.</h1></body></html>";
 }
 @PUT
 @Consumes(MediaType.TEXT_HTML)
 public void putHtml(String content) {    }
}
```

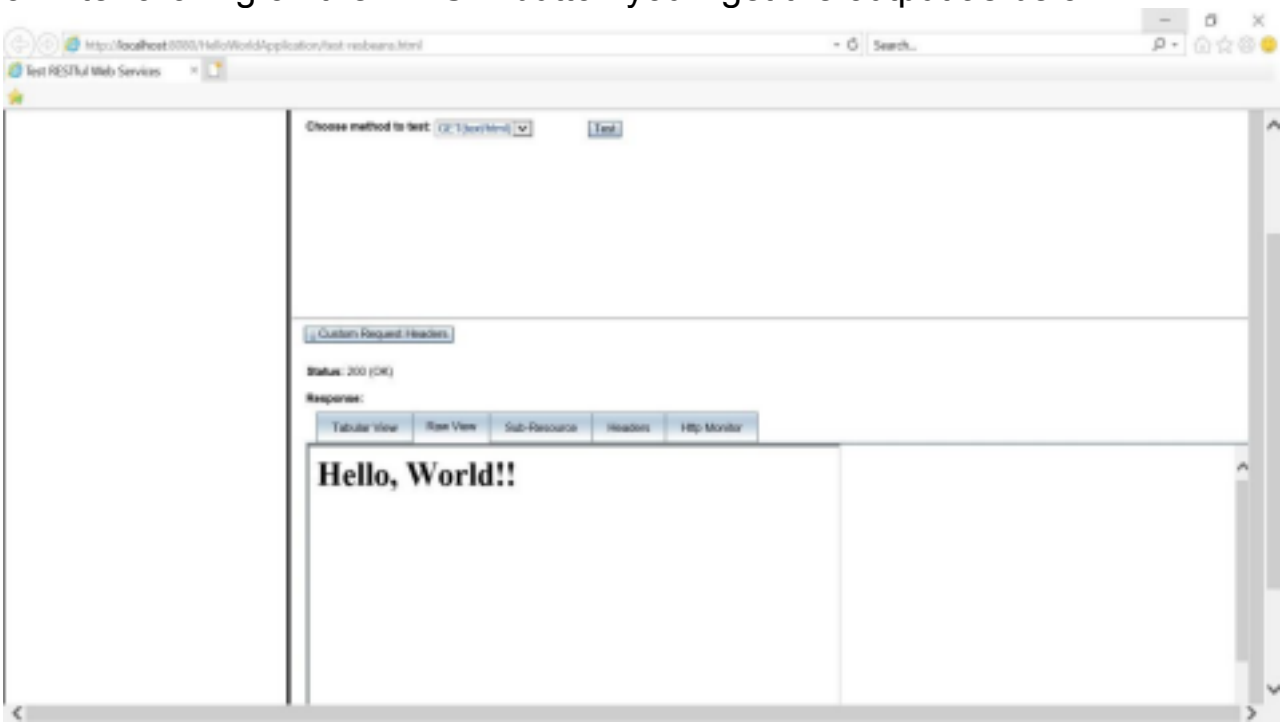
Output : 1. In the following snapshot, your “HELLOWORLDAPPLICATION” name is shown. Then click on “helloworld” i.e. your restful web service.



2. After clicking on the “helloworld” you’ll get two methods i.e. GET method and PUT method. But we have to select the GET method and then click on the “TEST” button.



3. After clicking on the “TEST” button you’ll get the output as below:



PRACTICAL : 9

AIM : INTRODUCTION TO AMAZON WEB SERVICES(AWS)

- **AMAZON WEB SERVICES(AWS):** Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform.
  - It offers over 175 fully featured services from data centers globally.
  - It has millions of customers connected with AWS It includes the fastest-growing start-ups , largest enterprises, and leading government agencies.
  - These agencies are using AWS to lower costs, become more agile, and innovate faster.

- **Functionality of AWS:**
  - AWS has significantly more services, and more features within those services, than any other cloud provider.
  - It develops from infrastructure technologies like compute, storage, and databases–to emerging
  - technologies, such as machine learning and artificial intelligence, data lakes and analytics, and Internet of Things. This makes it faster, easier, and more cost effective to move your existing

applications to the cloud and build nearly anything you can imagine.

- AWS also has the deepest functionality within those services. For example, AWS offers the widest variety of databases that are purpose-built for different types of applications so you can choose the right tool for the job to get the best cost and performance.

FEATURES OF AWS:

- Largest community of customers and partners:

AWS has the largest and most dynamic community, with millions of active customers and tens of thousands of partners globally. Customers across virtually every industry and of every size, including startups, enterprises, and public sector organizations, are running every imaginable use case on AWS.

- **Most secure**

AWS is architected to be the most flexible and secure cloud computing environment available today. Its core infrastructure is built to satisfy the security requirements for the military, global banks, and other high-sensitivity organizations. This is backed by a deep set of cloud security tools, with 230 security, compliance, and governance services and features.

- **Fastest pace of innovation**

With AWS, we can leverage the latest technologies to experiment and innovate more quickly. They are continually accelerating their pace of innovation to invent entirely new technologies , we can use to transform our business. For example, in 2014, AWS pioneered the serverless computing space with the launch of AWS Lambda, which lets developers run their code without provisioning or managing servers. And AWS built Amazon SageMaker, a fully managed machine learning service that empowers everyday developers and scientists to use machine learning, without any previous experience.

- Most proven operational expertise

AWS has unmatched experience, maturity, reliability, security, and performance that you can depend upon for your most important applications. For over 14 years, AWS has been delivering cloud services to millions of customers around the world running a wide variety of use cases. AWS has the most operational experience, at greater scale, of any cloud provider.

❖ SOME USES OF AWS:

- Adobe uses AWS to provide multi-terabyte operating environments for its customers by integrating its system with AWS Cloud. Adobe can focus on deploying and operating its own software instead of trying to deploy and manage the infrastructure.
- Airbnb, the online vacation rental marketplace for property owners and travelers to connect, maintains a huge infrastructure in AWS, using nearly all the available services.
- Autodesk develops software for the engineering, design, and entertainment industries. Using services like Amazon RDS and Amazon S3, Autodesk can focus on developing its machine learning tools instead of spending that time on managing the infrastructure
- BMW uses AWS for its new connected-car application, collecting sensor data from BMW 7-series cars to give drivers dynamically updated map information.
- Canon's imaging products division benefits from faster deployment times, lower cost, and global reach by using AWS to deliver cloud-based services such as mobile print and office imaging products.

- The world's largest cable company and the United States' leading provider of internet service, Comcast, uses AWS in a hybrid environment.
- Docker is a company helping to redefine the way developers build, ship, and run applications making use of containers. The Amazon EC2 container service helps them do it and many more.

❖FEATURES OR APPLICATIONS OF AWS:

- Security and durability - AWS encrypt the data, offering end-to-end privacy and storage.
- Flexibility - There is great flexibility in AWS, allowing developers to select the OS language and database.
- Ease of Use - AWS is easy to use. Developers can swiftly deploy and host applications, build new applications or migrate existing applications.
- Scalability - Applications can be easily scaled up or down depending on user requirements.
- Cost savings - Companies only pay for the computing power, storage and resources used, with no long-term commitments.

PRACTICAL: 10

AIM: INTRODUCTION TO MICROSOFT AZURE

MICROSOFT AZURE: Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications in the public cloud.

- Some facts about Azure which we should know about: - It was launched on February 1, 2010, significantly later than its main competitor, AWS.
  - It's free to start and follows a pay-per-use model, which means you pay only for the services you opt for.
  - Interestingly, 80 percent of the Fortune 500 companies use Azure services for their cloud computing needs.
  - Azure supports multiple programming languages, including Java, Node Js, and C#.
  - Azure has the number of data centres around the world. There are 42 Azure data centres spread around the globe, which is the highest number of data centres for any cloud platform.

- **SERVICES OF AZURE:** Azure provides more than 200 services which are divided into 18 categories. Some are given below:

Networking:

· Azure CDN

Azure CDN (Content Delivery Network) is for delivering content to users. It uses a high bandwidth, and content can be transferred to any person around the globe.

· **Express Route**

This service lets you connect your on-premise network to the Microsoft cloud or any other services that you want, through a private connection.

· **Virtual network**

The virtual network allows you to have any of the Azure services communicate with one another privately and securely.

- **Azure DNS**

This service allows you to host your DNS domains or system domains on Azure.

Compute Services:

- ·**Virtual Machine**

This service enables you to create a virtual machine in Windows, Linux or any other configuration in seconds.

- ·**Cloud Service**

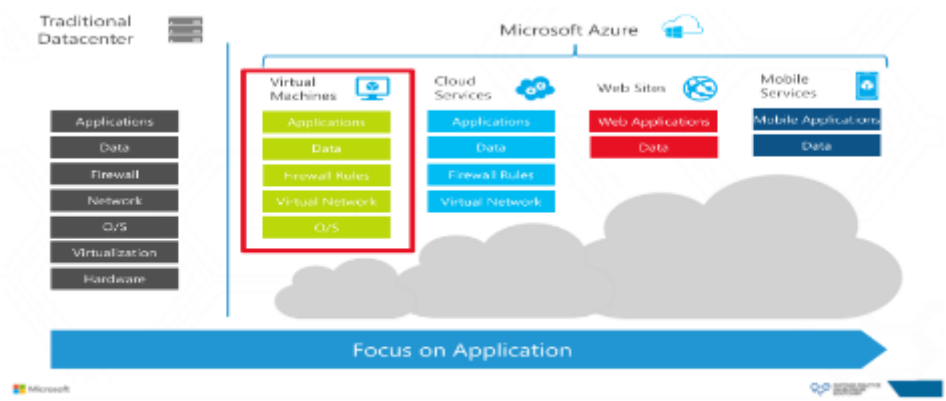
This service lets you create scalable applications within the cloud. Once the application is deployed, everything, including
provisioning, load balancing, and health monitoring, is taken care of by Azure.

- **Functions (for Applications and Websites)**

The best part about this service is that you need not worry about hardware requirements while developing applications or making websites because Azure takes care of that. All you need to do is provide the code. With functions, you can create applications in any programming language.

- **Mobile Services**

Azure Mobile Services provides a scalable cloud backend for building Windows Store, Windows Phone, Apple iOS, Android, and HTML/JavaScript applications. Store data in the cloud, authenticate users, and send push notifications to your application within minutes.



➢ **Storage:**

- **Disk Storage**

This service allows you to choose from either HDD (Hard Disk Drive) or SSD (Solid State Drive) as your storage option along with your virtual machine.

- **Blob Storage**

This service is optimized to store a massive amount of unstructured data, including text and even binary data.

- **File Storage**

This is a managed file storage service that can be accessed via industry SMB (server message block) protocol.

- **Queue Storage**

With queue storage, you can provide stable message queuing for a large workload. This service can be accessed from anywhere in this world.

- **USES OF AZURE :** Some of its uses are given below:
  - Application development: You can create any web application in Azure.
  - Testing: After developing an application successfully on the platform, you can test it.
  - Application hosting: Once the testing is done, Azure can help you host the application.
  - Create virtual machines : You can create virtual machines in any configuration you want with the help of Azure.
  - Integrate and sync features: Azure lets you integrate and sync virtual devices and directories.
  - Collect and store metrics: Azure lets you collect and store metrics, which can help you find what works.
  - Virtual hard drives: These are extensions of the virtual machines; they provide a huge amount of data storage.