SIES College of Arts, Science & Commerce

RISE WITH EDUCATION

NAAC REACCREDITED - 'A' GRADE

ISO 9001 : 2008

S.I.E.S College of Arts, Science and Commerce
Sion(W), Mumbai – 400 022.

CERTIFICATE

This is to certify that Mr. **MAHARAJA AAKASH THANGARAJ** Roll No. **TCS2223038** Has successfully completed the necessary course of experiments in the subject of **WebServices** during the academic year **2022 – 2023** complying with the requirements of **University of Mumbai**, for the course of **T.Y.BSc. Computer Science [Semester-5]**

Prof. In-Charge
Ms. Maya Nair
(WS)

Examination Date:
Examiner's Signature & Date:

Head of the Department
Prof. Manoj Singh

College Seal
And
Date

SR NO	Practical Name	SIGN
1	Create a TimeServer webservice in Java and Consume it in java and other technologies like php and .NET	
2	Create a Java WS for performing basic calculations like addition, subtraction ,multiplication and Division and create a java client that consumes the same	
3	Create a web service that gives – (i) NSE Index, (ii) BSE Index, (iii)Gold Rate. The values are stored in database. Also create a web client for a share trading firm that displays these values on its home page	
4	Create a web service for UGC that contains a method which accepts college name as parameter and returns the NAAC rating. The college names and their ratings are stored in database. Design a web client to test the above web service	
5	Design a web service for a channel containing 2 functions – 1st function called getBreakingNews which accepts date as string parameter and returns special news of that day, 2nd function called getPrediction accepts sunsign name as string parameter and returns predictions as string. Design a client to test the above web service	
6	Design a Restful webservice from a database table Employee with columns empid,empname and Designation. Test the webservice for the various http requests	
7	Design a Restful webservice from a database table Student with columns rollno, name and totalmarks. Create a restful client that displays the data by accessing restful service	
8	Create a WCF service to perform calculations like Addition, Subtraction , Multiplication and Division. Create a client for WCF which invokes the various operation	
9	Create a WCF service with different endpoint for Soap based and Rest based implementation	
10	To create a RESTful client for Practical No.7	

PRACTICAL 1

NAME: MAHARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM: Create a TimeServer webservice in Java and Consume it in java and other technologies like php and .NET

NET BEANS TIME SERVER CODE:

TimeService.java code:

```
package ts;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
@WebService(serviceName = "TimeService")
public class TimeService {
    @WebMethod(operationName = "getTimeString")
    public String getTimeAsString() {
        return new java.util.Date().toString();
    }
    @WebMethod(operationName = "getTimeElapsed")
    public long getTimeAsElapsed() {
        return new java.util.Date().getTime();
    }
}
```

NET BEANS TIME SERVER CLIENT CODE:

INDEX.JSP CODE:

```
<%--
    Document   : index
    Created on : Jul 27, 2022, 9:44:18 AM
    Author     : sies
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>SYSTEM</title>
    </head>
    <body>
        <h1>SYSTEM DATE</h1>
            <%-- start web service invocation --%><hr/>
    <%
    try {
      ts.TimeService_Service service = new ts.TimeService_Service();
      ts.TimeService port = service.getTimeServicePort();
      // TODO process result here
      long result = port.getTimeElapsed();
      out.println("Time as elapsed = "+result);
    } catch (Exception ex) {
      // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
        <%-- start web service invocation --%><hr/>
    <%
    try {
      ts.TimeService_Service service = new ts.TimeService_Service();
      ts.TimeService port = service.getTimeServicePort();
      // TODO process result here
      java.lang.String result = port.getTimeString();
      out.println("Time as String = "+result);
    } catch (Exception ex) {
    }
    %>
    <%-- end web service invocation --%><hr/>
    </body>
</html>
```

VB TIME SERVER CLIENT CODE:

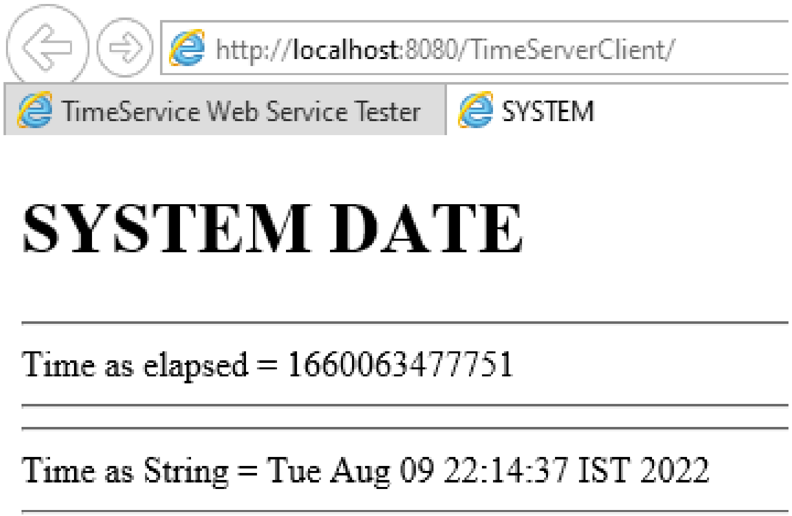
PROGRAM.CS CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace TimeServiceClient
{
    class Program
    {
        static void Main(string[] args)
        {
            ServiceReference1.TimeServiceClient client = new
ServiceReference1.TimeServiceClient();
            Console.WriteLine("Time as String: " + client.getTimeString());
            Console.WriteLine("Time Elapsed: " + client.getTimeElapsed());
            Console.Read();
        }
    }
}
```

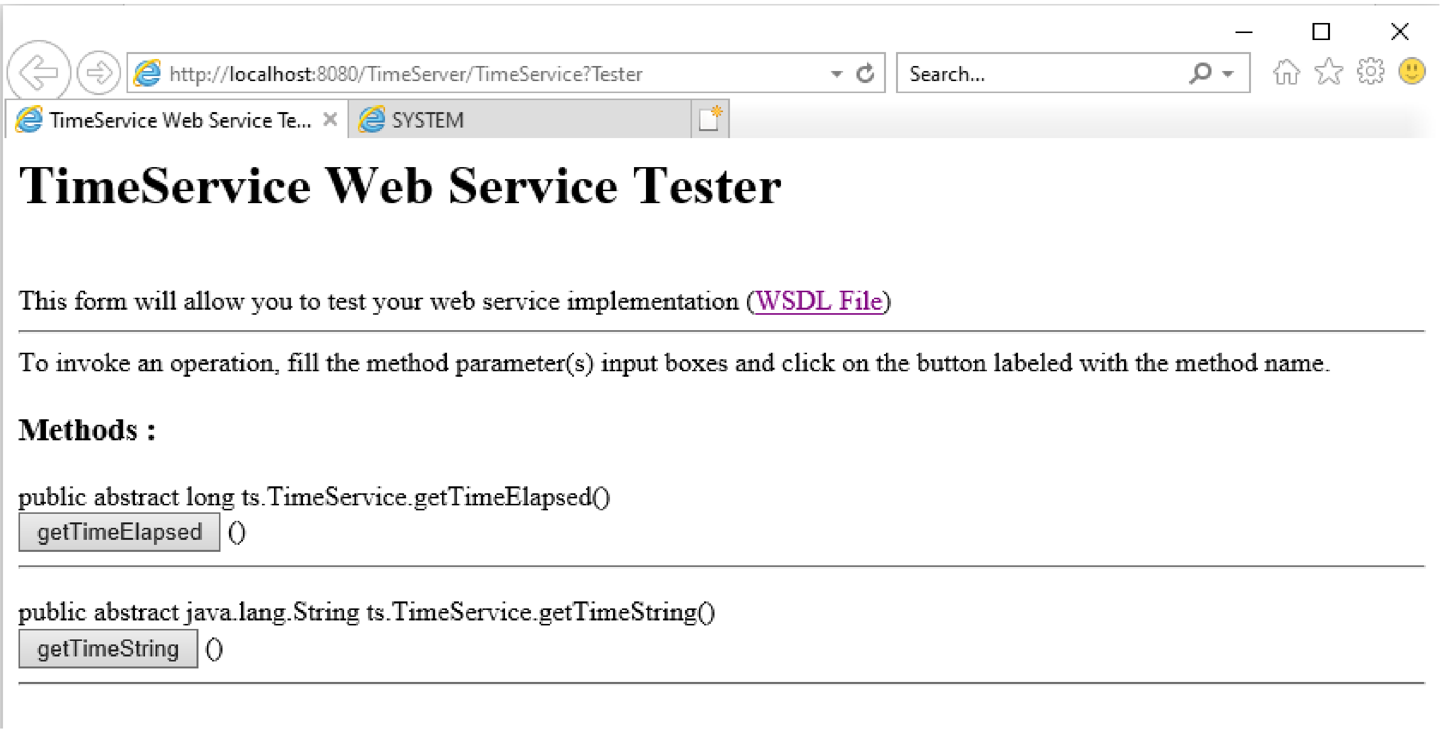
XAMPP TCLIENT.PHP CODE:

```
<?php

$client=new SoapClient("http://localhost:8080/TimeServer/TimeService?WSDL");

$t1=$client->getTimeAsElapsed();

echo "Elapsed time is",$t1->return; /* here the return is not a function but the name and
can be change by using @webresult before the function */

$t2=$client->getTimeAsString();

echo "<br>Time as String is",$t2->return;

?>
```

OUTPUT:


http://localhost:8080/TimeServerClient/
TimeService Web Service Tester
SYSTEM

```
SYSTEM DATE

Time as elapsed = 1660063477751

Time as String = Tue Aug 09 22:14:37 IST 2022
```


http://localhost:8080/TimeServer/TimeService?Tester
Search...
TimeService Web Service Te...
SYSTEM

TimeService Web Service Tester

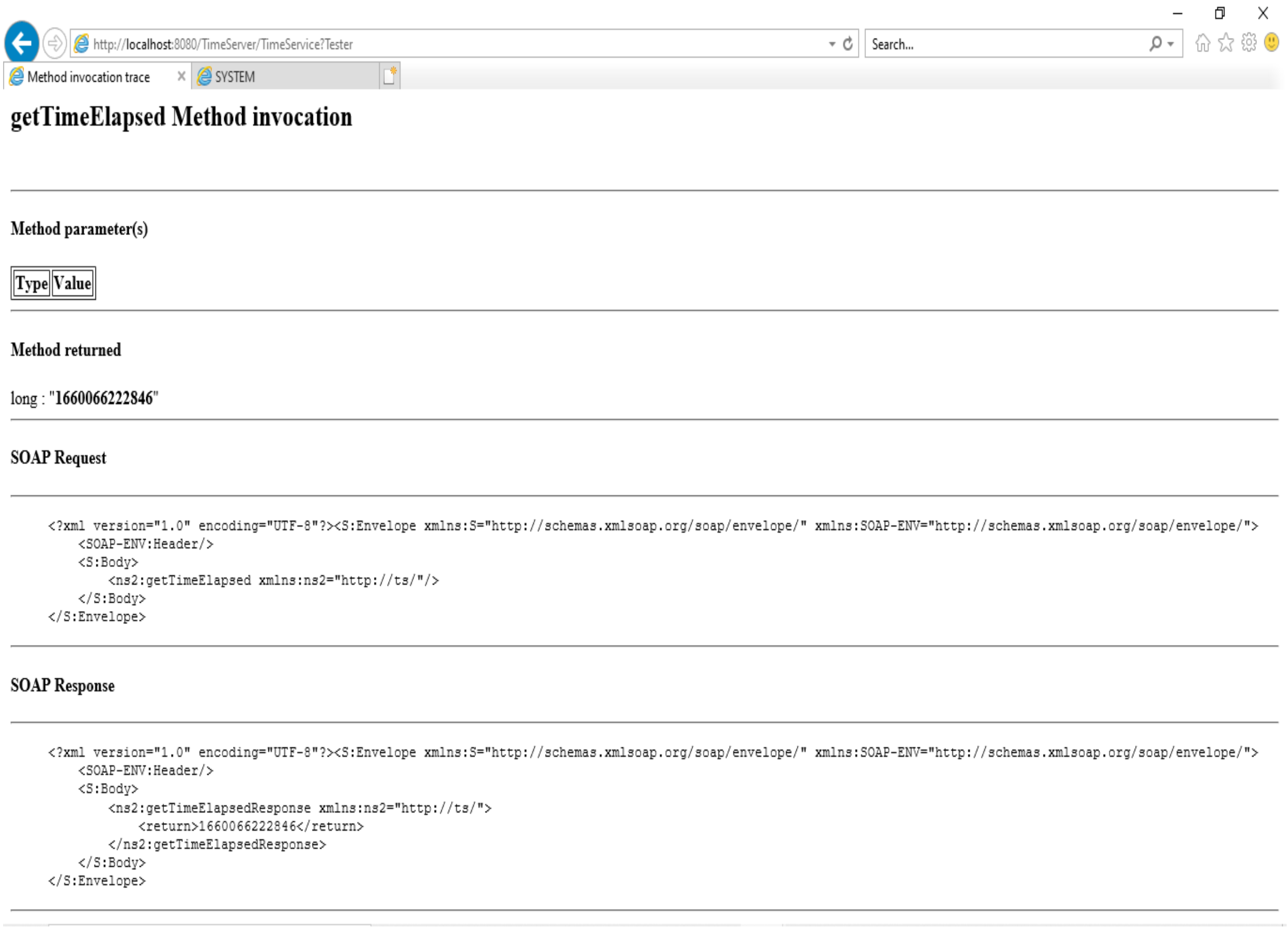
This form will allow you to test your web service implementation (WSDL File)

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

Methods :

public abstract long ts.TimeService.getTimeElapsed()
getTimeElapsed ()

public abstract java.lang.String ts.TimeService.getTimeString()
getTimeString ()



getTimeElapsed Method invocation

Method parameter(s)

Type	Value

Method returned

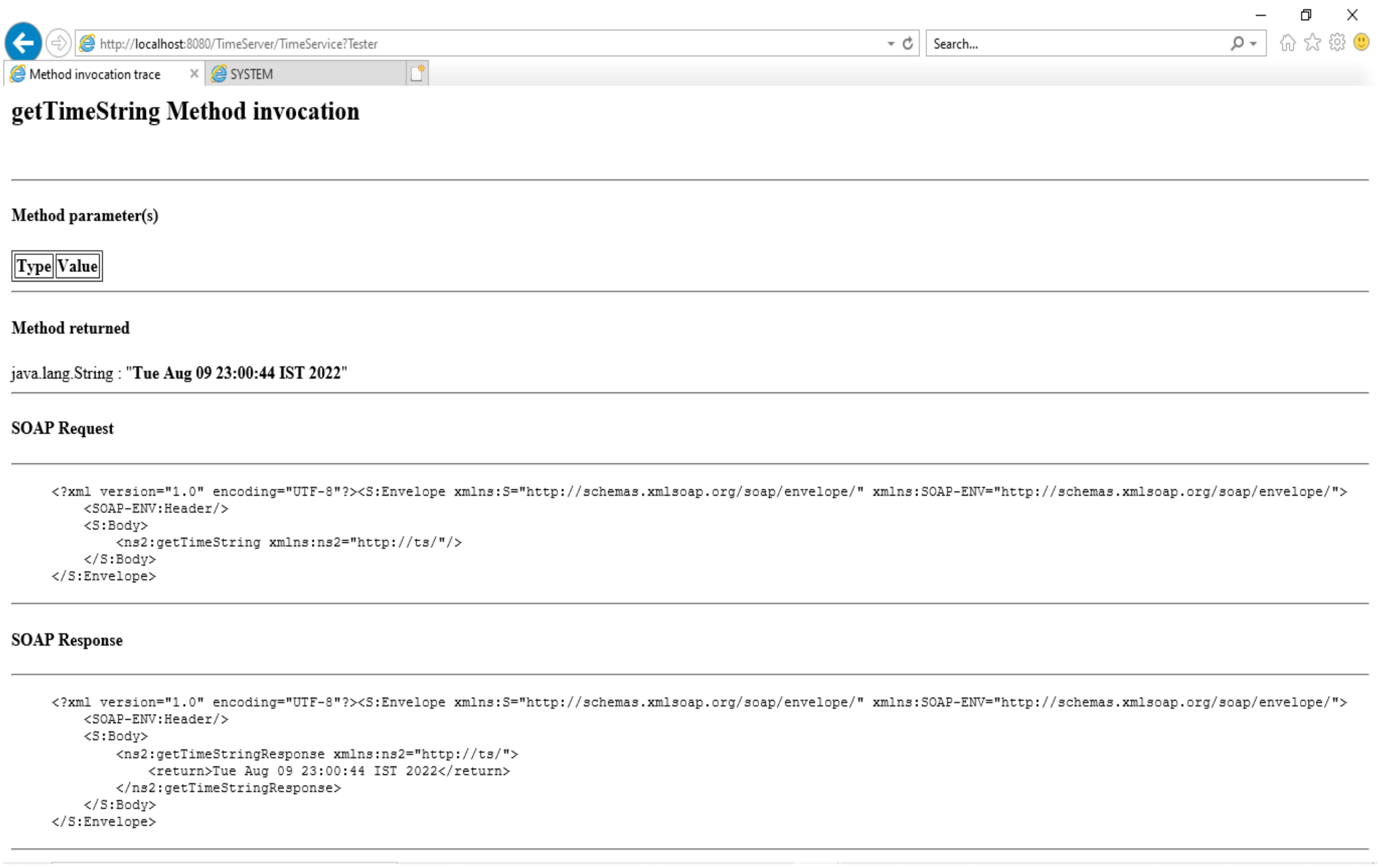
long : "1660066222846"

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:getTimeElapsed xmlns:ns2="http://ts/"/>
    </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:getTimeElapsedResponse xmlns:ns2="http://ts/">
            <return>1660066222846</return>
        </ns2:getTimeElapsedResponse>
    </S:Body>
</S:Envelope>
```



getTimeString Method invocation

Method parameter(s)

Type	Value

Method returned

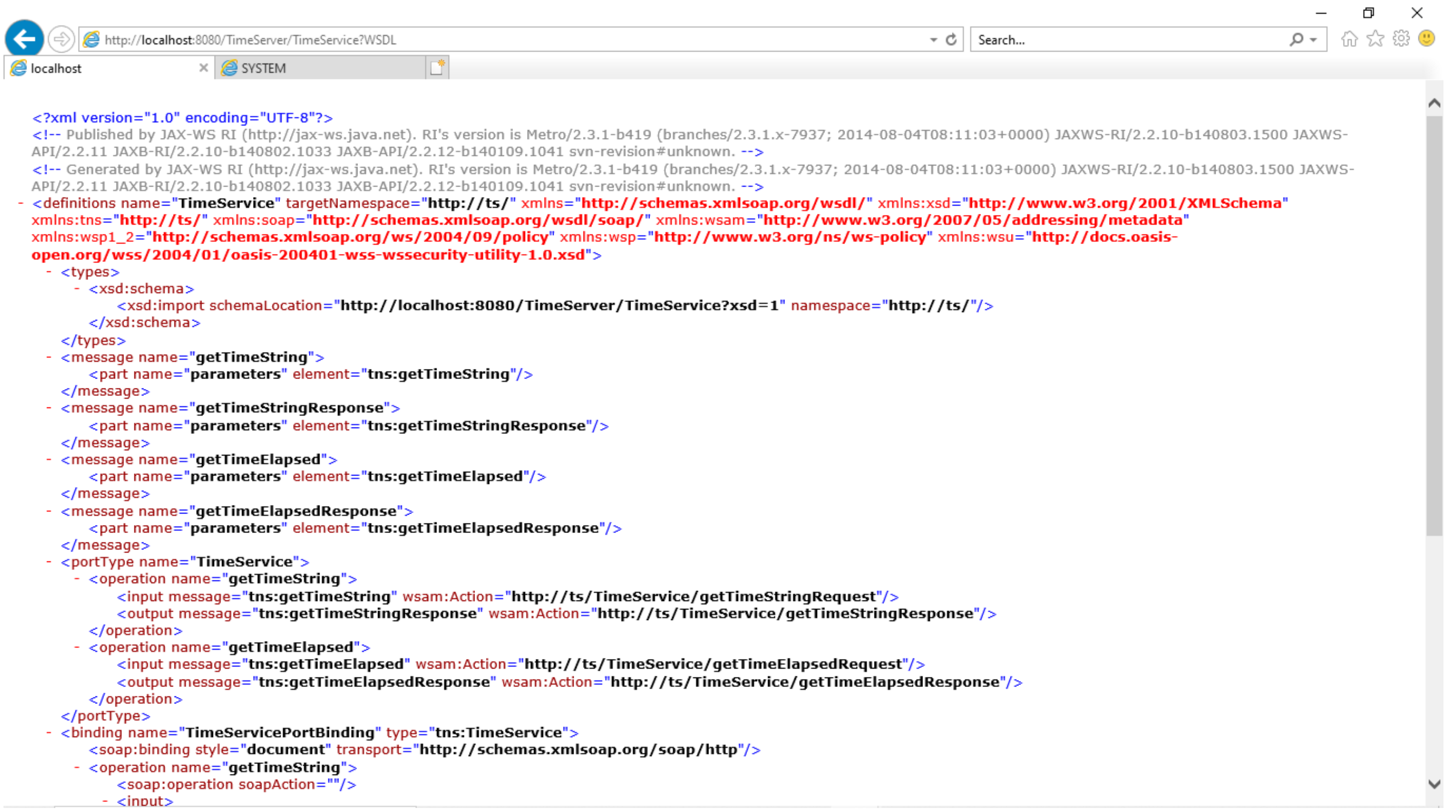
java.lang.String : "**Tue Aug 09 23:00:44 IST 2022**"

SOAP Request

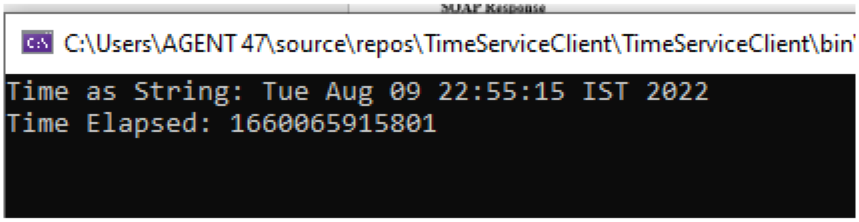
```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:getTimeString xmlns:ns2="http://ts/"/>
    </S:Body>
</S:Envelope>
```

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:getTimeStringResponse xmlns:ns2="http://ts/">
            <return>Tue Aug 09 23:00:44 IST 2022</return>
        </ns2:getTimeStringResponse>
    </S:Body>
</S:Envelope>
```



```
  <?xml version="1.0" encoding="UTF-8"?>
  <!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2.2.10-b140803.1500 JAXWS-
  API/2.2.11 JAXB-RI/2.2.10-b140802.1033 JAXB-API/2.2.12-b140109.1041 svn-revision#unknown. -->
  <!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2.2.10-b140803.1500 JAXWS-
  API/2.2.11 JAXB-RI/2.2.10-b140802.1033 JAXB-API/2.2.12-b140109.1041 svn-revision#unknown. -->
- <definitions name="TimeService" targetNamespace="http://ts/" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://ts/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsu="http://docs.oasis-
  open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  - <types>
    - <xsd:schema>
        <xsd:import schemaLocation="http://localhost:8080/TimeServer/TimeService?xsd=1" namespace="http://ts/"/>
      </xsd:schema>
    </types>
  - <message name="getTimeString">
      <part name="parameters" element="tns:getTimeString"/>
    </message>
  - <message name="getTimeStringResponse">
      <part name="parameters" element="tns:getTimeStringResponse"/>
    </message>
  - <message name="getTimeElapsed">
      <part name="parameters" element="tns:getTimeElapsed"/>
    </message>
  - <message name="getTimeElapsedResponse">
      <part name="parameters" element="tns:getTimeElapsedResponse"/>
    </message>
  - <portType name="TimeService">
    - <operation name="getTimeString">
        <input message="tns:getTimeString" wsam:Action="http://ts/TimeService/getTimeStringRequest"/>
        <output message="tns:getTimeStringResponse" wsam:Action="http://ts/TimeService/getTimeStringResponse"/>
      </operation>
    - <operation name="getTimeElapsed">
        <input message="tns:getTimeElapsed" wsam:Action="http://ts/TimeService/getTimeElapsedRequest"/>
        <output message="tns:getTimeElapsedResponse" wsam:Action="http://ts/TimeService/getTimeElapsedResponse"/>
      </operation>
    </portType>
  - <binding name="TimeServicePortBinding" type="tns:TimeService">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    - <operation name="getTimeString">
        <soap:operation soapAction=""/>
      - <input>
```



C:\Users\AGENT 47\source\repos\TimeServiceClient\TimeServiceClient\bin

```
Time as String: Tue Aug 09 22:55:15 IST 2022
Time Elapsed: 1660065915801
```

PRACTICAL 2

NAME: MAHAARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM: Create a Java WS for performing basic calculations like addition, subtraction, multiplication and Division and create a java client that consumes the same.

CODE:

WS:

```
package newpackage;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebResult;
@WebService(serviceName = "ws2")
public class ws2 {
  @WebMethod(operationName = "Addition")
  @WebResult(name="sum")
  public int Addition(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    return (a+b);
  }
  @WebMethod(operationName = "Subraction")
  @WebResult(name="sub")
  public int Subraction(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    return (a-b);
  }
  @WebMethod(operationName = "Multiplication")
  @WebResult(name="mul")
  public int Multiplication(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    return (a*b);
  }
  @WebMethod(operationName = "Division")
  @WebResult(name="div")
```

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

WS CLIENT INDEX.HTML CODE:

```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="client.jsp">
            Enter First Number: <input type="text" name="first" value="" /><br>
            Enter Second Number: <input type="text" name="second" value="" /><br>
            Add<input type="radio" name="operations" value="add" /><br>
            Sub<input type="radio" name="operations" value="sub" /><br>
            Mul<input type="radio" name="operations" value="mul" /><br>
            Div<input type="radio" name="operations" value="div" /><br>
            <input type="submit" value="Submit" />
        </form>
    </body>
</html>
```

WS CLIENT CLIENT.JSP CODE:

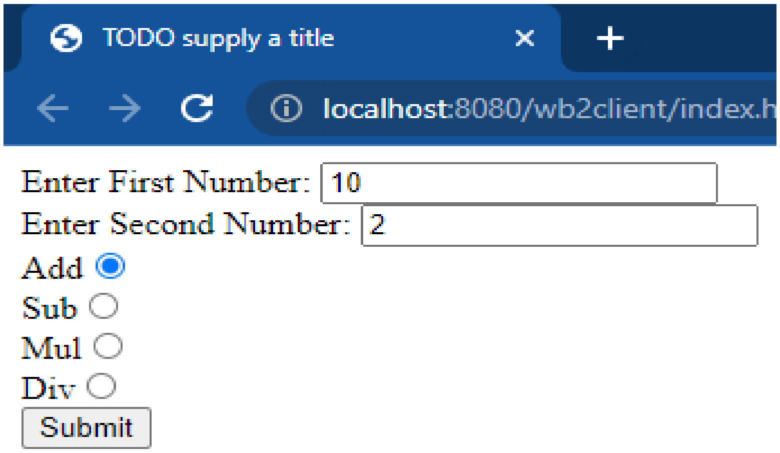
```
<%--
    Document   : client
    Created on : Aug 11, 2022, 10:23:08 AM
    Author     : sies
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
```

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <%
      int z=Integer.parseInt(request.getParameter("first"));
      int y=Integer.parseInt(request.getParameter("second"));
      String choice=request.getParameter("operations");
      if(choice.equals("add"))
      {
  try {
        newpackage.Ws2_Service service = new newpackage.Ws2_Service();
        newpackage.Ws2 port = service.getWs2Port();
         // TODO initialize WS operation arguments here
        int a =z;
        int b = y;
        // TODO process result here
        double result = port.addition(a, b);
        out.println("Result = "+result);
  } catch (Exception ex) {
        // TODO handle custom exceptions here
  }
      }
      else if(choice.equals("sub"))
      {
  try {
        newpackage.Ws2_Service service = new newpackage.Ws2_Service();
        newpackage.Ws2 port = service.getWs2Port();
         // TODO initialize WS operation arguments here
        int a =z;
```

```
     int b =y;
     // TODO process result here
     double result = port.subraction(a, b);
     out.println("Result = "+result);
} catch (Exception ex) {
     // TODO handle custom exceptions here
}
    }
    else if(choice.equals("mul"))
    {
try {
     newpackage.Ws2_Service service = new newpackage.Ws2_Service();
     newpackage.Ws2 port = service.getWs2Port();
      // TODO initialize WS operation arguments here
     int a = z;
     int b =y;
     // TODO process result here
     double result = port.multiplication(a, b);
     out.println("Result = "+result);
} catch (Exception ex) {
     // TODO handle custom exceptions here
}
    }
    else if(choice.equals("div"))
    {
try {
     newpackage.Ws2_Service service = new newpackage.Ws2_Service();
     newpackage.Ws2 port = service.getWs2Port();
      // TODO initialize WS operation arguments here
     int a = z;
     int b =y;
     // TODO process result here
```

```
            double result = port.division(a, b);
            out.println("Result = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
        }
      %>
    </body>
</html>
```

OUTPUT:



TODO supply a title

localhost:8080/wb2client/index.h

Enter First Number: 10

Enter Second Number: 2

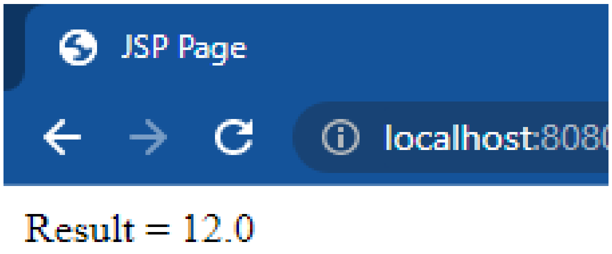
Add ●

Sub ○

Mul ○

Div ○

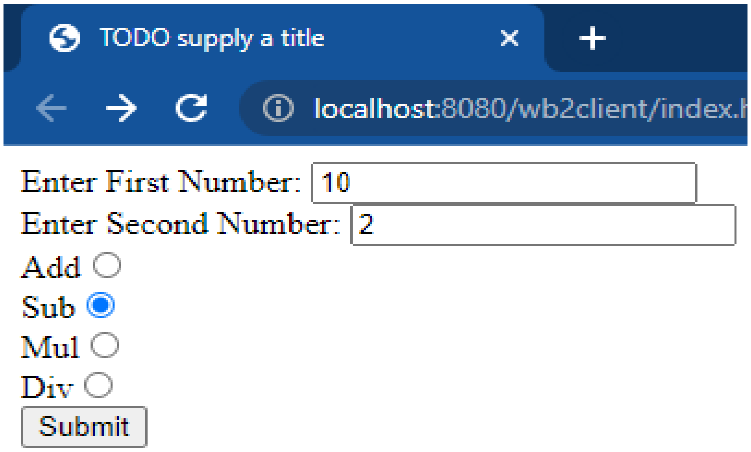
Submit



JSP Page

localhost:808

Result = 12.0



TODO supply a title

localhost:8080/wb2client/index.h

Enter First Number: 10

Enter Second Number: 2

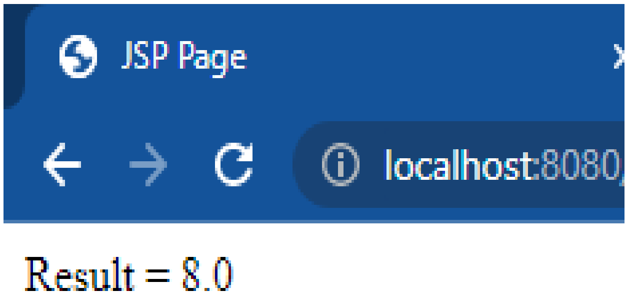
Add ○

Sub ●

Mul ○

Div ○

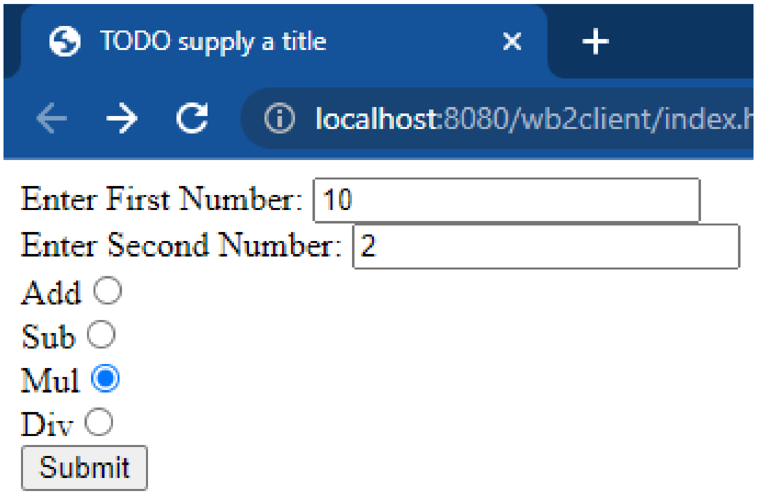
Submit



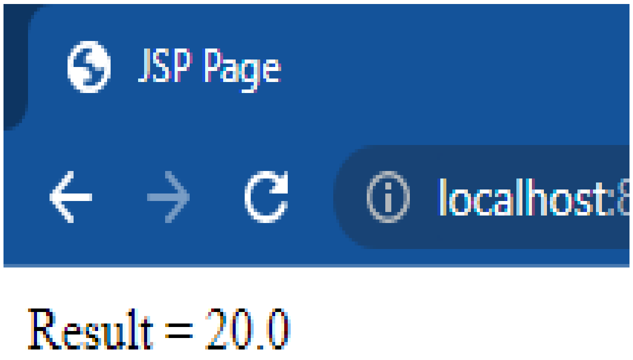
JSP Page

localhost:8080/

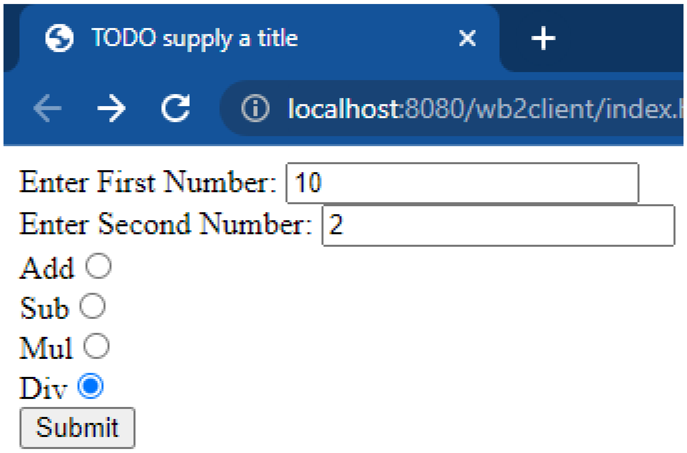
Result = 8.0



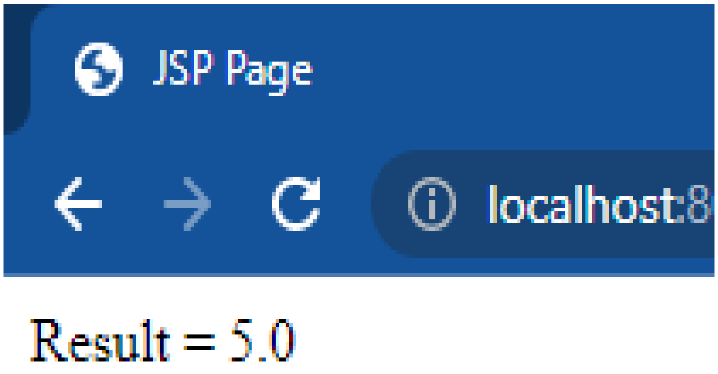
Enter First Number: 10
Enter Second Number: 2
Add ○
Sub ○
Mul ●
Div ○
Submit



Result = 20.0



Enter First Number: 10
Enter Second Number: 2
Add ○
Sub ○
Mul ○
Div ●
Submit



Result = 5.0

PRACTICAL 3

NAME: MAHAARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM: Create a web service that gives – (i) NSE Index, (ii) BSE Index, (iii)Gold Rate. The values are stored in database. Also create a web client for a share trading firm that displays these values on its home page.

CODE:

STOCKSERVICE.JAVA:

```
package stock;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import java.sql.*;
@WebService(serviceName = "StockService")
public class StockService {
   @WebMethod(operationName = "getNSE")
   public long getNSE() {
      long nse=0;
      try{
         //load driver
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         //Connection creation
         Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/StockDatabase","tmadata","tma@123");
         Statement stmt=con.createStatement();
         ResultSet rs=stmt.executeQuery("Select * from stockdata");
         rs.next();
         nse=rs.getInt("NSE");
      }
      catch(Exception e){
         e.printStackTrace();
```

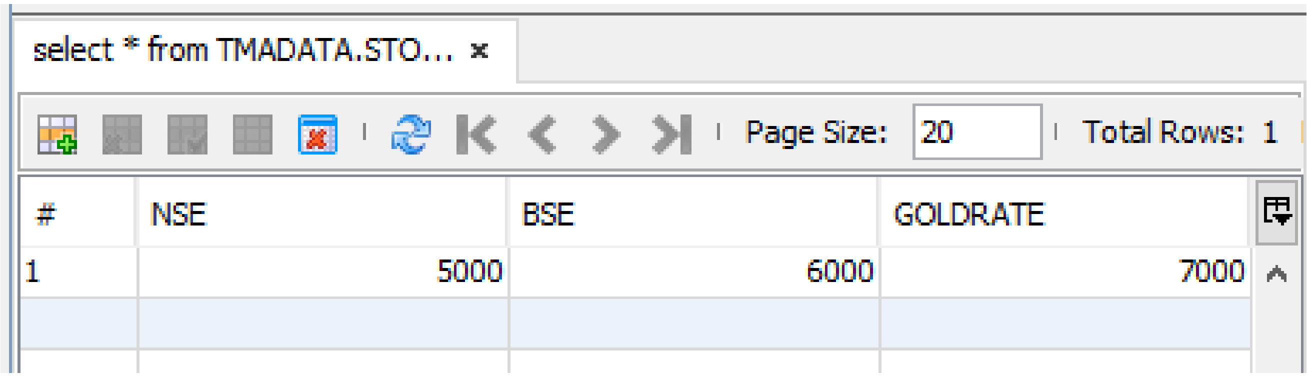
```
        }
        return nse;
    }
    @WebMethod(operationName = "getBSE")
    public long getBSE() {
        long bse=0;
        try{
            //load driver
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            //Connection creation
            Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/StockDatabase","tmadata","tma@123");
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from stockdata");
            rs.next();
            bse=rs.getInt("BSE");
        }
        catch(Exception e){
            e.printStackTrace();
        }
        return bse;
    }
    @WebMethod(operationName = "getGoldRate")
    public long getGoldRate() {
        long goldrate=0;
        try{
            //load driver
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            //Connection creation
            Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/StockDatabase","tmadata","tma@123");
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from stockdata");
            rs.next();
            goldrate=rs.getInt("GoldRate");
        }
        catch(Exception e){
            e.printStackTrace();
        }
        return goldrate;
    }
}
```

CLIENT.JSP CODE:

```
<%--
    Document   : client
    Created on : Aug 18, 2022, 9:41:34 AM
    Author     : sies
--%>

<%@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>
            <%-- start web service invocation --%><hr/>
    <%
    try {
            stock.StockService_Service service = new stock.StockService_Service();
            stock.StockService port = service.getStockServicePort();
            // TODO process result here
            long result = port.getBSE();
            out.println("BSE = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
    <%-- start web service invocation --%><hr/>
    <%
    try {
            stock.StockService_Service service = new stock.StockService_Service();
            stock.StockService port = service.getStockServicePort();
            // TODO process result here
            long result = port.getGoldRate();
            out.println("Gold rate = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
    <%-- start web service invocation --%><hr/>
    <%
    try {
            stock.StockService_Service service = new stock.StockService_Service();
            stock.StockService port = service.getStockServicePort();
            // TODO process result here
            long result = port.getNSE();
            out.println("NSE = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
```

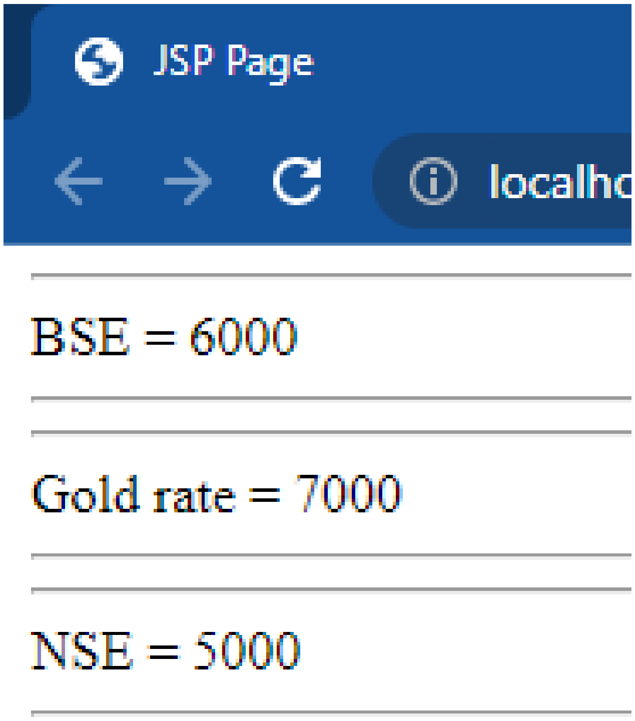
```
    </body>
</html>
```

OUTPUT:



Page Size: 20 Total Rows: 1

#	NSE	BSE	GOLDRATE
1	5000	6000	7000



BSE = 6000

Gold rate = 7000

NSE = 5000

PRACTICAL 4

NAME: MAHAARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM: Create a web service for UGC that contains a method which accepts college name as parameter and returns the NAAC rating. The college names and their ratings are stored in database. Design a web client to test the above web service.

CODE:

UGC.JAVA:

```
package CollegeGrade;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

@WebService(serviceName = "UGC")

public class UGC {

    @WebMethod(operationName = "getRatings")

    public String getRatings(@WebParam(name = "CollegeName") String CollegeName) {

        String grade="";

        try{

            //loading driver

            Class.forName("org.apache.derby.jdbc.ClientDriver"); //Passing the driver class url

            Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/UGCDATA","UGC","UGC123");

            PreparedStatement pstmt = con.prepareStatement("SELECT * from UGCList where CollegeName = ?");

            pstmt.setString(1, CollegeName);

            ResultSet rs = pstmt.executeQuery();

            rs.next();

            grade = rs.getString("COLLEGEGRADE");
        }
        catch(Exception e){
            e.printStackTrace(); //shows the error in glassfish console
        }
        return grade;
    }
}
```

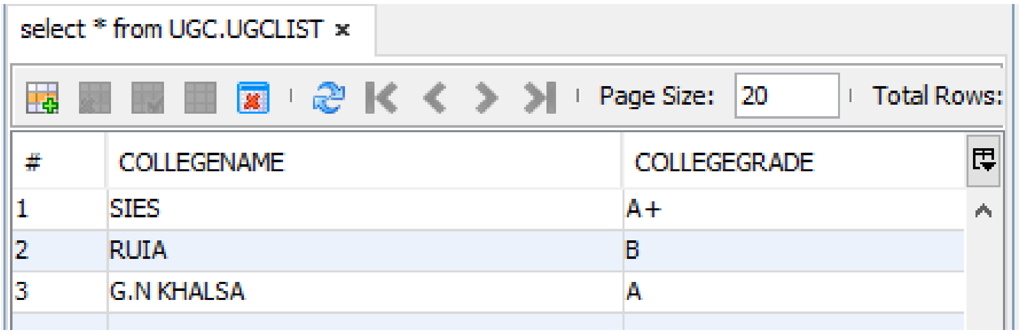
CLIENT.JSP CODE:

```
<%--
    Document   : client
    Created on : Aug 18, 2022, 10:49:46 AM
    Author     : sies
--%>

<%@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>
            <%-- start web service invocation --%><hr/>
    <%
        String grade = request.getParameter("value1");
    try {
             collegegrade.UGC_Service service = new collegegrade.UGC_Service();
             collegegrade.UGC port = service.getUGCPort();
              // TODO initialize WS operation arguments here
             java.lang.String x = grade;
             // TODO process result here
            java.lang.String result = port.getRatings(x);
            out.println("Result = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>


    </body>
</html>
```

INDEX.HTML CODE:

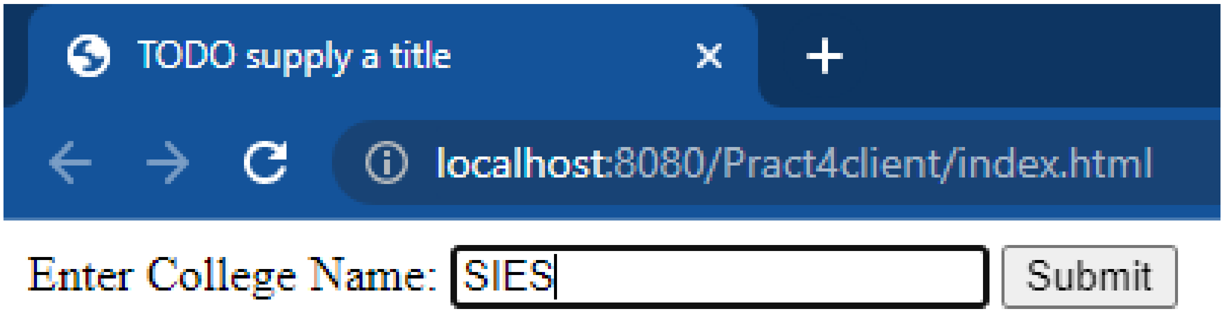
```
<!DOCTYPE html>
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="client.jsp" method="POST">
            Enter College Name: <input type="text" name="value1">
            <input type="submit" >
        </form>
    </body>
</html>
```

OUTPUT:



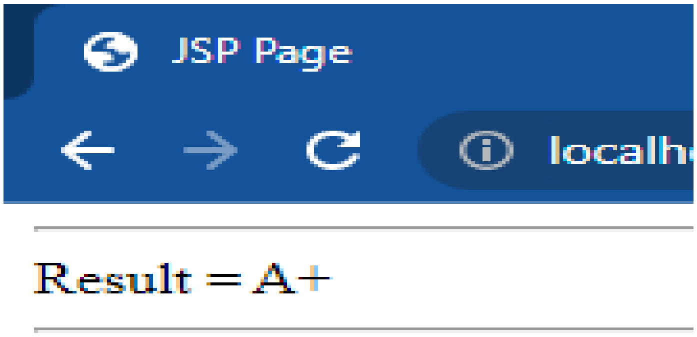
select * from UGC.UGCLIST ×

#	COLLEGENAME	COLLEGEGRADE
1	SIES	A+
2	RUIA	B
3	G.N KHALSA	A

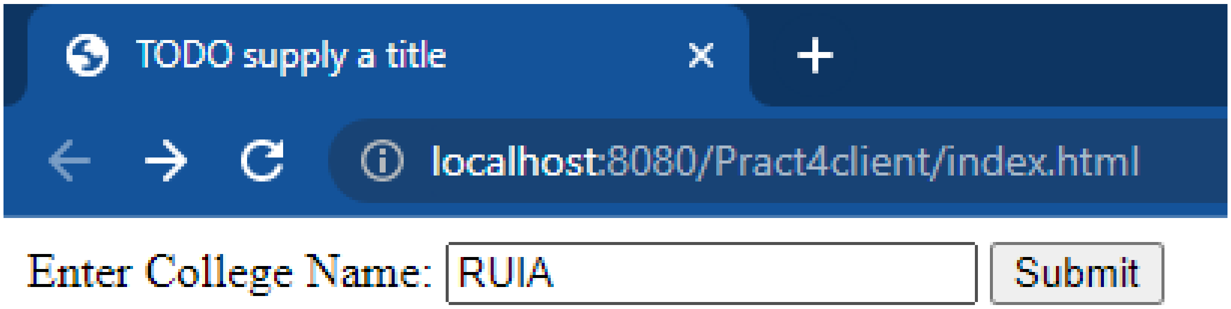


localhost:8080/Pract4client/index.html

Enter College Name: SIES

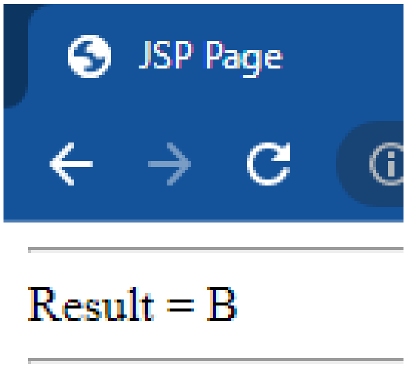


Result = A+



localhost:8080/Pract4client/index.html

Enter College Name: RUIA



Result = B

PRACTICAL 5

NAME: MAHARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM:Design a web service for a channel containing 2 functions – 1st function called getBreakingNews which accepts date as string parameter and returns special news of that day, 2nd function called getPrediction accepts sunsign name as string parameter and returns predictions as string. Design a client to test the above web service.

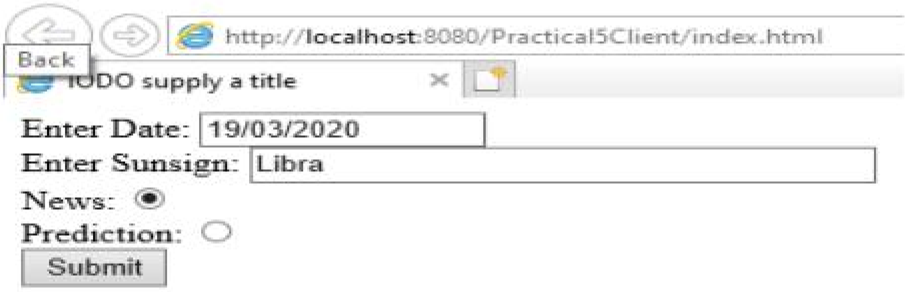
Code:

```
Prac5.java:
package pracs5;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import java.sql.*;
@WebService(serviceName = "Prac5")
public class Prac5 {
 /**
 * Web service operation
 */
 @WebMethod(operationName = "getBreakingNews")
 public String getBreakingNews(@WebParam(name = "Date") String Date)
 {
 String news="";
 try
 {
 Class.forName("org.apache.derby.jdbc.ClientDriver");
 Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Practical5DB",
"andy","andyboi");
 PreparedStatement pstmt = con.prepareStatement("select * from
BreakingNews where Date=?");
 pstmt.setString(1, Date);
 ResultSet rs = pstmt.executeQuery();
 rs.next();
 news=rs.getString("BreakingNews");
 }
 catch (Exception e)
 {
 e.printStackTrace();
 }
 return news;
 }
 /**
 * Web service operation
 */
 @WebMethod(operationName = "getPredictions")
 public String getPredictions(@WebParam(name = "Sunsign") String
Sunsign)
 {
 String predictions="";
 try
 {
 Class.forName("org.apache.derby.jdbc.ClientDriver");
 Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Practical5DB",
"jishan","jishan");
 PreparedStatement pstmt = con.prepareStatement("select * from
Predictions where Sunsign=?");
 pstmt.setString(1, Sunsign);
 ResultSet rs = pstmt.executeQuery();
 rs.next();
 predictions=rs.getString("Predictions");
 }
 catch (Exception e)
 {
 e.printStackTrace();
 }
 return predictions;
 }
}
```

Client.jsp:

```
<%--
 Document : client
 Created on : Aug 25, 2022, 9:30:19 AM
 Author : jishan
--%>
<%@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>
 <%
 String choice1 = request.getParameter("Choice");
 if(choice1.equals("News"))
 {
 try {
prac5.Prac5_Service service = new prac5.Prac5_Service();
prac5.Prac5 port = service.getPrac5Port();
// TODO initialize WS operation arguments here
java.lang.String date = request.getParameter("Date");
// TODO process result here
java.lang.String result = port.getBreakingNews(date);
out.println("Breaking News = "+result);
 } catch (Exception ex) {
// TODO handle custom exceptions here
 }
 }
 else{

 try {
prac5.Prac5_Service service = new prac5.Prac5_Service();
prac5.Prac5 port = service.getPrac5Port();
// TODO initialize WS operation arguments here
java.lang.String sunsign = request.getParameter("Sunsign");
// TODO process result here
java.lang.String result = port.getPredictions(sunsign);
out.println("Predictions = "+result);
 } catch (Exception ex) {
// TODO handle custom exceptions here
 }
 }
 %>
 </body>
</html>
```

OUTPUT:



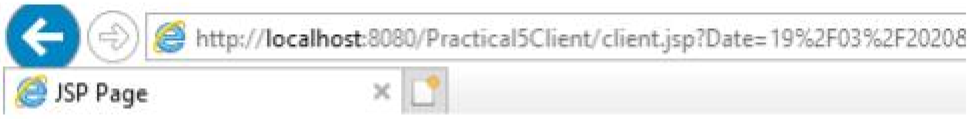
Enter Date: 19/03/2020

Enter Sunsign: Libra

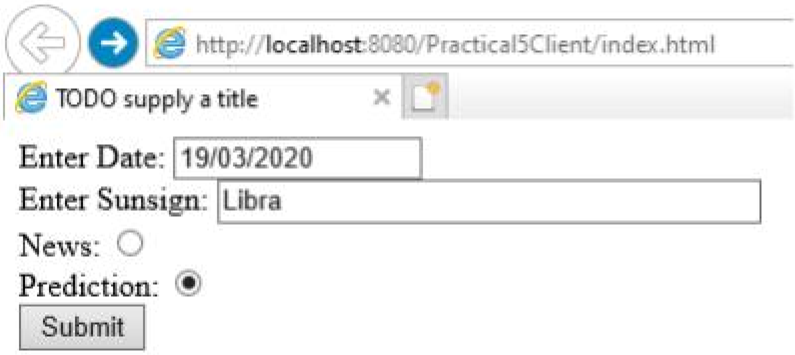
News: ●

Prediction: ○

Submit


http://localhost:8080/Practical5Client/client.jsp?Date=19%2F03%2F2020&
JSP Page

Breaking News = Coronavirus caused a disruption of services in India



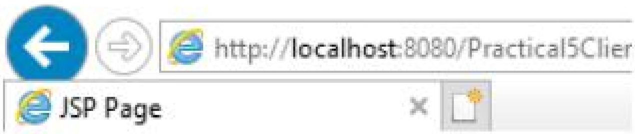
Enter Date: 19/03/2020

Enter Sunsign: Libra

News: ○

Prediction: ●

Submit



Predictions = You will travel by bus today

PRACTICAL 6

NAME: MAHARAJA AAKASH THANGARAJ

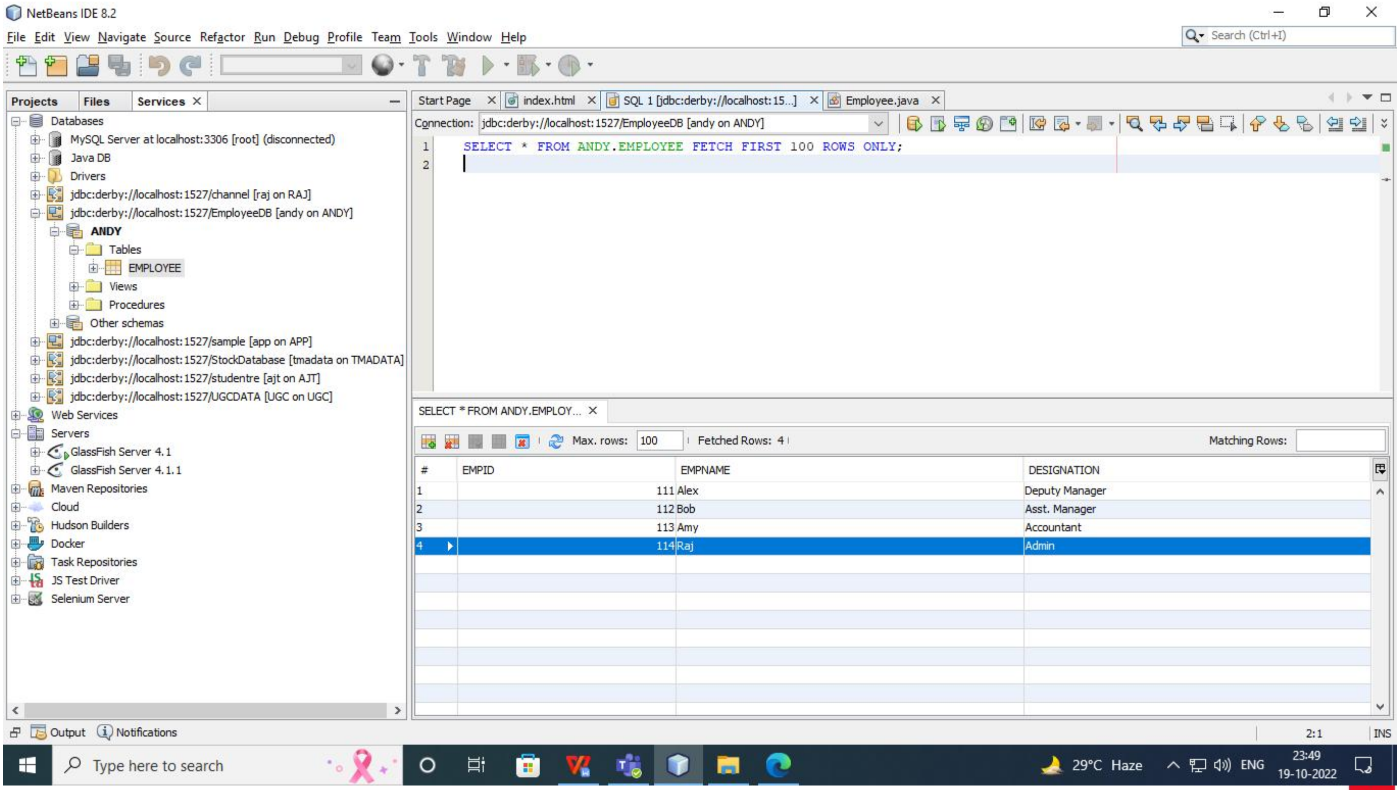
ROLL: TCS2223038

AIM:Design a Restful webservice from a database table Employee with columns empid,empname and Designation. Test the webservice for the various http requests

Employee.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 empdsn;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author AGENT47
 */
@Entity
@Table(name = "EMPLOYEE")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Employee.findAll", query = "SELECT e FROM Employee e")
    , @NamedQuery(name = "Employee.findByEmpid", query = "SELECT e FROM Employee e WHERE e.empid = :empid")
    , @NamedQuery(name = "Employee.findByEmpname", query = "SELECT e FROM Employee e WHERE e.empname
= :empname")
    , @NamedQuery(name = "Employee.findByDesignation", query = "SELECT e FROM Employee e WHERE e.designation
= :designation")})
public class Employee implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @NotNull
    @Column(name = "EMPID")
    private Integer empid;
    @Size(max = 50)
    @Column(name = "EMPNAME")
    private String empname;
    @Size(max = 50)
    @Column(name = "DESIGNATION")
    private String designation;

    public Employee() {
    }
    public Employee(Integer empid) {
        this.empid = empid;
    }

    public Integer getEmpid() {
        return empid;
    }

    public void setEmpid(Integer empid) {
        this.empid = empid;
    }

    public String getEmpname() {
        return empname;
    }

    public void setEmpname(String empname) {
        this.empname = empname;
    }

    public String getDesignation() {
        return designation;
    }

    public void setDesignation(String designation) {
        this.designation = designation;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (empid != null ? empid.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Employee)) {
            return false;
        }
        Employee other = (Employee) object;
        if ((this.empid == null && other.empid != null) || (this.empid != null && !this.empid.equals(other.empid))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "empdsn.Employee[ empid=" + empid + " ]";
    }

}
```

OUTPUT:

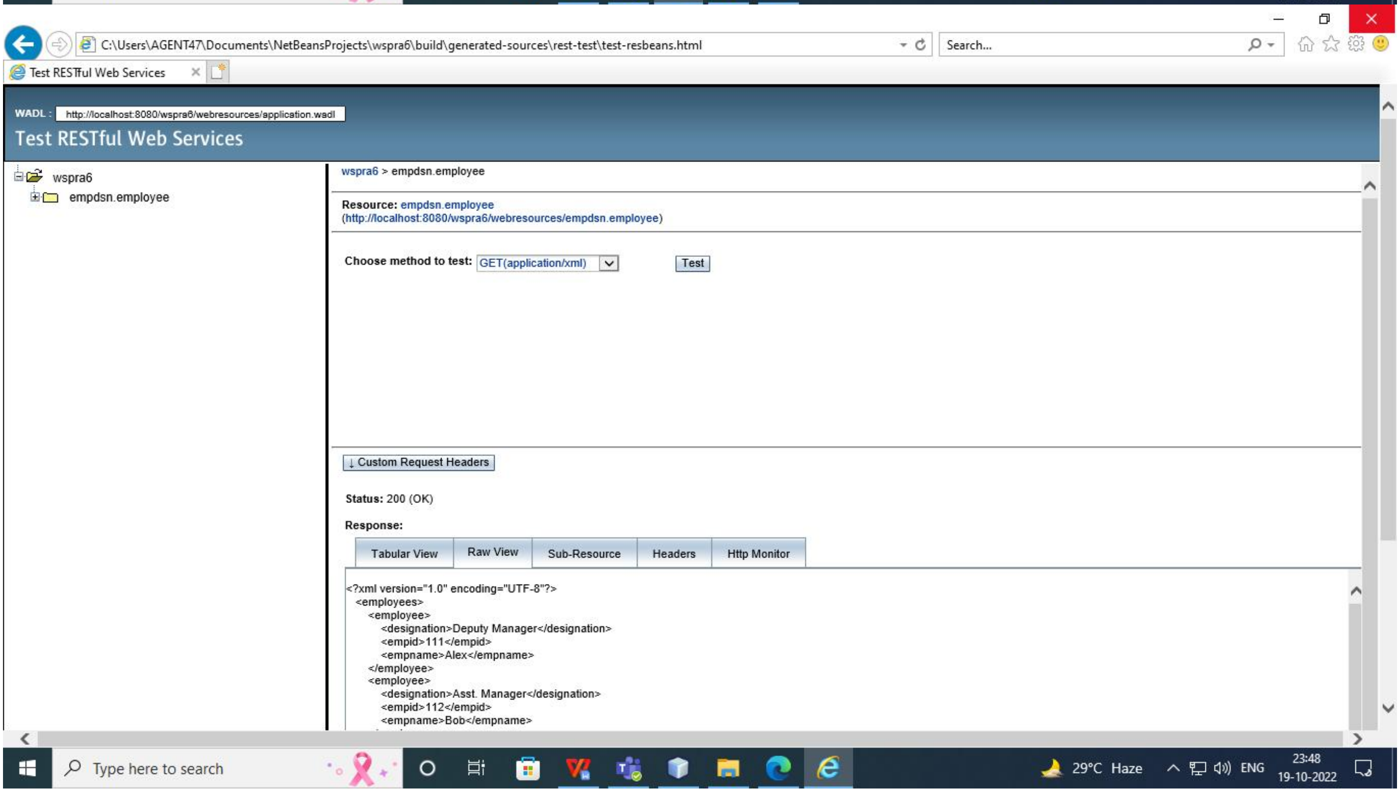

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Databases
MySQL Server at localhost:3306 [root] (disconnected)
Java DB
Drivers
jdbc:derby://localhost:1527/channel [raj on RAJ]
jdbc:derby://localhost:1527/EmployeeDB [andy on ANDY]
ANDY
Tables
EMPLOYEE
Views
Procedures
Other schemas
jdbc:derby://localhost:1527/sample [app on APP]
jdbc:derby://localhost:1527/StockDatabase [tmadata on TMADATA]
jdbc:derby://localhost:1527/studentre [ajt on AJT]
jdbc:derby://localhost:1527/UGCDATA [UGC on UGC]
Web Services
Servers
GlassFish Server 4.1
GlassFish Server 4.1.1
Maven Repositories
Cloud
Hudson Builders
Docker
Task Repositories
JS Test Driver
Selenium Server
Start Page
index.html
SQL 1 [jdbc:derby://localhost:15...]
Employee.java
SELECT * FROM ANDY.EMPLOY...
Output
Notifications
2:1
INS
Type here to search
29°C Haze
ENG
23:49
19-10-2022

Connection: jdbc:derby://localhost:1527/EmployeeDB [andy on ANDY]

```
SELECT * FROM ANDY.EMPLOYEE FETCH FIRST 100 ROWS ONLY;
```

Max. rows: 100 Fetched Rows: 4 Matching Rows:

#	EMPID	EMPNAME	DESIGNATION
1	111	Alex	Deputy Manager
2	112	Bob	Asst. Manager
3	113	Amy	Accountant
4	114	Raj	Admin



WADL : http://localhost:8080/wspra6/webresources/application.wadl

Test RESTful Web Services

wspra6 > empdsn.employee

Resource: empdsn.employee (http://localhost:8080/wspra6/webresources/empdsn.employee)

Choose method to test: GET(application/xml)

Status: 200 (OK)

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
  <employees>
    <employee>
      <designation>Deputy Manager</designation>
      <empid>111</empid>
      <empname>Alex</empname>
    </employee>
    <employee>
      <designation>Asst. Manager</designation>
      <empid>112</empid>
      <empname>Bob</empname>
```

PRACTICAL 7

NAME: MAHARAJA AAKASH THANGARAJ

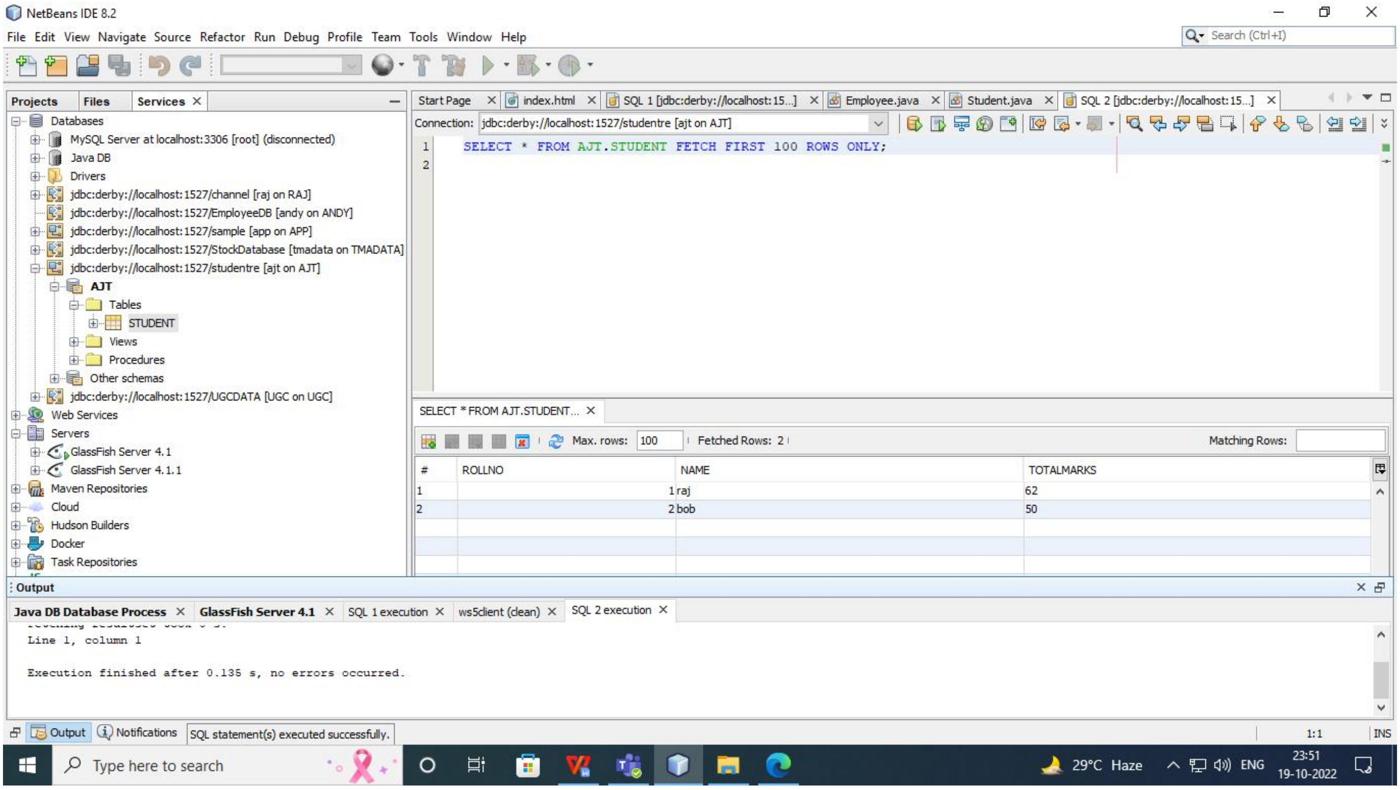
ROLL: TCS2223038

AIM:Design a Restful webservice from a database table Student with columns rollno, name and totalmarks. Create a restful client that displays the data by accessing restful service

Student.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 student;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author AGENT47
 */
@Entity
@Table(name = "STUDENT")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Student.findAll", query = "SELECT s FROM Student s")
    , @NamedQuery(name = "Student.findByRollno", query = "SELECT s FROM Student s WHERE s.rollno = :rollno")
    , @NamedQuery(name = "Student.findByName", query = "SELECT s FROM Student s WHERE s.name = :name")
    , @NamedQuery(name = "Student.findByTotalmarks", query = "SELECT s FROM Student s WHERE s.totalmarks = :totalmarks")})
public class Student implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @NotNull
    @Column(name = "ROLLNO")
    private Integer rollno;
    @Size(max = 50)
    @Column(name = "NAME")
    private String name;
    @Size(max = 50)
    @Column(name = "TOTALMARKS")
    private String totalmarks;

    public Student() {
    }

    public Student(Integer rollno) {
        this.rollno = rollno;
    }

    public Integer getRollno() {
        return rollno;
    }

    public void setRollno(Integer rollno) {
        this.rollno = rollno;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getTotalmarks() {
        return totalmarks;
    }

    public void setTotalmarks(String totalmarks) {
        this.totalmarks = totalmarks;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (rollno != null ? rollno.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Student)) {
            return false;
        }
        Student other = (Student) object;
        if ((this.rollno == null && other.rollno != null) || (this.rollno != null && !this.rollno.equals(other.rollno))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "student.Student[ rollno=" + rollno + " ]";
    }

}
```

OUTPUT:


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
Databases
MySQL Server at localhost:3306 [root] (disconnected)
Java DB
Drivers
jdbc:derby://localhost:1527/channel [raj on RAJ]
jdbc:derby://localhost:1527/EmployeeDB [andy on ANDY]
jdbc:derby://localhost:1527/sample [app on APP]
jdbc:derby://localhost:1527/StockDatabase [tmadata on TMADATA]
jdbc:derby://localhost:1527/studentre [ajt on AJT]
AJT
Tables
STUDENT
Views
Procedures
Other schemas
jdbc:derby://localhost:1527/UGCDATA [UGC on UGC]
Web Services
Servers
GlassFish Server 4.1
GlassFish Server 4.1.1
Maven Repositories
Cloud
Hudson Builders
Docker
Task Repositories
Start Page
index.html
SQL 1 [jdbc:derby://localhost:15...]
Employee.java
Student.java
SQL 2 [jdbc:derby://localhost:15...]

Connection: jdbc:derby://localhost:1527/studentre [ajt on AJT]

```
SELECT * FROM AJT.STUDENT FETCH FIRST 100 ROWS ONLY;
```

SELECT * FROM AJT.STUDENT...

Max. rows: 100 Fetched Rows: 2 Matching Rows:

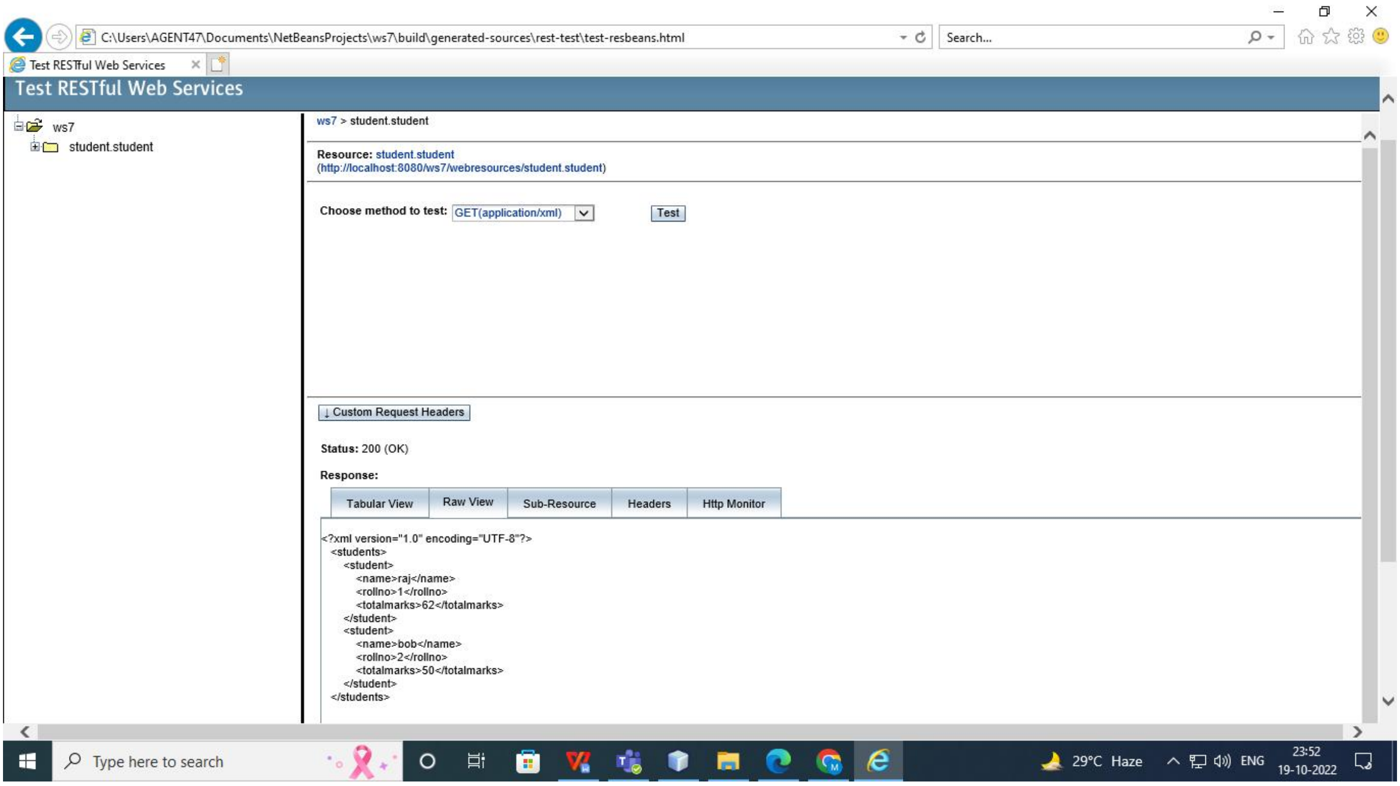
#	ROLLNO	NAME	TOTALMARKS
1	1	raj	62
2	2	bob	50

Output
Java DB Database Process
GlassFish Server 4.1
SQL 1 execution
ws5client (clean)
SQL 2 execution

```
Line 1, column 1

Execution finished after 0.135 s, no errors occurred.
```

SQL statement(s) executed successfully.

1:1
INS
Type here to search
29°C Haze
ENG
23:51
19-10-2022


C:\Users\AGENT47\Documents\NetBeansProjects\ws7\build\generated-sources\rest-test\test-resbeans.html
Search...
Test RESTful Web Services

Test RESTful Web Services

ws7 > student.student

Resource: student.student
(http://localhost:8080/ws7/webresources/student.student)

Choose method to test: GET(application/xml) Test

↓ Custom Request Headers

Status: 200 (OK)

Response:

Tabular View
Raw View
Sub-Resource
Headers
Http Monitor

```
<?xml version="1.0" encoding="UTF-8"?>
  <students>
    <student>
      <name>raj</name>
      <rollno>1</rollno>
      <totalmarks>62</totalmarks>
    </student>
    <student>
      <name>bob</name>
      <rollno>2</rollno>
      <totalmarks>50</totalmarks>
    </student>
  </students>
```

Type here to search
29°C Haze
ENG
23:52
19-10-2022

PRACTICAL 8

NAME: MAHARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM:Create a WCF service to perform calculations like Addition, Subtraction , Multiplication and Division. Create a client for WCF which invokes the various operations

WebApplication1.aspx.cs:

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

namespace WebApplication1
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
      ServiceReference1.Service1Client client = new ServiceReference1.Service1Client();
      double a = double.Parse(TextBox1.Text);
      double b = double.Parse(TextBox2.Text);
      TextBox3.Text = Convert.ToString(client.Sum(a, b));
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
      ServiceReference1.Service1Client client = new ServiceReference1.Service1Client();
      double a = double.Parse(TextBox1.Text);
      double b = double.Parse(TextBox2.Text);
      TextBox3.Text = Convert.ToString(client.Diff(a, b));
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
      ServiceReference1.Service1Client client = new ServiceReference1.Service1Client();
      double a = double.Parse(TextBox1.Text);
      double b = double.Parse(TextBox2.Text);
      TextBox3.Text = Convert.ToString(client.Mul(a, b));
    }
    protected void Button4_Click(object sender, EventArgs e)
    {
      ServiceReference1.Service1Client client = new ServiceReference1.Service1Client();
      double a = double.Parse(TextBox1.Text);
      double b = double.Parse(TextBox2.Text);
      TextBox3.Text = Convert.ToString(client.Div(a, b));
    }
  }
}
```

WebApplication1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

```
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <div>
    <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    <br />
     <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Add" />
     <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Sub" />
    <asp:Button ID="Button3" runat="server" OnClick="Button3_Click" Text="Multiply" />
 <asp:Button ID="Button4" runat="server" OnClick="Button4_Click" Text="Div" />
 <br />
 <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
 </div>
    </form>
</body>
</html>
```

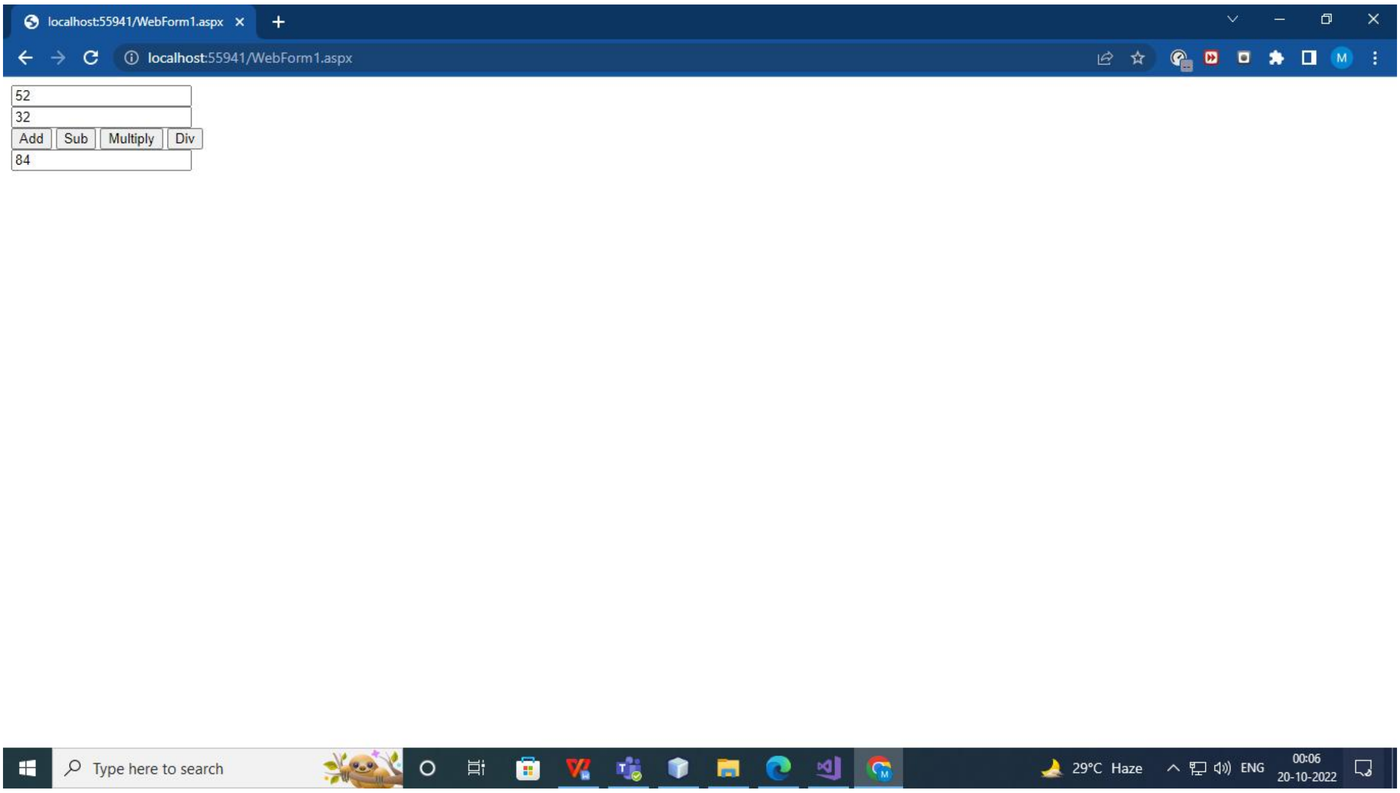
IService1.cs:

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

namespace wsprac8
{
// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name
"IService1" in both code and config file together.
 [ServiceContract]
 public interface IService1
    {
        [OperationContract]
        double Sum(double a, double b);
        [OperationContract]
        double Diff(double a, double b);
        [OperationContract]
        double Mul(double a, double b);
        [OperationContract]
        double Div(double a, double b);
    }
}
```

Service1.svc.cs:

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

namespace wsprac8
{
// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name
"IService1" in both code and config file together.
 [ServiceContract]
 public interface IService1
    {
        [OperationContract]
        double Sum(double a, double b);
        [OperationContract]
        double Diff(double a, double b);
        [OperationContract]
        double Mul(double a, double b);
        [OperationContract]
        double Div(double a, double b);
    }
}
```

OUTPUT:



52

32

84

PRACTICAL 9

NAME: MAHARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM:Create a WCF service with different endpoint for Soap based and Rest based implementation

IService1.cs:

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

namespace wsprac9
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name
"IService1" in both code and config file together.
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        [System.ServiceModel.Web.WebInvoke(Method = "GET", UriTemplate = "/Sayhello/{value}",
RequestFormat = System.ServiceModel.Web.WebMessageFormat.Json, ResponseFormat =
System.ServiceModel.Web.WebMessageFormat.Json)]
        string SayHello(string value);
    }
}
```

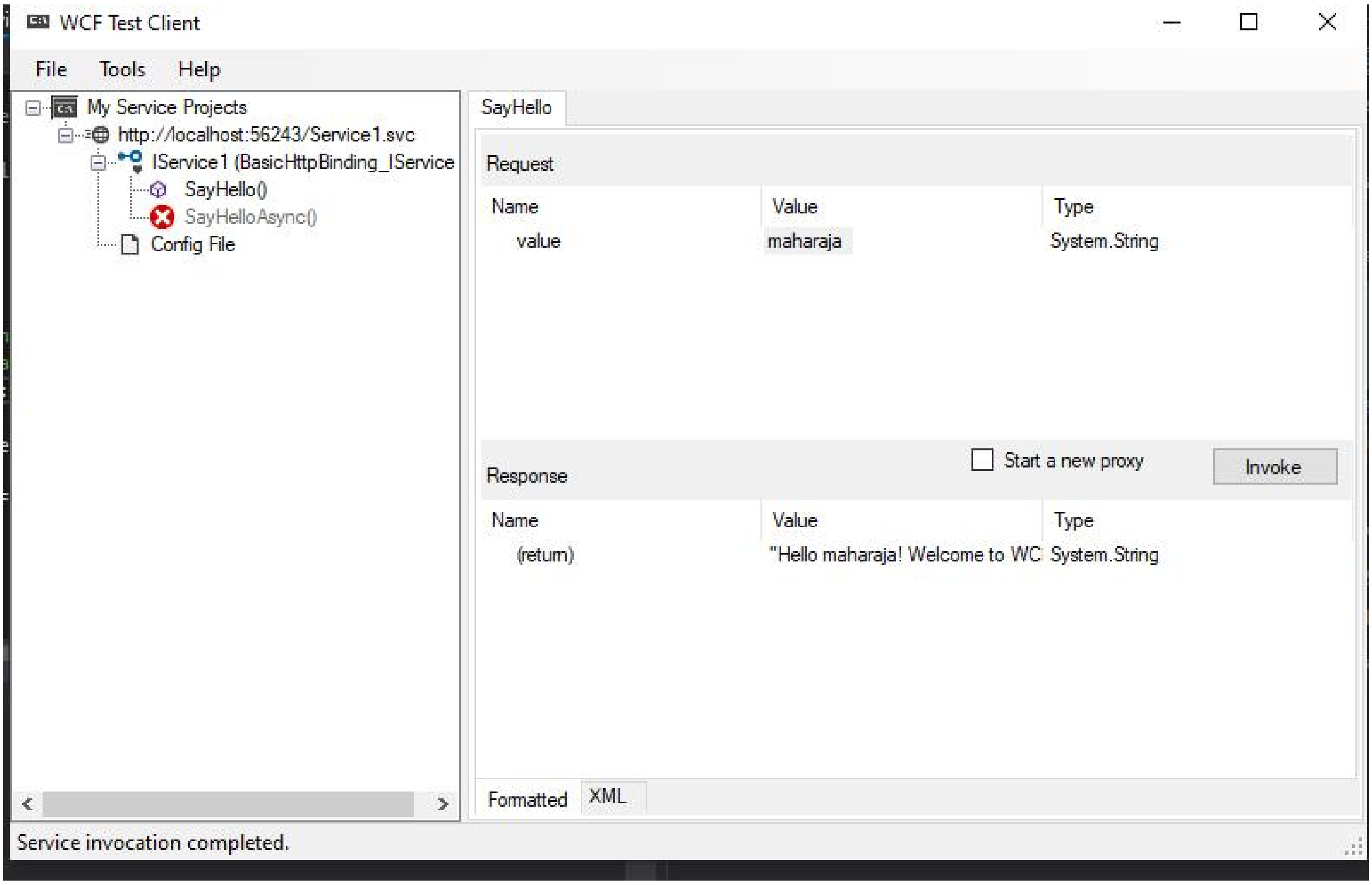
Service1.svc.cs:

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

namespace wsprac9
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name
"Service1" in code, svc and config file together.
    // NOTE: In order to launch WCF Test Client for testing this service, please select Service1.svc
or Service1.svc.cs at the Solution Explorer and start debugging.
    public class Service1 : IService1
    {
        public string SayHello(string value)
        {
            return string.Format($"Hello {value}! Welcome to WCF");
        }
    }
}
```

Web.config:

```
<?xml version="1.0"?>
<configuration>
 <appSettings>
 <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
 </appSettings>
 <system.web>
 <compilation debug="true" targetFramework="4.6.1" />
 <httpRuntime targetFramework="4.6.1"/>
 </system.web>
 <system.serviceModel>
 <services>
 <service name="WCFHelloApp.HelloService">
 <endpoint address="jsonservice" binding="webHttpBinding"
contract="WCFHelloApp.IHelloService" behaviorConfiguration="web">
 </endpoint>
 <endpoint address="soapservice" binding="basicHttpBinding"
contract="WCFHelloApp.IHelloService">
 </endpoint>
 </service>
 </services>
 <behaviors>
 <serviceBehaviors>
 <behavior>
 <!-- To avoid disclosing metadata information, set the values below to false before deployment -->
 <serviceMetadata httpGetEnabled="true" httpsGetEnabled="true"/>
 <!-- To receive exception details in faults for debugging purposes, set the value below to true.
Set to false before deployment to avoid disclosing exception information -->
 <serviceDebug includeExceptionDetailInFaults="false"/>
 </behavior>
 </serviceBehaviors>
 <endpointBehaviors>
 <behavior name="web">
 <webHttp/>
 </behavior>
 </endpointBehaviors>
 </behaviors>
 <protocolMapping>
 <add binding="basicHttpsBinding" scheme="https" />
 </protocolMapping>
 <serviceHostingEnvironment aspNetCompatibilityEnabled="true"
multipleSiteBindingsEnabled="true" />
 </system.serviceModel>
 <system.webServer>
 <modules runAllManagedModulesForAllRequests="true"/>
 <!--
 To browse web app root directory during debugging, set the value below to true.
 Set to false before deployment to avoid disclosing web app folder information.
 -->
 <directoryBrowse enabled="true"/>
 </system.webServer>
</configuration>
```

OUTPUT:



Request

Name	Value	Type
value	maharaja	System.String

Response

Name	Value	Type
(return)	"Hello maharaja! Welcome to WC	System.String

Service invocation completed.

PRACTICAL 10

NAME: MAHARAJA AAKASH THANGARAJ

ROLL: TCS2223038

AIM:To create a RESTful client for Practical No.7

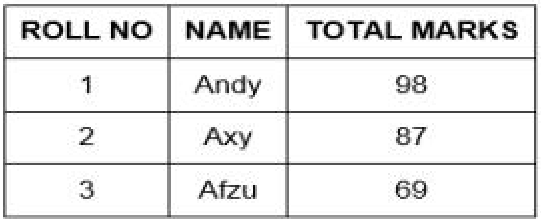
Index.html:

```
<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>
 <h1> </h1>
 <br>
 <input type="submit" formaction="getData.jsp" value="Get Data">
 </form>
 </body>
</html>
```

getData.jsp:

```
<%--
 Document : getData
 Created on : Oct 15, 2022, 7:17:22 PM
 Author : User
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
 <head>
<title>TODO supply a title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial- scale=1.0">
<style>
 table
 {
 font-family: arial, sans-serif;
 border-collapse: collapse;
 }
 td, th
 {
 border: 1px solid #000000;
 text-align: center;
 padding: 8px;
 }
</style>
<script>
var request = new XMLHttpRequest();
request.open('GET',
'http://localhost:8080/Student/webresources/stupack.student/', true);
request.onload = function ()
{
// begin accessing JSON data here
 var data = JSON.parse(this.response);
 for (var i = 0; i < data.length; i++)
 {
 var table = document.getElementById("myTable");
 var row = table.insertRow();
 var cell1 = row.insertCell(0);
 var cell2 = row.insertCell(1);
 var cell3 = row.insertCell(2);
 cell1.innerHTML = data[i].rollno;
 cell2.innerHTML = data[i].name;
 cell3.innerHTML = data[i].totalmarks;
 }
};
request.send();
</script>
</head>
<body>
<table id="myTable">
<tr>
<th>ROLL NO</th>
<th>NAME</th>
<th>TOTAL MARKS</th>
</tr>
</table>
</body>
</html>
```

OUTPUT:



ROLL NO	NAME	TOTAL MARKS
1	Andy	98
2	Axy	87
3	Afzu	69