And Date

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. no | Practical name | Pg. no | Sign |
| 1 | Create a TimeServer webservice in Java and Consume it in java and other technologies like php and .NET | 4 |  |
| 2 | Create a Java WS for performing basic calculations like addition, subtraction ,multiplication and Division and create a java client that consumes the same. | 7 |  |
| 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 | 13 |  |
| 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. | 21 |  |
| 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 w eb service. | 26 |  |
| 6 | Design a Restful webservice from a database table  Employee with columns empid,empname and Designation. Test the webservice for the various http requests | 34 |  |
| 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. | 38 |  |
| 8 | Create a WCF service to perform calculations like Addition, Subtraction , Multiplication and Division. Create a client for WCF which invokes the various operations. | 46 |  |
| 9 | Create a WCF service with different endpoint for Soap based and Rest based implementation. | 49 |  |

# PRAC – 1

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

1) Web Service :

package pack;

import java.util.Date; import javax.jws.WebService; import javax.jws.WebMethod; import javax.jws.WebParam;

@WebService(serviceName = "prac1")

public class prac1 {

@WebMethod(operationName = "getTime")

public String getTime() {

//TODO write your implementation code here:

return new Date().toString();

}

@WebMethod(operationName = "getTimeElapsed")

public long getTimeElapsed() {

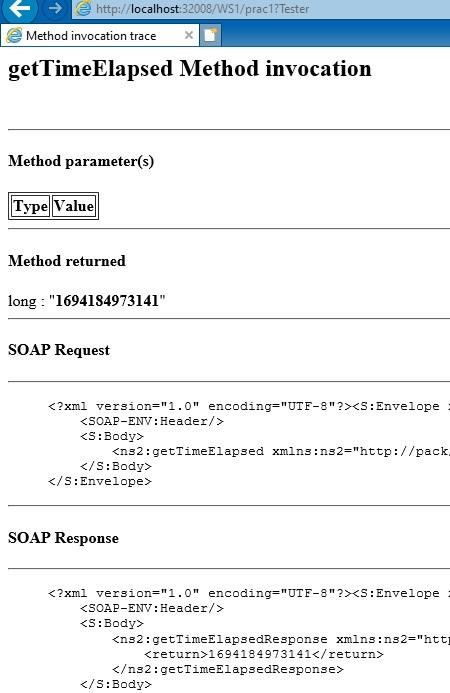
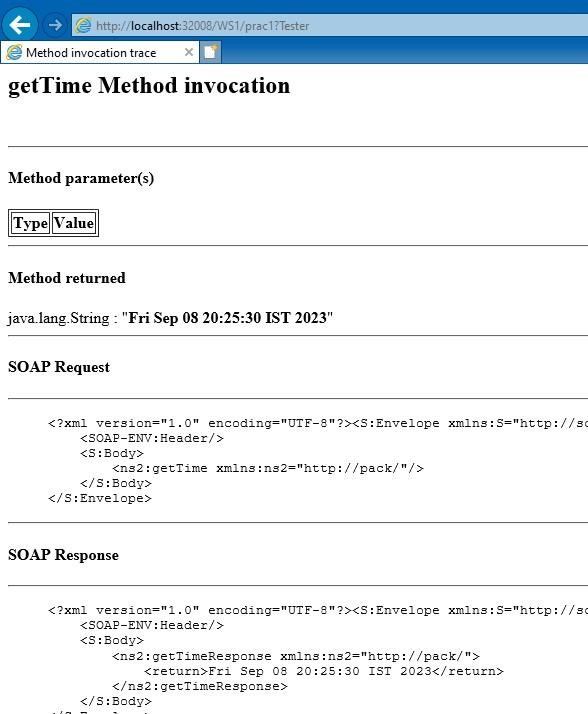
//TODO write your implementation code here:

return new Date().getTime();

}

}

Testing Web Service :



2) Web Service Client :

<%--

Document : newjsp

Created on : 7 Sep, 2023, 8:06:26 PM

Author : DELL

--%>

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

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>WS1</title>

</head>

<body>

<h1>

<%-- start web service invocation --%><hr/>

<%

try {

pack.Prac1\_Service service = new pack.Prac1\_Service(); pack.Prac1 port = service.getPrac1Port(); // TODO process result here java.lang.String result = port.getTime(); out.println("Result =

"+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

%>

<%-- end web service invocation --%><hr/>

<%-- start web service invocation --%><hr/>

<%

try {

pack.Prac1\_Service service = new pack.Prac1\_Service(); pack.Prac1 port = service.getPrac1Port(); // TODO process result here long result = port.getTimeElapsed(); out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

%>

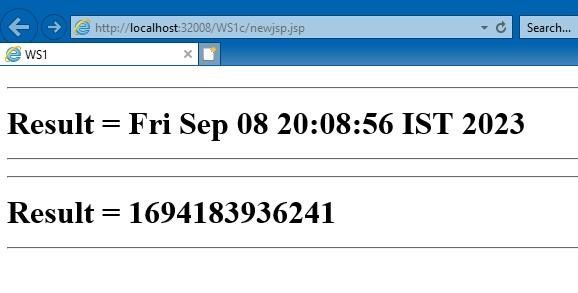
<%-- end web service invocation --%><hr/>

</h1>

</body>

</html>

**Output :**



# PRAC – 2

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

1) Web Service :

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

@WebService(serviceName = "CalService")

public class CalService {

@WebMethod(operationName = "Add")

public long Add(@WebParam(name = "num1") long num1, @WebParam(name = "num2") long num2) {

//TODO write your implementation code here:

long add = num1+num2;

return add;

}

@WebMethod(operationName = "Sub")

public long Sub(@WebParam(name = "num1") long num1, @WebParam(name = "num2") long num2) {

//TODO write your implementation code here:

long sub = num1-num2;

return sub;

}

@WebMethod(operationName = "Mul")

public long Mul(@WebParam(name = "num1") long num1, @WebParam(name = "num2") long num2) {

//TODO write your implementation code here:

long mul = num1\*num2;

return mul;

}

@WebMethod(operationName = "Div")

public long Div(@WebParam(name = "num1") long num1, @WebParam(name = "num2") long num2) {

//TODO write your implementation code here:

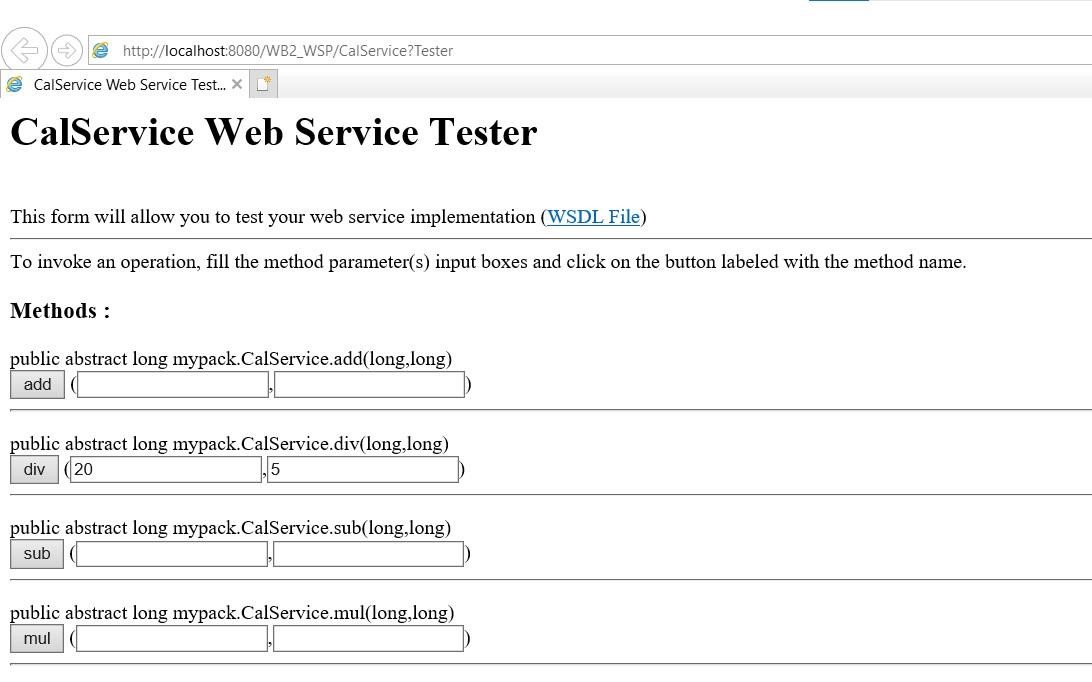
long div = num1/num2;

return div;

}

}

Testing Web Service :



2) Web Service Client :

## 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 action="Calculate.jsp">

Enter num1: <input type="text" name="num1" value="" /> <br> <br>

Enter num2: <input type="text" name="num2" value="" /> <br> <br>

<input type="radio" name="opr" value="add" />Add

<input type="radio" name="opr" value="sub" />Subtract

<input type="radio" name="opr" value="mul" />Multiply

<input type="radio" name="opr" value="div" />Divide <br> <br>

<input type="submit" value="Calculate" />

</form>

</body>

</html>

## Calculate.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>

<%

Integer first=Integer.parseInt(request.getParameter("num1"));

Integer second=Integer.parseInt(request.getParameter("num2"));

String Choice = request.getParameter("opr"); if("add".equals(Choice))

{

try {

mypack.CalService\_Service service = new mypack.CalService\_Service(); mypack.CalService port = service.getCalServicePort(); long result = port.add(first, second); out.println("Result =

"+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

}

else if("sub".equals(Choice))

{

try {

mypack.CalService\_Service service = new mypack.CalService\_Service(); mypack.CalService port = service.getCalServicePort(); long result = port.sub(first, second); out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

}

else if("mul".equals(Choice))

{

try {

mypack.CalService\_Service service = new mypack.CalService\_Service(); mypack.CalService port = service.getCalServicePort(); long result = port.mul(first, second); out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

}

else if("div".equals(Choice))

{

try {

mypack.CalService\_Service service = new mypack.CalService\_Service(); mypack.CalService port = service.getCalServicePort(); long result = port.div(first, second); out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

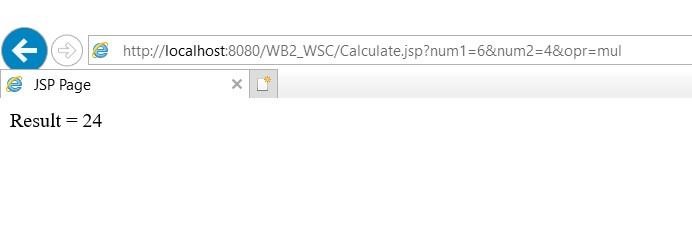
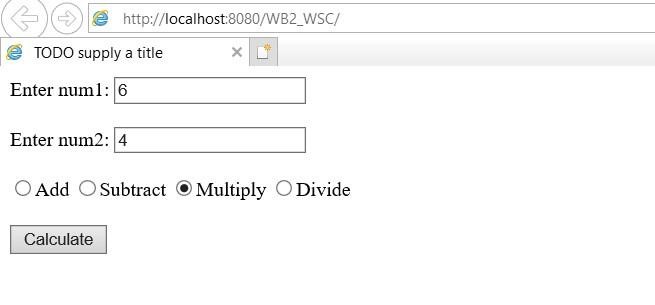
}

%>

</body>

</html>

**Output :**



# PRAC – 3

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 pageSource Code :

1) Web Service :

/\*

* 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 pack;

import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet; import java.sql.Statement; import javax.jws.WebService; import javax.jws.WebMethod; import javax.jws.WebParam;

/\*\*

\*

* @author DELL

\*/

@WebService(serviceName = "prac3")

public class prac3 {

/\*\*

* Web service operation

\*/

@WebMethod(operationName = "getNSE")

public int getNSE() {

//TODO write your implementation code here:

int nse=0;

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con =

DriverManager.getConnection("jdbc:derby://localhost:1527/Stockdb", "rohan”,"rohan");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM STOCKDATA");

rs.next(); nse = rs.getInt("NSE");

con.close();

}

catch(Exception e)

{}

return nse;

}

/\*\*

\* Web service operation

\*/

@WebMethod(operationName = "getBSE")

public int getBSE() {

//TODO write your implementation code here:

int bse=0;

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con =

DriverManager.getConnection("jdbc:derby://localhost:1527/Stockdb", "rohan","rohan");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM STOCKDATA");

rs.next(); bse = rs.getInt("BSE");

con.close();

}

catch(Exception e)

{}

return bse;

}

/\*\*

\* Web service operation

\*/

@WebMethod(operationName = "getGOLDRATE")

public int getGOLDRATE() {

//TODO write your implementation code here:

int goldrate=0;

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

// Connection Creation

Connection con =

DriverManager.getConnection("jdbc:derby://localhost:1527/Stockdb", "rohan","rohan");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM STOCKDATA");

rs.next();

goldrate = rs.getInt("GOLDRATE");

con.close();

}

catch(Exception e)

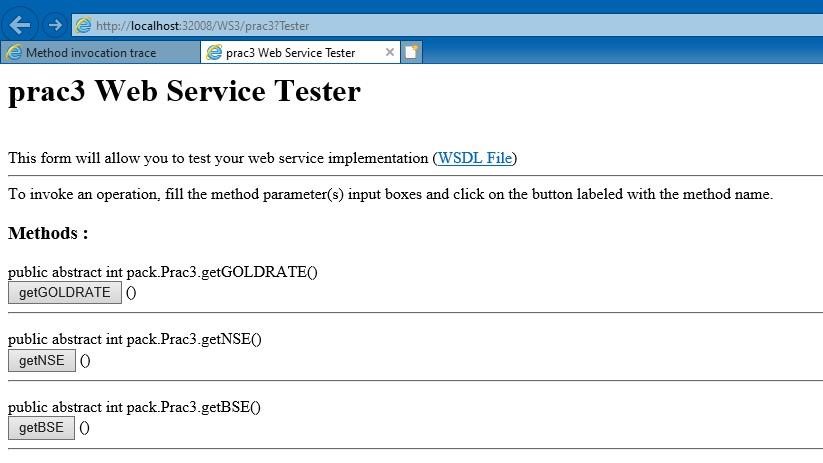
{}

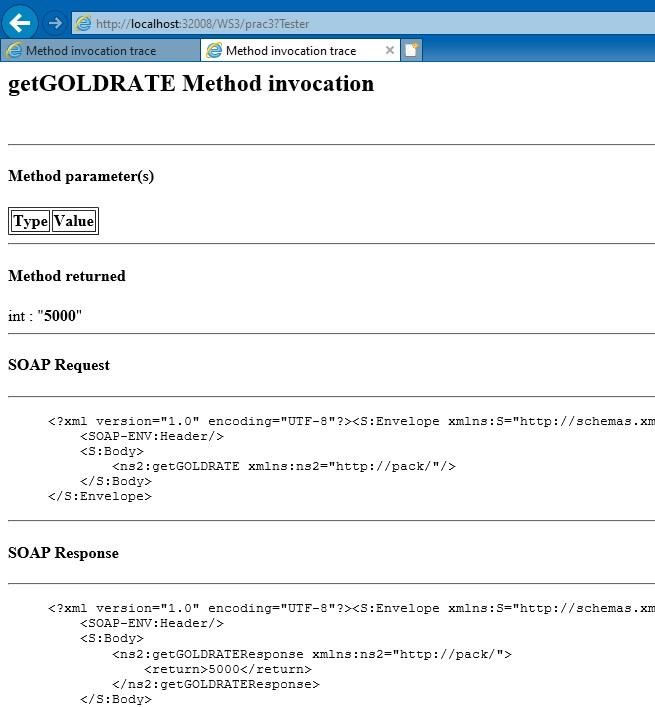
return goldrate;

}

}

Testing Web Service :





2) Web Service Client :

## 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 action="StockApp.jsp">

Select one of the following:<br>

Get NSE Index<input type="radio" name="stock" value="nse" /><br><br> Get BSE Index<input type="radio" name="stock" value="bse" /><br><br>

Get GoldRate<input type="radio" name="stock" value="gold" /><br><br>

<input type="submit" value="Submit" /><br>

</form>

</body>

</html>

## StockApp.jsp

<%--

Document : StockApp

Created on : 8 Sep, 2023, 5:48:46 PM

Author : DELL

--%>

<%@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 choice=request.getParameter("stock");

if (choice.equals("nse"))

{

try {

pack.Prac3\_Service service = new pack.Prac3\_Service(); pack.Prac3 port = service.getPrac3Port(); // TODO process result here int result = port.getNSE(); out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

}

else if(choice.equals("bse"))

{

try {

pack.Prac3\_Service service = new pack.Prac3\_Service(); pack.Prac3 port = service.getPrac3Port(); // TODO process result here int result = port.getBSE(); out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

}

else if(choice.equals("gold"))

{

try {

pack.Prac3\_Service service = new pack.Prac3\_Service(); pack.Prac3 port = service.getPrac3Port(); // TODO process result here int result = port.getGOLDRATE(); out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

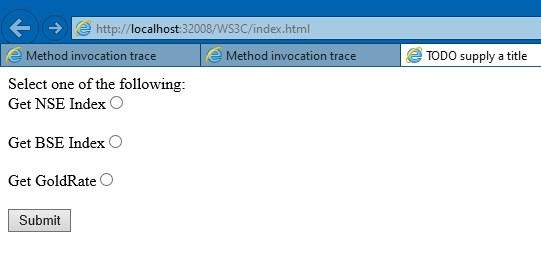
}

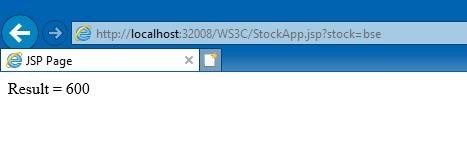
%>

</body>

</html>

**Output :**





# PRAC – 4

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. Source Code :

1) Web Service :

/\*

* 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 pack;

import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet; import java.sql.Statement; import javax.jws.WebService; import javax.jws.WebMethod; import javax.jws.WebParam;

/\*\*

\*

* @author DELL

\*/

@WebService(serviceName = "prac4")

public class prac4 {

/\*\*

* Web service operation

\*/

@WebMethod(operationName = "getRating") public String getRating(@WebParam(name = "colName") String colName) {

//TODO write your implementation code here: String rating=null;

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

// Connection Creation

Connection con =

DriverManager.getConnection("jdbc:derby://localhost:1527/Ratedb", "rohan”,"rohan");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM RATINGS WHERE COLLEGENAME='"+colName+"' ");

rs.next(); rating= rs.getString("RATING");

con.close();

}

catch(Exception e)

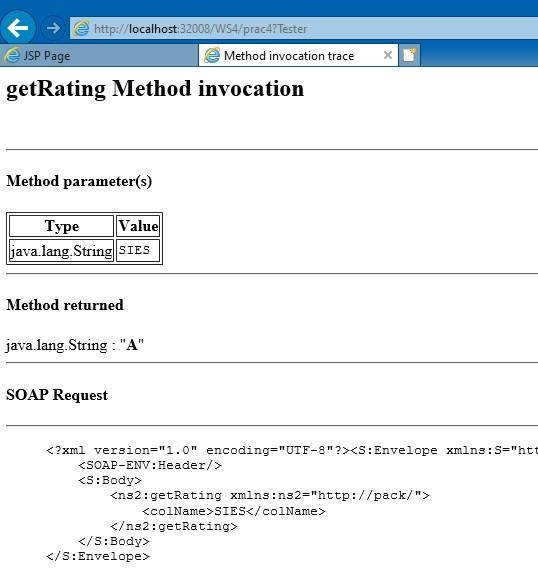
{}

return rating;

}

}

Testing Web Service :



2) Web Service Client :

## 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 action = "Rate.jsp">

ENTER THE COLLEGE: <input type="text" name="rate" value="" /><BR><BR>

<input type="submit" value="Check Rating" />

</form>

</body>

</html>

## Rate.jsp

<%--

Document : Rate

Created on : 8 Sep, 2023, 6:52:22 PM

Author : DELL

--%>

<%@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 clgname=request.getParameter("rate");

try {

pack.Prac4\_Service service = new pack.Prac4\_Service(); pack.Prac4 port = service.getPrac4Port(); // TODO initialize WS operation arguments here java.lang.String colName = clgname; // TODO process result here java.lang.String result = port.getRating(colName); out.println("College Name:"+clgname);

out.println("; ");

out.println("NAAC RATING= "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

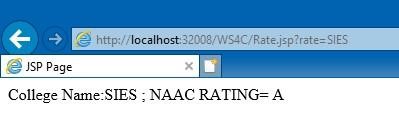
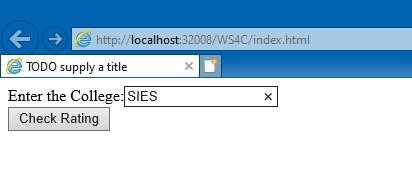
}

%>

</body>

</html>

**Output :**



# PRAC – 5

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. Source Code :

1) Web Service :

/\*

* 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 pack;

import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet; import java.sql.Statement; import javax.jws.WebService; import javax.jws.WebMethod; import javax.jws.WebParam;

/\*\*

\*

* @author DELL

\*/

@WebService(serviceName = "prac5")

public class prac5 {

/\*\*

* Web service operation

\*/

@WebMethod(operationName = "getBreakingNews") public String getBreakingNews(@WebParam(name = "Date") String Date) {

//TODO write your implementation code here: String date=null;

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

// Connection Creation

Connection con =

DriverManager.getConnection("jdbc:derby://localhost:1527/Channeldb", "rohan","rohan");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM BREAKINGNEWS WHERE DATE='"+Date+"' ");

rs.next();

date= rs.getString("NEWS");

con.close();

}

catch(Exception e)

{}

return date;

}

/\*\*

\* Web service operation

\*/

@WebMethod(operationName = "getPrediction") public String getPrediction(@WebParam(name = "SunSign") String SunSign) {

//TODO write your implementation code here: String sign=null;

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

// Connection Creation

Connection con =

DriverManager.getConnection("jdbc:derby://localhost:1527/Channeldb", "rohan","rohan"); Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM PREDICTION WHERE SUNSIGN='"+SunSign+"' ");

rs.next(); sign= rs.getString("PREDICT");

con.close();

}

catch(Exception e)

{}

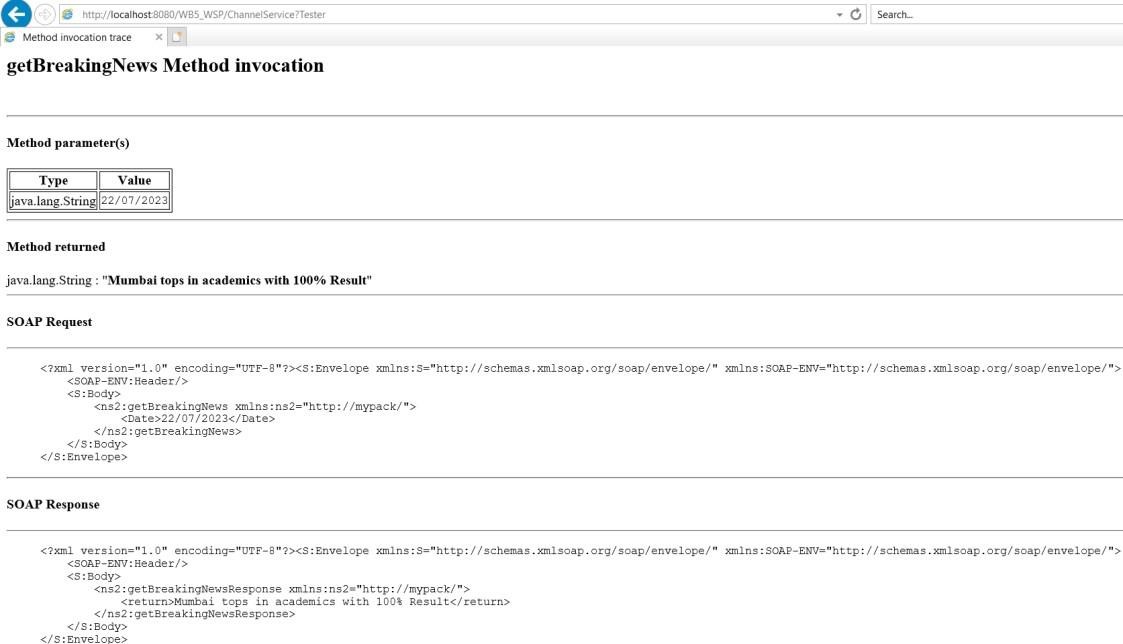
return sign;

}

}

Testing Web Service :





2) Web Service Client :

## 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 action = "Channel.jsp">

ENTER THE DATE FOR NEWS: <input type="text" name="date" value="" /><BR><BR>

ENTER THE SUNSIGN : <input type="text" name="sign" value="" /><BR><BR>

<input type="submit" value="Check " />

</form>

</body>

</html>

## Channel.jsp

<%--

Document : Channel

Created on : 8 Sep, 2023, 7:36:05 PM

Author : DELL

--%>

<%@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 News=request.getParameter("date");

try {

pack.Prac5\_Service service = new pack.Prac5\_Service(); pack.Prac5 port = service.getPrac5Port(); // TODO initialize WS operation arguments here java.lang.String date = News; // TODO process result here java.lang.String result = port.getBreakingNews(date); out.println("News

is "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

%>

<br>

<%

String Predict=request.getParameter("sign");

try {

pack.Prac5\_Service service = new pack.Prac5\_Service(); pack.Prac5 port = service.getPrac5Port(); // TODO initialize WS operation arguments here java.lang.String sunSign = Predict; // TODO process result here java.lang.String result = port.getPrediction(sunSign); out.println("Prediction is

"+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

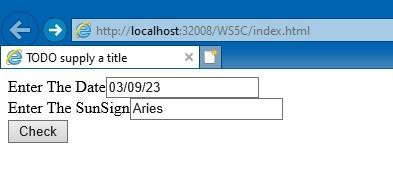
}

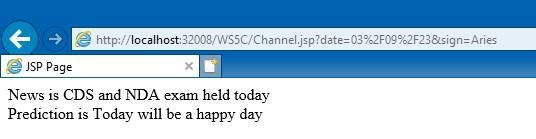
%>

</body>

</html>

**Output :**





# PRAC – 6

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

1) EmployeeFacadeRest:

/\*

* 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 pack.service;

import java.util.List; import javax.ejb.Stateless; import javax.persistence.EntityManager; import javax.persistence.PersistenceContext; import javax.ws.rs.Consumes; import javax.ws.rs.DELETE; import javax.ws.rs.GET; import javax.ws.rs.POST; import javax.ws.rs.PUT; import javax.ws.rs.Path; import javax.ws.rs.PathParam; import javax.ws.rs.Produces; import javax.ws.rs.core.MediaType; import pack.Emp;

/\*\*

\*

\* @author DELL

\*/

@Stateless @Path("pack.emp") public class EmpFacadeREST extends AbstractFacade<Emp> {

@PersistenceContext(unitName = "WS6PU") private EntityManager em;

public EmpFacadeREST() {

super(Emp.class);

}

@POST

@Override

@Consumes({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON}) public void create(Emp entity) { super.create(entity);

}

@PUT

@Path("{id}")

@Consumes({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public void edit(@PathParam("id") Integer id, Emp entity) {

super.edit(entity);

}

@DELETE

@Path("{id}") public void remove(@PathParam("id") Integer id) { super.remove(super.find(id));

}

@GET

@Path("{id}")

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public Emp find(@PathParam("id") Integer id) {

return super.find(id);

}

@GET

@Override

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public List<Emp> findAll() { return super.findAll();

}

@GET

@Path("{from}/{to}")

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public List<Emp> findRange(@PathParam("from") Integer from, @PathParam("to") Integer to)

{

return super.findRange(new int[]{from, to});

}

@GET

@Path("count")

@Produces(MediaType.TEXT\_PLAIN)

public String countREST() { return

String.valueOf(super.count());

}

@Override protected EntityManager getEntityManager() {

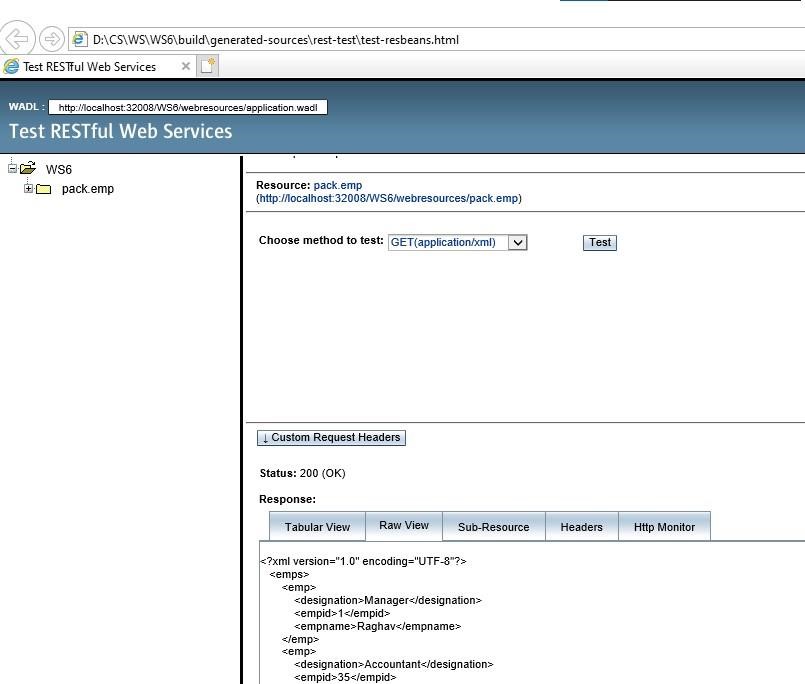
return em;

}

}

Testing Web Service :

## 1) GET(Application/xml)



# PRAC – 7

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. Source Code :

1) StudFacadeRest

/\*

* 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 pack.service;

import java.util.List; import javax.ejb.Stateless; import javax.persistence.EntityManager; import javax.persistence.PersistenceContext; import javax.ws.rs.Consumes; import javax.ws.rs.DELETE; import javax.ws.rs.GET; import javax.ws.rs.POST; import javax.ws.rs.PUT; import javax.ws.rs.Path; import javax.ws.rs.PathParam; import javax.ws.rs.Produces; import javax.ws.rs.core.MediaType; import pack.Stud;

/\*\*

\*

\* @author DELL

\*/

@Stateless @Path("pack.stud") public class StudFacadeREST extends AbstractFacade<Stud> {

@PersistenceContext(unitName = "WS7PU") private EntityManager em;

public StudFacadeREST() {

super(Stud.class);

}

@POST

@Override

@Consumes({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public void create(Stud entity) { super.create(entity);

}

@PUT

@Path("{id}")

@Consumes({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public void edit(@PathParam("id") Integer id, Stud entity) {

super.edit(entity);

}

@DELETE

@Path("{id}") public void remove(@PathParam("id") Integer id) { super.remove(super.find(id));

}

@GET

@Path("{id}")

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public Stud find(@PathParam("id") Integer id) {

return super.find(id);

}

@GET

@Override

@Produces({ MediaType.APPLICATION\_JSON})

public List<Stud> findAll() { return super.findAll();

}

@GET

@Path("{from}/{to}")

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public List<Stud> findRange(@PathParam("from") Integer from, @PathParam("to") Integer to)

{

return super.findRange(new int[]{from, to});

}

@GET

@Path("count")

@Produces(MediaType.TEXT\_PLAIN)

public String countREST() { return

String.valueOf(super.count());

}

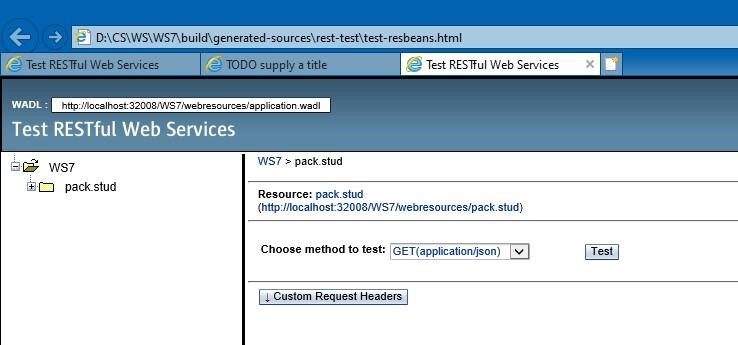
@Override protected EntityManager getEntityManager() {

return em;

}

}

Testing Web Service :



2) Web Service Client :

## 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 action="Student.jsp">

<input type="submit" name="submit" value="GET STUDENTS">

</form>

</body>

</html>

## StuTable.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>

<style> table{

font-family: arial,sans-serif;

border-collapse: collapse;

}

th,td{ text-

align:center; border:1px solid #000000;

padding:8px;

}

</style> <script>

var request = new XMLHttpRequest();

request.open('GET','http://localhost:21900/WB7\_WSP/webresources/mypack.student',true)

;

request.onload=function()

{

var data=JSON.parse(this.response);

for(var i=0;i<data.length;i++)

{

var table = document.getElementById("StudTable");

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="StudTable">

<tr>

<th>Rollno</th>

<th>Name</th>

<th>Marks</th>

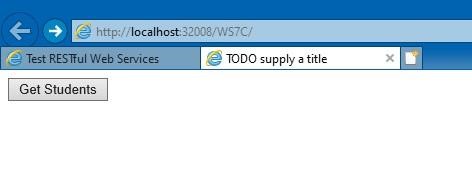
</tr>

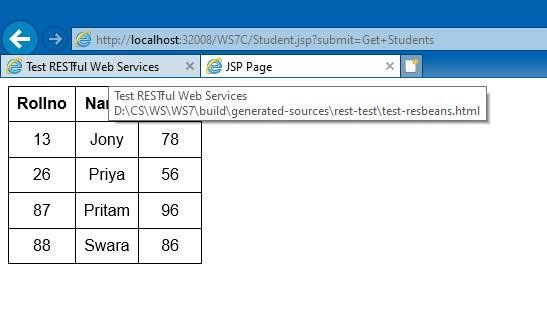
</table>

</body>

</html>

**Output :**



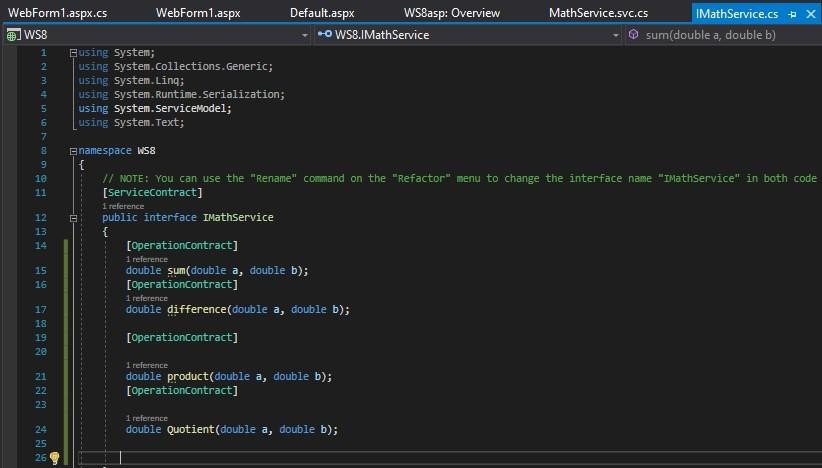


# PRAC – 8

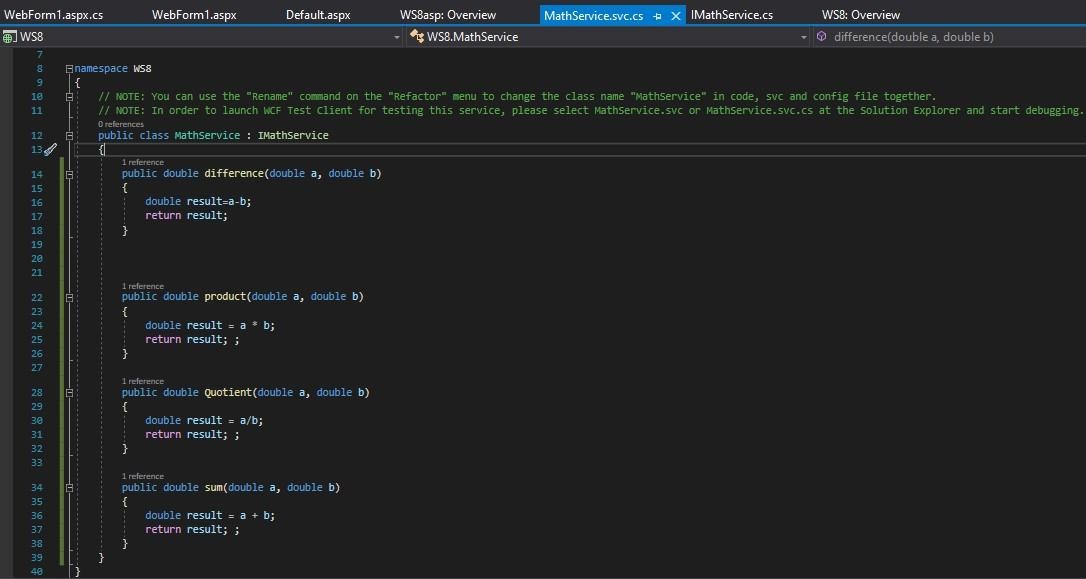
Aim : Create a WCF service to perform calculations like Addition, Subtraction , Multiplication and Division. Create a client for WCF which invokes the various operations.

Source Code :

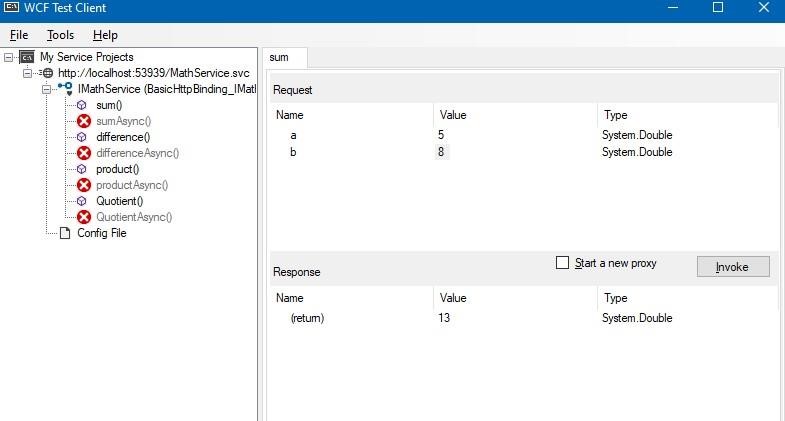
**IMathService.cs (Interface)**



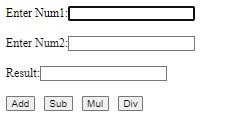
**MathService.svc.cs (Implementation)**



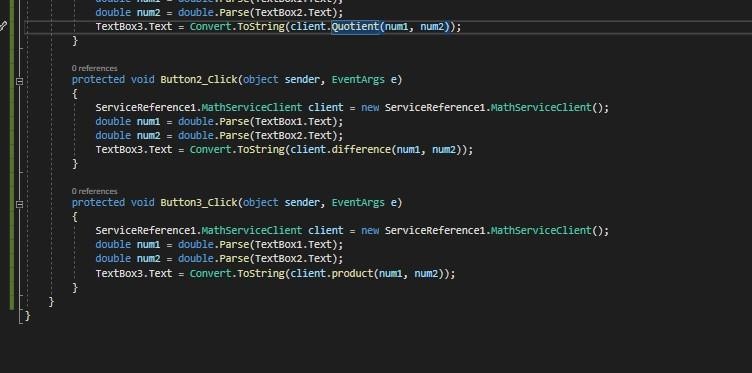
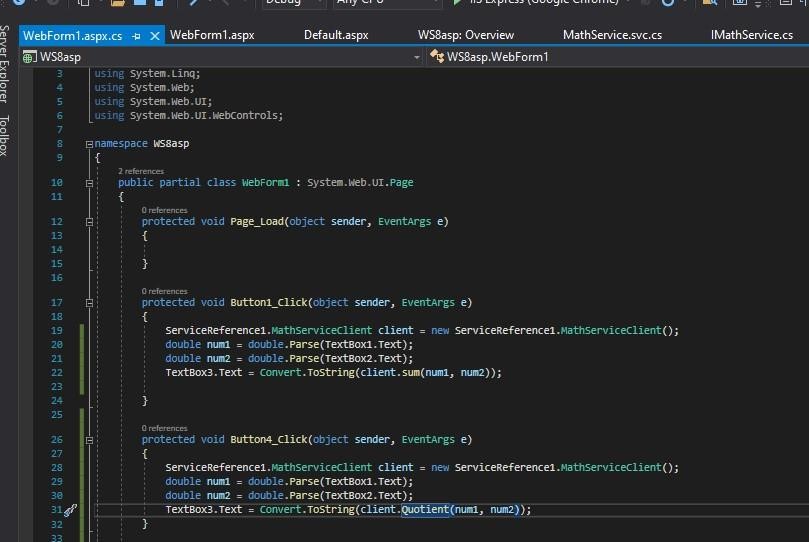
Testing Web Service :



Web Service Client : **MathClient.aspx:**

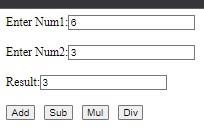
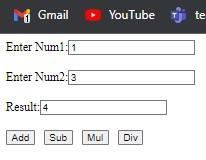


**MathClient.aspx.cs:**

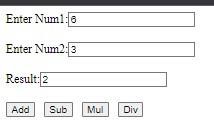
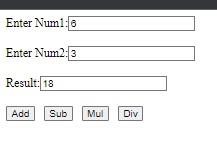


**Output :**

**Addition: Subtraction:**



**Multiplication: Division:**

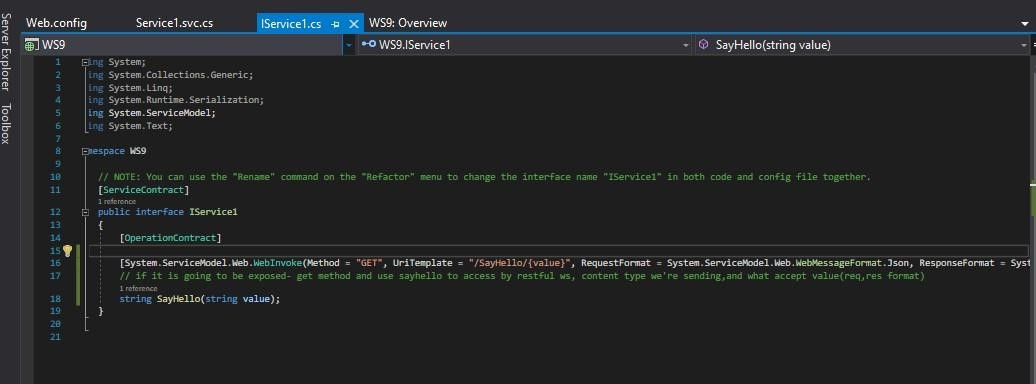


# PRAC – 9

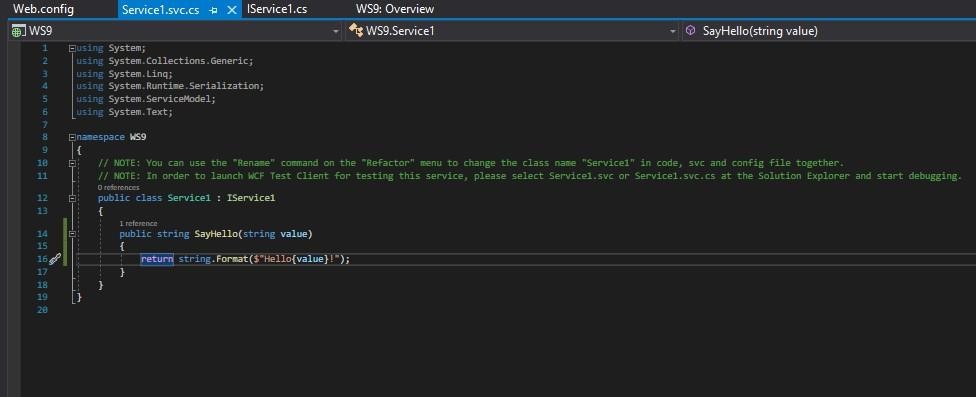
Aim : Create a WCF service with different endpoint for Soap based and Rest based implementation Source Code :

1) Web Service :

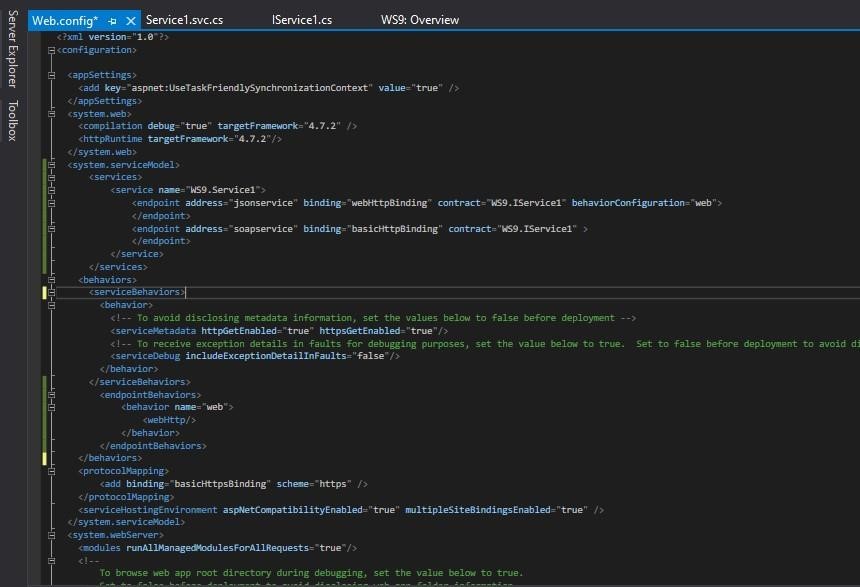
**IService1.cs**



**Service1.svc.cs**



**Web.config**



Testing Web Service :