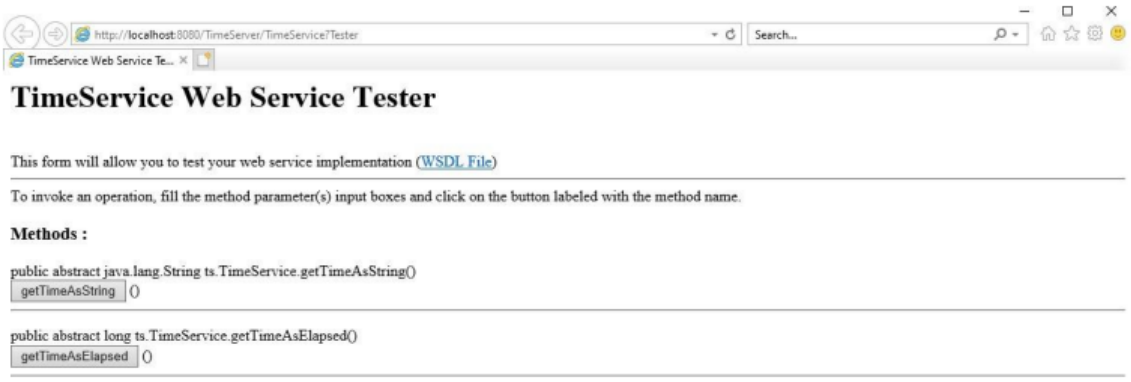
	returns special news of that day, 2nd function called getPrediction accepts sunsign name as string parameter and returns predictions as string. Design a client to test the above web service.		
6	Design a Restful webservice from a database table Employee with columns empid,empname and Designation. Test the webservice for the various http requests	56	
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.	64	
8	Create a WCF service to perform calculations like Addition, Subtraction, Multiplication and Division. Create a client for WCF which invokes the various operations.	71	
9	Create a WCF service with different endpoint for Soap based and Rest based implementation	80	
10	Design a Restful webservice and create a restful client that displays the data by fetching using the HTTP GET method	85	

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

```java
package timeserver;

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

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

    /**
     * Web service operation
     */
    @WebMethod(operationName = "operation1")
    public String getTimeAsString() {
        //TODO write your implementation code here:
        return new Date().toString();

    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "operation2")
    public long getTimeAsElapsed() {
        //TODO write your implementation code here:
        return new Date().getTime();
    }
}
```



Hello World!

Result = Sun Sep 10 22:27:55 IST 2023

Result = 1694365075332

operation1 Method invocation

Method parameter(s)

Type	Value

Method returned

java.lang.String : "**Sun Sep 10 22:28:40 IST 2023**"

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:operation1 xmlns:ns2="http://timeserver/"/>
    </S:Body>
</S:Envelope>
```


SOAP Response

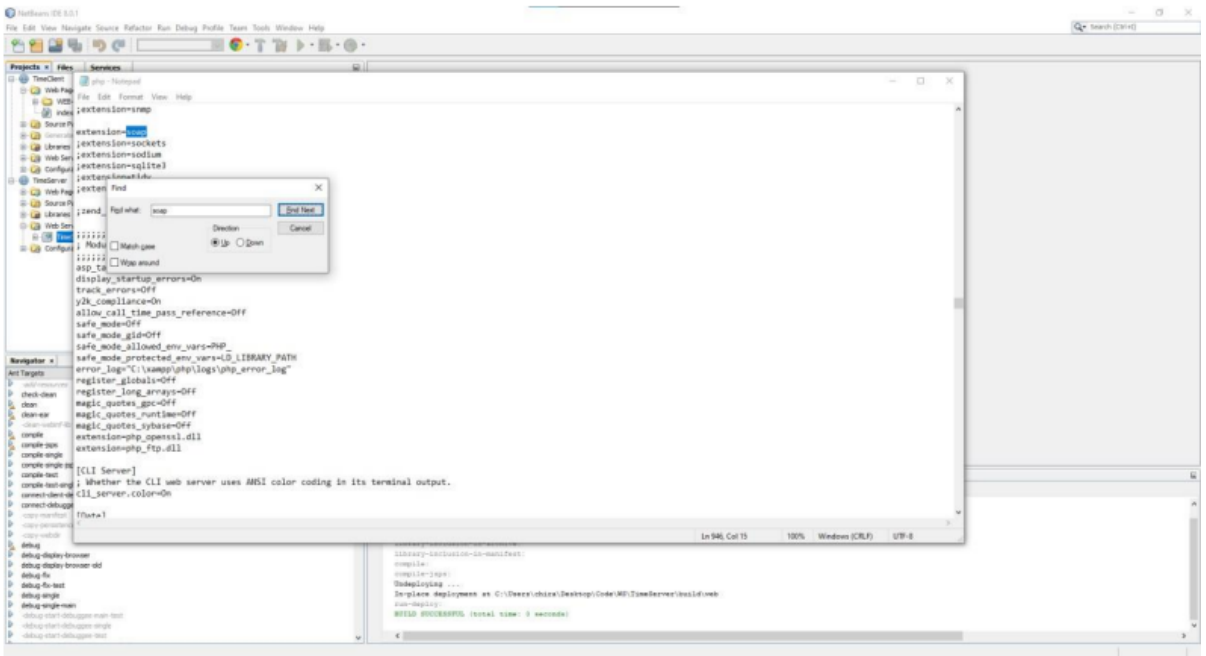

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:operation1Response xmlns:ns2="http://timeserver/">
            <return>Sun Sep 10 22:28:40 IST 2023</return>
        </ns2:operation1Response>
    </S:Body>
</S:Envelope>
```


.NET

```
using System;

namespace WSPrac1
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            ServiceReference1.TimeServiceClient client = new ServiceReference1.TimeServiceClient();
            Console.WriteLine(value: "Time as String =" + client.getTimeAsString());
            Console.WriteLine(value: "Time Elapsed =" + client.getTimeAsElapsed());
            Console.Read();
        }
    }
}
```

```
C:\Users\chira\Desktop\Code\WS\dotNetTimeClient\dotNetTimeClient\bin\Debug\net6.0\dotNetTimeClient.exe
Time as String =Thu Jul 28 23:30:31 IST 2022
Time Elapsed = 1659031231466
```

.php


;extension=snmp
extension=soap
;extension=sockets
;extension=sodium
;extension=sqlite3
find
Find what: soap
Find Next
Cancel
Direction
Up
Down
Match case
Wrap around
display_startup_errors=On
track_errors=Off
y2k_compliance=On
allow_call_time_pass_reference=Off
safe_mode=Off
safe_mode_gid=Off
safe_mode_allowed_env_vars=PHP_
safe_mode_protected_env_vars=LD_LIBRARY_PATH
error_log="C:\xampp\php\logs\php_error_log"
register_globals=Off
register_long_arrays=Off
magic_quotes_gpc=Off
magic_quotes_runtime=Off
magic_quotes_sybase=Off
extension=php_openssl.dll
extension=php_ftp.dll
[CLI Server]
; Whether the CLI web server uses ANSI color coding in its terminal output.
cli_server.color=On

Code:

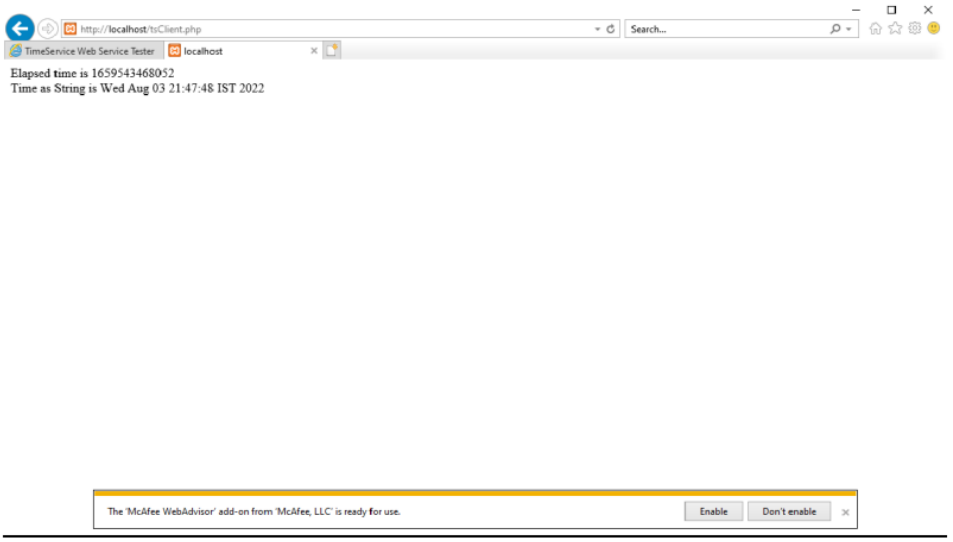
<?php

```
$client = new
SoapClient("http://localhost:
8080/TimeServer/TimeServi
ce?WSDL");
$t1 = $client ->
getTimeAsElapsed();
echo "Elapsed time is ",$t1
-> return;
```

7

```
$t2 = $client ->
getTimeAsString();
echo "<br>Time as String is
",$t2 -> return;
```

?>



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

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

/**
 *
 * @author Dinesh
 */
@WebService(serviceName = "CalcService")
public class CalcService {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "Additin")
    public float Additin(@WebParam(name = "x") float x, @WebParam(name = "y") float y) {
        //TODO write your implementation code here:
        return x + y;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "Substraction")
    public float Substraction(@WebParam(name = "x") float x, @WebParam(name = "y") float y) {
        //TODO write your implementation code here:
        return x - y;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "Multiplication")
    public float Multiplication(@WebParam(name = "x") float x, @WebParam(name = "y") float y) {
        //TODO write your implementation code here:
        return x * y;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "Division")
    public float Division(@WebParam(name = "x") float x, @WebParam(name = "y") float y) {
        //TODO write your implementation code here:
        return x / y;
    }

}
```

calc.CalcService > Division

Operations (4) — Add Operation... | Remove Operation

Additin

Parameters	Output	Faults	Description
Parameter Name		Parameter Type	
x		float	
y		float	

Substraction

Parameters	Output	Faults	Description
Parameter Name		Parameter Type	
x		float	
y		float	

Multiplication

Parameters	Output	Faults	Description
Parameter Name		Parameter Type	
x		float	
y		float	

Division

Parameters	Output	Faults	Description
Parameter Name		Parameter Type	
x		float	
y		float	

```html
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="index.jsp">
            Enter first Number <input type="text" name="x" id="x" /><br>
            Enter Second Number<input type="text" name="y" id="y" /><br>
            <input type="radio" name="Raddiobtngrp" value="Add" />Add<br>
            <input type="radio" name="Raddiobtngrp" value="Sub" />Add<br>
            <input type="radio" name="Raddiobtngrp" value="Multiply" />Add<br>
            <input type="radio" name="Raddiobtngrp" value="Div" />Add<br>
            <input type="submit" value="Calculat" name="submit" />

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

```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>
    <%-- start web service invocation --%><hr/>
    <%
        int a = Integer.parseInt(request.getParameter("x"));
        int b = Integer.parseInt(request.getParameter("y"));
        String choice = request.getParameter("RadiobtnGroup");

        if (choice.equals("Add")) {
            calc.CalcService_Service service = new calc.CalcService_Service();
            calc.CalcService port = service.getCalcServicePort();

            // TODO process result here
            float result = port.additin(a, b);
            out.println("Result = " + result);
        } else if (choice.equals("Sub")) {
            calc.CalcService_Service service = new calc.CalcService_Service();
            calc.CalcService port = service.getCalcServicePort();

            // TODO process result here
            float result = port.substraction(a, b);
            out.println("Result = " + result);
        } else if (choice.equals("Multiply")) {
            calc.CalcService_Service service = new calc.CalcService_Service();
            calc.CalcService port = service.getCalcServicePort();

            // TODO process result here
            float result = port.multiplication(a, b);
            out.println("Result = " + result);
        } else if (choice.equals("Div")) {
            calc.CalcService_Service service = new calc.CalcService_Service();
            calc.CalcService port = service.getCalcServicePort();

            // TODO process result here
            float result = port.division(a, b);
            out.println("Result = " + result);
        }
    %>
</body>
</html>
```

CalcService 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 float calc.CalcService.substraction(float,float)
substraction (,)

public abstract float calc.CalcService.multiplication(float,float)
multiplication (,)

public abstract float calc.CalcService.additin(float,float)
additin (,)

public abstract float calc.CalcService.division(float,float)
division (,)

multiplication Method invocation

Method parameter(s)

Type	Value
float	53
float	43

Method returned

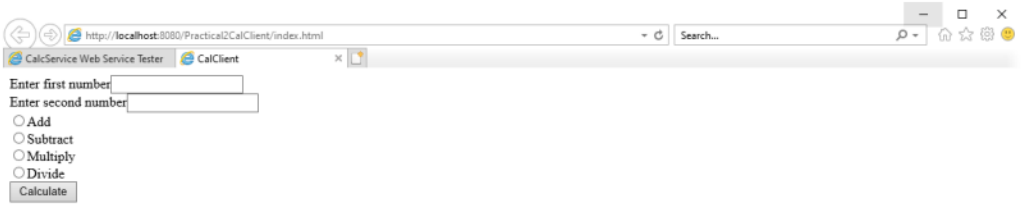
float : "**2279.0**"

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:Multiplication xmlns:ns2="http://calc/">
            <x>53.0</x>
            <y>43.0</y>
        </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://calc/">
            <return>2279.0</return>
```



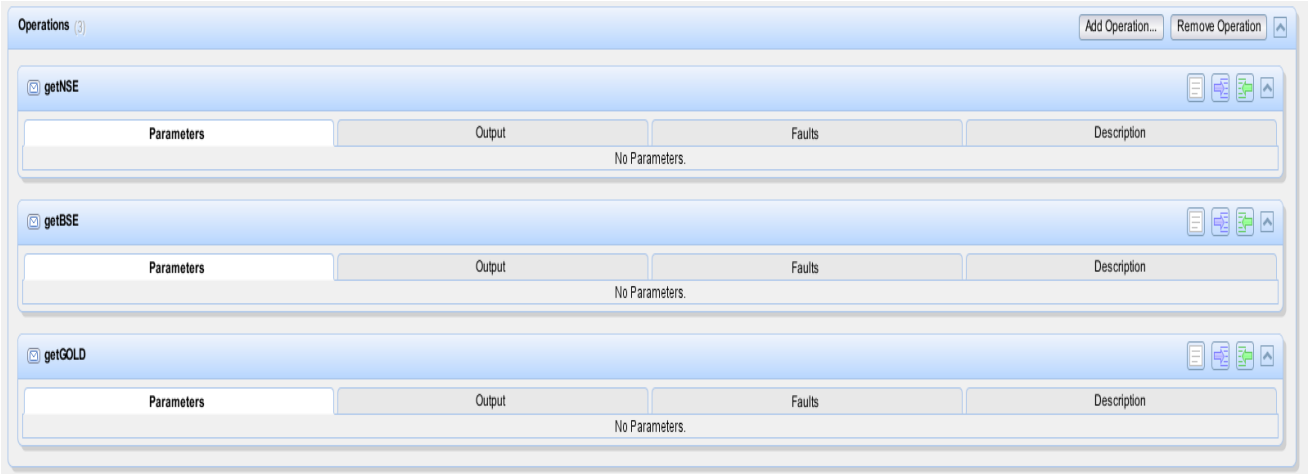
3. Create a web service that gives – (i) NSE Index, (ii) BSE Index, (iii)Gold Rate. The values are stored in database. Also create a web client for a share trading firm that displays these values on its home page

```
package stock;

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

    /**
     * Web service operation
     */
    @WebMethod(operationName = "getNSE")
    public long getNSE()  {
        long nse = 0;
        try{
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/StockDB
","user1","user@123");
            Statement smt=con.createStatement();
            ResultSet rs=smt.executeQuery("SELECT * FROM
STOCKDATA");
            rs.next();
            nse=rs.getInt("NSE");
       }
        catch(Exception e){
```

```
            e.printStackTrace();
        }
        //TODO write your implementation code here:
        return nse;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "getBSE")
    public long getBSE() {
        long bse = 0;
        try{
             Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/StockDB
","user1","user@123");
            Statement smt=con.createStatement();
            ResultSet rs=smt.executeQuery("SELECT * FROM
STOCKDATA");
            rs.next();
            bse=rs.getInt("BSE");


        }
        catch(Exception e){
             e.printStackTrace();
        }
        //TODO write your implementation code here:
        return bse;
    }
```

```
    /**
     * Web service operation
     */
    @WebMethod(operationName = "getGOLD")
    public long getGOLD()  {
        long gold = 0;
        try{
        Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/StockDB
","user1","user@123");
            Statement smt=con.createStatement();
            ResultSet rs=smt.executeQuery("SELECT * FROM
STOCKDATA");
            rs.next();
            gold=rs.getInt("GOLDRATE");
        }
        catch(Exception e){
             e.printStackTrace();


        }
        //TODO write your implementation code here:
        return gold;
    }
}
```



NewWebService Web Service Tester

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

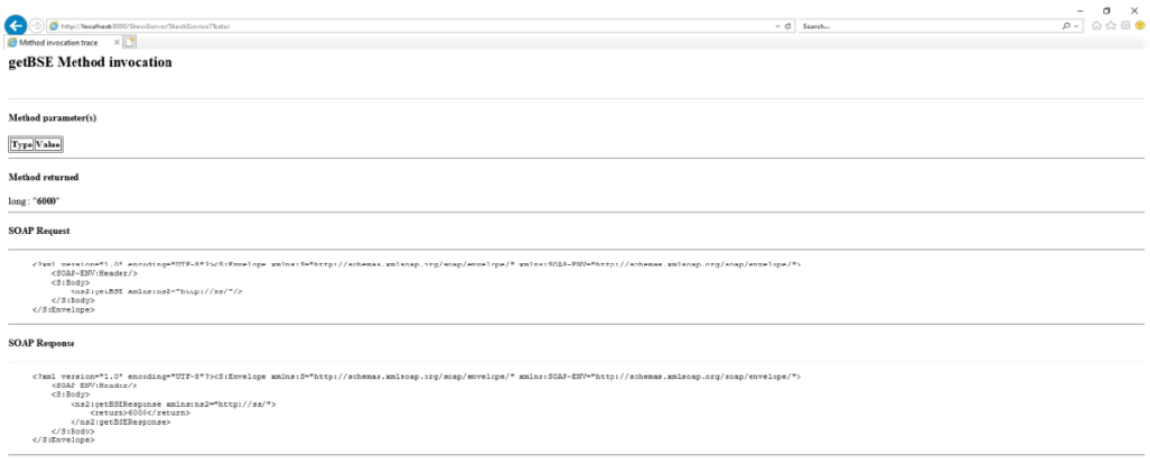
To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract long stock.NewWebService.getGOLD()
getGOLD ()

public abstract long stock.NewWebService.getNSE()
getNSE ()

public abstract long stock.NewWebService.getBSE()
getBSE ()

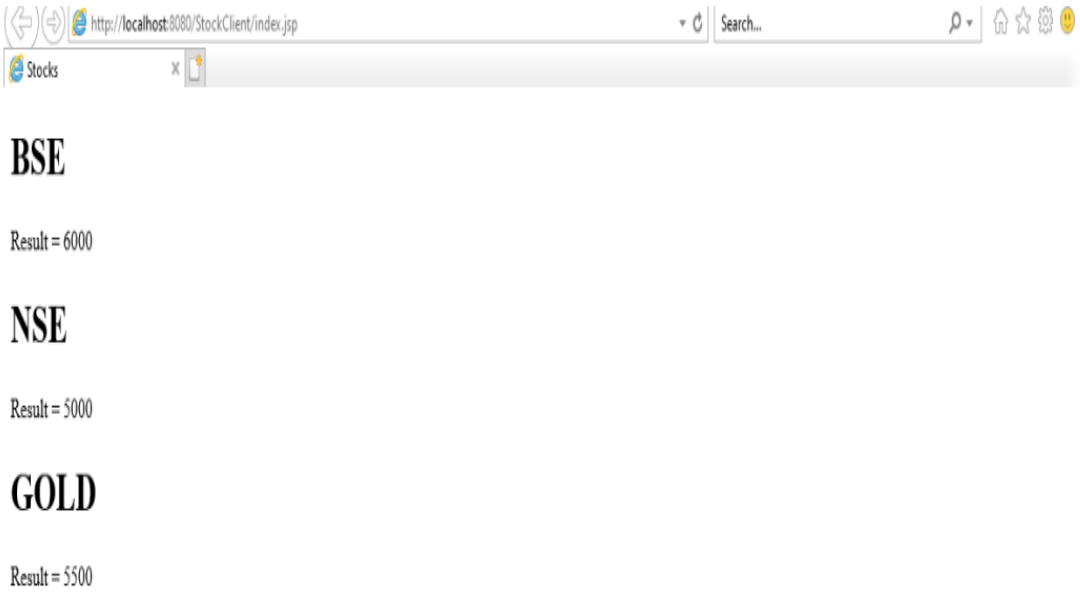


JSP CODE

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

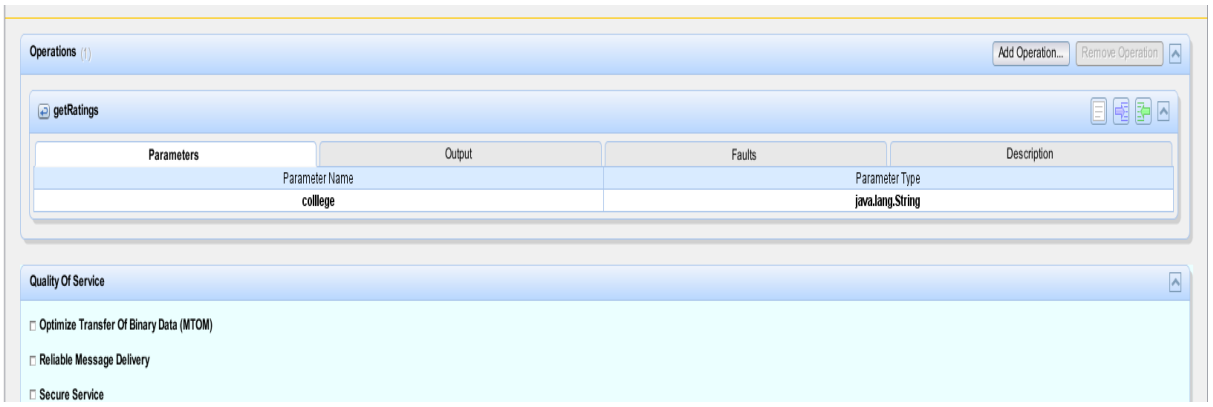
```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
            <%-- start web service invocation --%><hr/> <h1>NSE</h1>
    <%
    try {
            stock.NewWebService_Service service = new stock.NewWebService_Service();
            stock.NewWebService port = service.getNewWebServicePort();
            // TODO process result here
            long result = port.getBSE();
            out.println("Result = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
    %>    <%-- start web service invocation --%><hr/> <h1>BSE</h1>
    <%
    try {
            stock.NewWebService_Service service = new stock.NewWebService_Service();
            stock.NewWebService port = service.getNewWebServicePort();
            // TODO process result here
            long result = port.getGOLD();
            out.println("Result = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>

 <h1>GOLDRATE</h1>
    <%
    try {
```

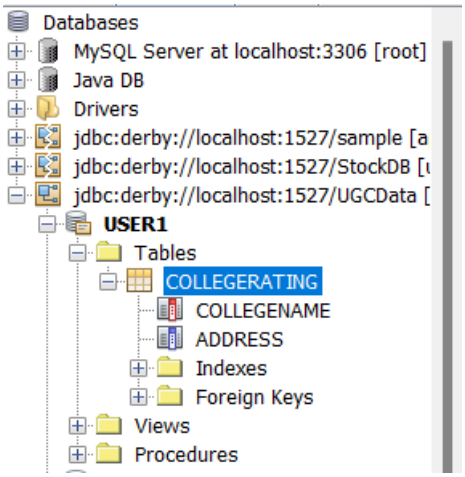
```
        stock.NewWebService_Service service = new stock.NewWebService_Service();
        stock.NewWebService port = service.getNewWebServicePort();
        // TODO process result here
        long result = port.getNSE();
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>

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

    </body>
</html>
```

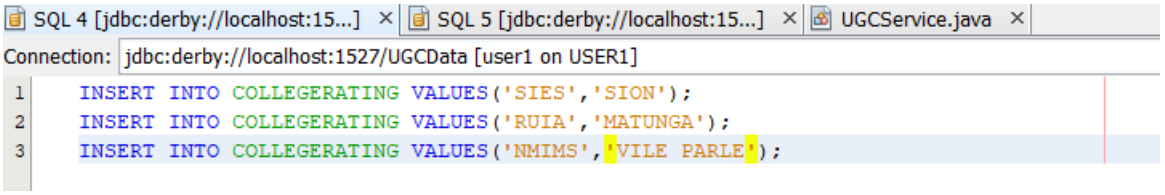


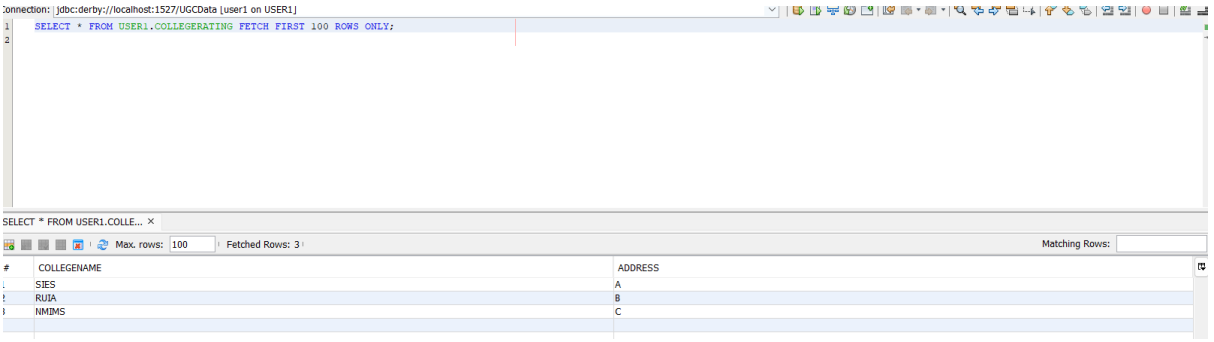
4.Create a web service for UGC that contains a method which accepts college name as parameter and returns the NAAC rating. The college names and their ratings are stored in database. Design a web client to test the above web service.

```
 */
package data;

import java.sql.*;

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

/**
 *
 * @author Dinesh
 */
@WebService(serviceName = "UGCService")
public class UGCService {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "getRatings")
    public String getRatings(@WebParam(name = "colllege") String colllege) {
        //TODO write your implementation code here:
        String ratings="";
        try{
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/UGCData","user1","user@123");
            PreparedStatement pstmt=con.prepareStatement("select * from COLLEGERATING WHERE COLLEGENAME=?");
            pstmt.setString(1, colllege);
            ResultSet rs=pstmt.executeQuery();
            rs.next();
            ratings=rs.getString("ADDRESS");



        }
        catch(Exception e){
            e.printStackTrace();
        }
        return ratings;
    }
}
```









UGCService 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 java.lang.String data.UGCService.getRatings(java.lang.String)

getRatings (SIES)

---

getRatings Method invocation

Method parameter(s)

Type	Value
java.lang.String	SIES

Method returned

java.lang.String : "**A**"

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:getRatings xmlns:ns2="http://data/">
            <colllege>SIES</colllege>
        </ns2:getRatings>
    </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:getRatingsResponse xmlns:ns2="http://data/">
            <return>A</return>
        </ns2:getRatingsResponse>
```

```
<%--
    Document   : index
    Created on : 12 Sep, 2023, 8:20:42 PM
    Author     : Dinesh
--%>

<%@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>
        <form action="index.jsp" method="POST">
        Enter College Name <input type="text" name="value1">
        <input type="submit" name="btnsubmit" value="Submit"/>
        </form>
    <%
    try {
        data.UGCService_Service service = new data.UGCService_Service();
        data.UGCService port = service.getUGCServicePort();
         // TODO initialize WS operation arguments here
        java.lang.String colllege = request.getParameter("value1");
        // TODO process result here
        java.lang.String result = port.getRatings(colllege);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>

    </body>
</html>
```

Enter College Name SIES Submit

Result = A

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.

```java
package mypack;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import java.sql.*;
/**
 *
 * @author Dinesh
 */
@WebService(serviceName = "NewWebService")
public class NewWebService {

    /**
     * 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/Practical5","user1", "user1@123");
    PreparedStatement pstmt = con.prepareStatement("SELECT * FROM BREAKINGNEWS WHERE DATE=?");
    pstmt.setString(1, date);
    ResultSet rs = pstmt.executeQuery();
    rs.next();
    news = rs.getString("NEWS");
    } catch (Exception e) {
    e.printStackTrace();
    }
    return news;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "getPrediction")
    public String getPrediction(@WebParam(name = "Sunsignname") String Sunsignname) {
        String prediction = "";
try {
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Practical5","user1", "user1@123");
    PreparedStatement pstmt = con.prepareStatement("SELECT * FROM FUTURE WHERE SUNSIGN=?");
    pstmt.setString(1, Sunsignname);
    ResultSet rs = pstmt.executeQuery();
    rs.next();
    prediction = rs.getString("PREDICTIONS");
    } catch (Exception e) {
    e.printStackTrace();
    }
    return prediction;
    }
}
```

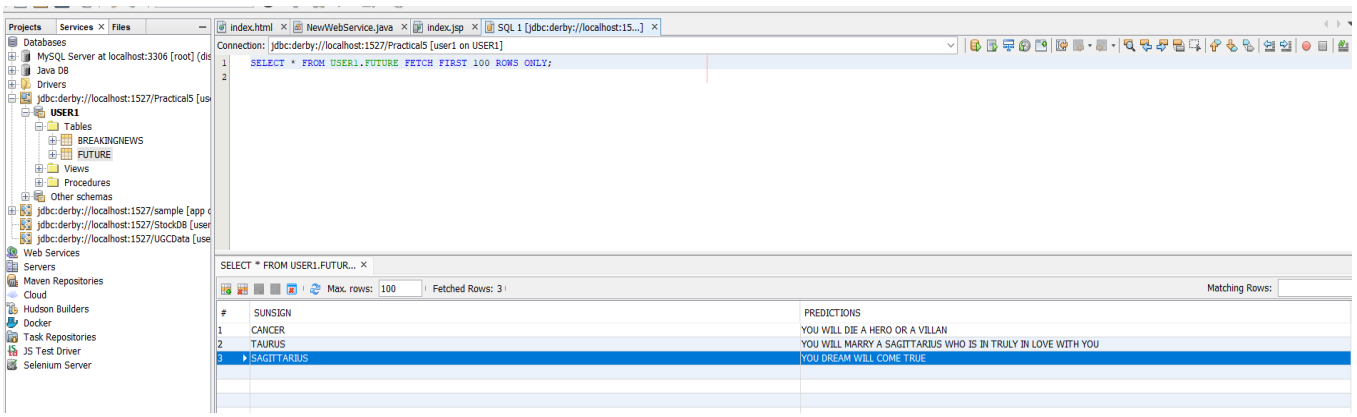
Operations (2) — Add Operation... | Remove Operation

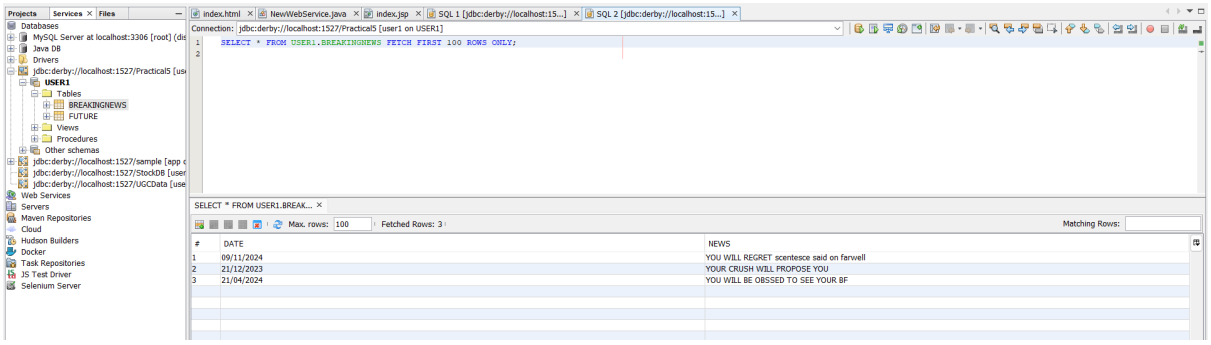
getBreakingNews

Parameters	Output	Faults	Description
Parameter Name		Parameter Type	
date		java.lang.String	

getPrediction

Parameters	Output	Faults	Description
Parameter Name		Parameter Type	
Sunsignname		java.lang.String	


index.html
NewWebService.java
index.jsp
SQL 1 [jdbc:derby://localhost:15...]
Connection: jdbc:derby://localhost:1527/Practical5 [user1 on USER1]
SELECT * FROM USER1.FUTURE FETCH FIRST 100 ROWS ONLY;
SELECT * FROM USER1.FUTUR...
Max. rows: 100
Fetched Rows: 3
Matching Rows:
SUNSIGN PREDICTIONS
1 CANCER YOU WILL DIE A HERO OR A VILLAN
2 TAURUS YOU WILL MARRY A SAGITTARIUS WHO IS IN TRULY IN LOVE WITH YOU
3 SAGITTARIUS YOU DREAM WILL COME TRUE


index.html
NewWebService.java
index.jsp
SQL 1 [jdbc:derby://localhost:15...]
SQL 2 [jdbc:derby://localhost:15...]
Connection: jdbc:derby://localhost:1527/Practical5 [user1 on USER1]
SELECT * FROM USER1.BREAKINGNEWS FETCH FIRST 100 ROWS ONLY;
SELECT * FROM USER1.BREAK...
Max. rows: 100
Fetched Rows: 3
Matching Rows:
DATE NEWS
1 09/11/2024 YOU WILL REGRET scentesce said on farwell
2 21/12/2023 YOUR CRUSH WILL PROPOSE YOU
3 21/04/2024 YOU WILL BE OBSSED TO SEE YOUR BF

```java
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>NEWS & PREDICTION</h1>
        <form action="index.jsp" method="POST">
        Enter Date <input type="text" name="Value1"><br>
        Enter Sun Sign <input type="text" name="Value2"><br>
        <input type="submit" name="btnsubmit" value="Submit"/>
        </form>
        <hr/>
            <%-- start web service invocation --%><hr/>
    <%
    try {
        mypack.NewWebService_Service service = new mypack.NewWebService_Service();
        mypack.NewWebService port = service.getNewWebServicePort();
         // TODO initialize WS operation arguments here
        java.lang.String date = request.getParameter("Value1");

        java.lang.String result = port.getBreakingNews(date);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
    <%-- start web service invocation --%><hr/>
    <%
    try {
        mypack.NewWebService_Service service = new mypack.NewWebService_Service();
        mypack.NewWebService port = service.getNewWebServicePort();
         // TODO initialize WS operation arguments here
        java.lang.String sunsignname = request.getParameter("Value2");
        // TODO process result here
        java.lang.String result = port.getPrediction(sunsignname);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>

    </body>
</html>
```

NewWebService 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 java.lang.String mypack.NewWebService.getBreakingNews(java.lang.String)
getBreakingNews (21/12/2023)

---

public abstract java.lang.String mypack.NewWebService.getPrediction(java.lang.String)
getPrediction (TAURUS)

---

getBreakingNews Method invocation

Method parameter(s)

Type	Value
java.lang.String	21/12/2023

Method returned

java.lang.String : "**YOUR CRUSH WILL PROPOSE YOU**"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-E
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:getBreakingNews xmlns:ns2="http://mypack/">
            <date>21/12/2023</date>
        </ns2:getBreakingNews>
    </S:Body>
```


getPrediction Method invocation

Method parameter(s)

Type	Value
java.lang.String	TAURUS

Method returned

java.lang.String : "**YOU WILL MARRY A SAGITTARIUS WHO IS IN TRULY IN LOVE WITH YOU**"

SOAP Request

NEWS & PREDICTION

Enter Date

Enter Sun Sign

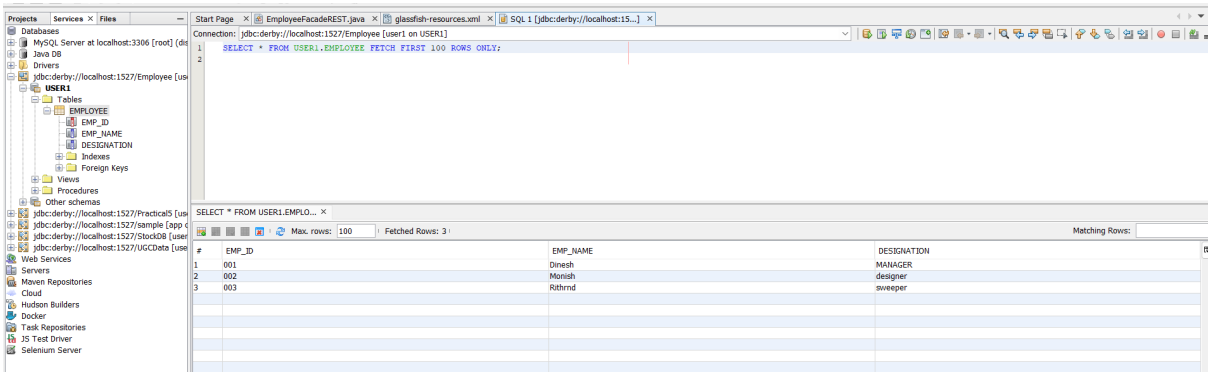
Submit

Result = YOUR CRUSH WILL PROPOSE YOU

Result = YOU DREAM WILL COME TRUE

---

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



CODE:

```
package mypack.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 mypack.Employee;

/**
 *
 * @author Dinesh
 */
@Stateless
@Path("mypack.employee")
public class EmployeeFacadeREST extends
AbstractFacade<Employee> {

    @PersistenceContext(unitName = "Prac6PU")
    private EntityManager em;

    public EmployeeFacadeREST() {
        super(Employee.class);
    }

    @POST
    @Override
    @Consumes({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public void create(Employee entity) {
        super.create(entity);
    }

    @PUT
    @Path("{id}")
    @Consumes({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public void edit(@PathParam("id") String id, Employee entity) {
```

```
        super.edit(entity);
    }

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

    @GET
    @Path("{id}")
    @Produces({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public Employee find(@PathParam("id") String id) {
        return super.find(id);
    }

    @GET
    @Override
    @Produces({ MediaType.APPLICATION_JSON})
    public List<Employee> findAll() {
        return super.findAll();
    }

    @GET
    @Path("{from}/{to}")
    @Produces({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public List<Employee> findRange(@PathParam("from") Integer from,
@PathParam("to") Integer to) {
        return super.findRange(new int[]{from, to});
```

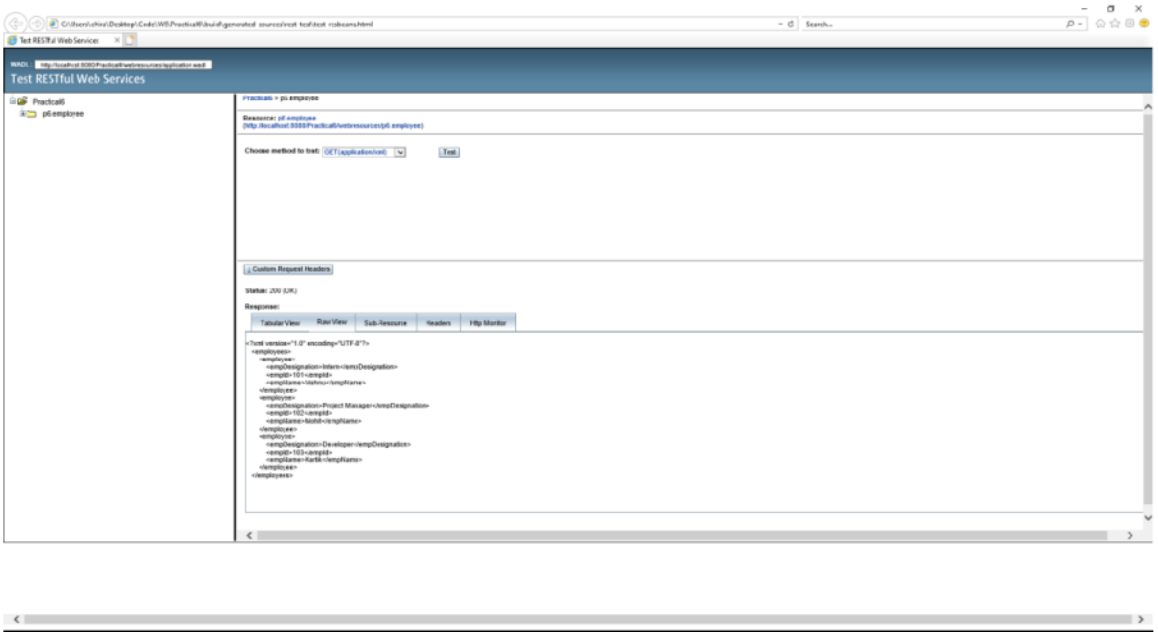
```
    }

    @GET
    @Path("count")
    @Produces(MediaType.TEXT_PLAIN)
    public String countREST() {
        return String.valueOf(super.count());
    }

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

}
```

.html code

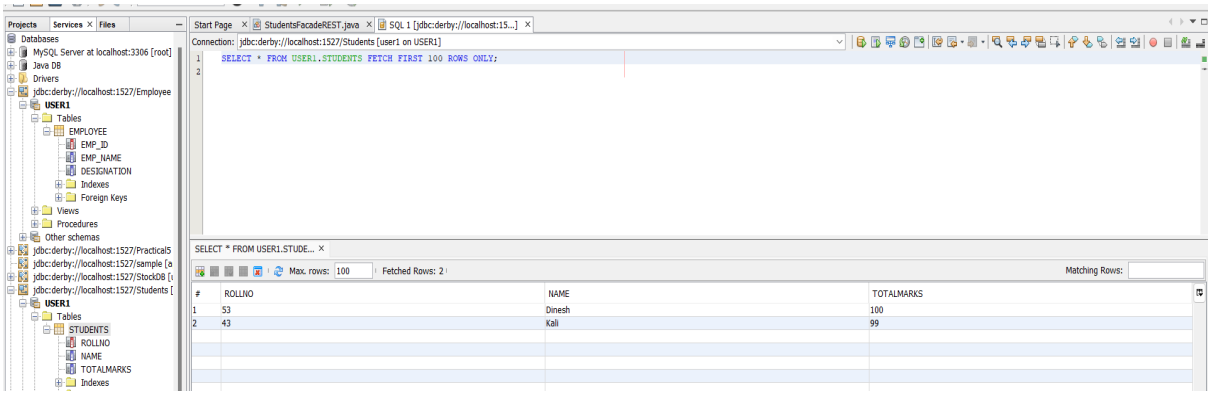
```
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form>
            <input type="submit" formaction="index.jsp" value="Get Employees">
        </form>

    </body>
</html>
```

.jsp Code

```
<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;
            }
            td,th{
                border: 1px solid blue;
                text-align: center;
                padding: 8px;
            }
        </style>
        <script>
            var request=new XMLHttpRequest();
            request.open('GET','http://localhost:8080/Prac6/webresources/mypack.employee/',true)
            request.onload=function()
            {
                var data=JSON.parse(this.response);
                for (var i=0;i<data.length;i++){
                    var table=document.getElementById("EMPLOYEE");
                    var row=table.insertRow();
                    var cell1=row.insertCell(0);
                    var cell2=row.insertCell(1);
                    var cell3=row.insertCell(2);
                    cell1.innerHTML=data[i].empId;
                    cell2.innerHTML=data[i].empName;
                    cell3.innerHTML=data[i].desgination;
                }
            };
            request.send();
        </script>
    </head>
    <body>
        <table id="EMPLOYEE">
            <tr>
                <th>Emp ID</th>
                <th>Emp Name</th>
                <th>Designation</th>
            </tr>
    </body>
</html>
```



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.

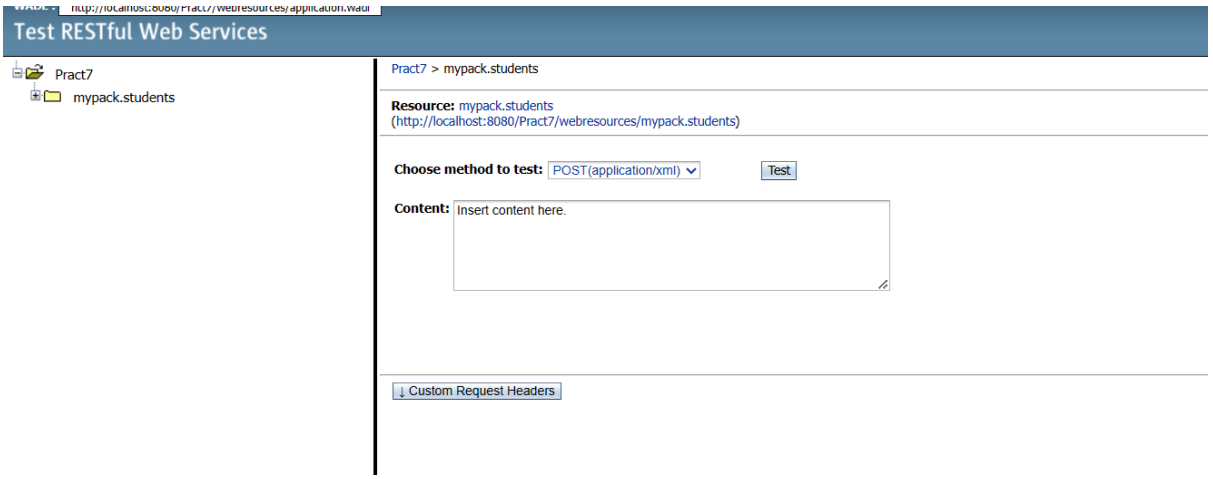



```
package mypack.service;

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 mypack.Students;

/**
 *
 * @author Dinesh
 */
@Stateless
@Path("mypack.students")
public class StudentsFacadeREST extends AbstractFacade<Students> {

    @PersistenceContext(unitName = "Pract7PU")
    private EntityManager em;

    public StudentsFacadeREST() {
        super(Students.class);
    }

    @POST
    @Override
    @Consumes({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public void create(Students entity) {
```

```
        super.create(entity);
    }

    @PUT
    @Path("{id}")
    @Consumes({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public void edit(@PathParam("id") String id, Students entity) {
        super.edit(entity);
    }

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

    @GET
    @Path("{id}")
    @Produces({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public Students find(@PathParam("id") String id) {
        return super.find(id);
    }

    @GET
    @Override
    @Produces({ MediaType.APPLICATION_JSON})
    public List<Students> findAll() {
        return super.findAll();
    }
```

```
    @GET
    @Path("{from}/{to}")
    @Produces({MediaType.APPLICATION_XML,
MediaType.APPLICATION_JSON})
    public List<Students> 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;
    }

}
```




```
public class StudentsFacadeREST extends AbstractFacade<Students> {
```

```
--%>

<%@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;
            }
            td,th{
                border: 1px solid blue;
                text-align: center;
                padding: 8px;
            }
        </style>
        <script>
            var request=new XMLHttpRequest();
            request.open('GET','http://localhost:8080/Pract7/webresources/mypack.students/',true)
            request.onload=function()
            {
                var data=JSON.parse(this.response);
                for (var i=0;i<data.length;i++){
                    var table=document.getElementById("Students");


                    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="Students">
            <tr>
                <th>Roll no</th>
                <th>Student Name</th>
                <th>TotalMarks</th>
            </tr>
    </body>
```

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

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

IService.cs code

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

namespace Prac8
{
    // 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 Product(double a, double b);
```

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

        [OperationContract]
        double Quotient(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 Prac8
{
    // 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 double Difference(double a, double b)
        {
            double result = a - b;
            return result;

        }

        public double Product(double a, double b)
        {
            double result = a * b;
            return result;

        }

        public double Quotient(double a, double b)
        {
            double result = a / b;
            return result;

        }

        public double Sum(double a, double b)
        {
            double result = a + b;
            return result;

        }
    }
}
```

Webform code

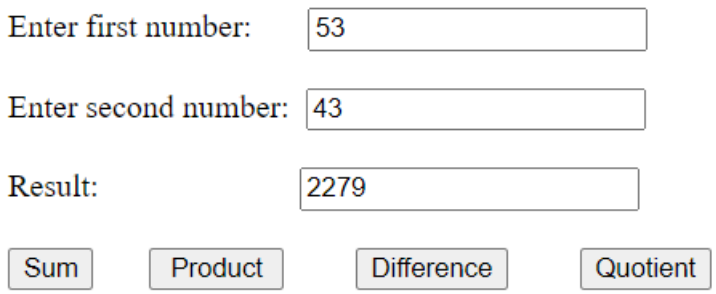
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Prac8.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
             Enter first number:      
            <asp:TextBox ID="fno" runat="server"></asp:TextBox>
            <br />
            <br />
            Enter second number: 
            <asp:TextBox ID="sno" runat="server" ></asp:TextBox>
            <br />
            <br />

Result:             &
nbsp;          
            <asp:TextBox ID="res" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Sum"
/>
     
            <asp:Button ID="Button2" runat="server" OnClick="Button2_Click"
Text="Product" Width="67px" />
       
            <asp:Button ID="Button3" runat="server" OnClick="Button3_Click"
Text="Difference" />
       
            <asp:Button ID="Button4" runat="server" OnClick="Button4_Click"
Text="Quotient" />
            <br />
            <br />

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

Webform.aspx.cs

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

namespace Prac8
{
    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 num1 = double.Parse(fno.Text);
            double num2 = double.Parse(sno.Text);
            res.Text = Convert.ToString(client.Sum(num1, num2));
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            ServiceReference1.Service1Client client = new
ServiceReference1.Service1Client();
            double num1 = double.Parse(fno.Text);
            double num2 = double.Parse(sno.Text);
            res.Text = Convert.ToString(client.Product(num1, num2));
        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            ServiceReference1.Service1Client client = new
ServiceReference1.Service1Client();
            double num1 = double.Parse(fno.Text);
            double num2 = double.Parse(sno.Text);
            res.Text = Convert.ToString(client.Difference(num1, num2));
        }

        protected void Button4_Click(object sender, EventArgs e)
        {
            ServiceReference1.Service1Client client = new
ServiceReference1.Service1Client();
            double num1 = double.Parse(fno.Text);
            double num2 = double.Parse(sno.Text);
            res.Text = Convert.ToString(client.Quotient(num1, num2));
        }
    }

}
```

Output



9. Create a WCF service with different endpoint for Soap based and Rest based implementation

Service1.svc code

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

namespace Pract9
{
    // 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($"Your DREAM WILL COME TRUE {value}!");
        }

    }
}
```

IService1.cs code

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

namespace Pract9
{
    // 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)] // if it is going to
be exposed- get method and use sayhello to access by restful ws, content type we're
sending,and what accept value(req,res format)
        string SayHello(string value);

    }
}
```

Webconfig code

```
<?xml version="1.0"?>
<configuration>

  <appSettings>
    <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2"/>
  </system.web>
  <system.serviceModel>
        <services>
               <service name="Pract9.Service1">
                      <endpoint address="jsonservice" binding="webHttpBinding"
contract="Pract9.IService1" behaviorConfiguration="web"></endpoint>
                      <endpoint address="soapservice" binding="basicHttpBinding"
contract="Pract9.IService1"></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