

**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 | Practical Name | SIGN |  |
|  |  |
| NO |  |  |  |
| 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();

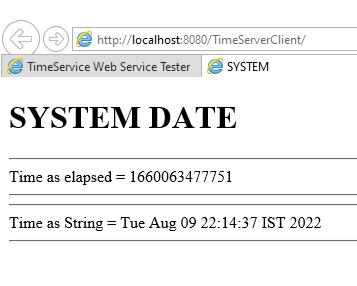
echo "Elapsed time is",$t1->return; /\* here the return is not a function but the name and can be change by using @webresult before the function \*/

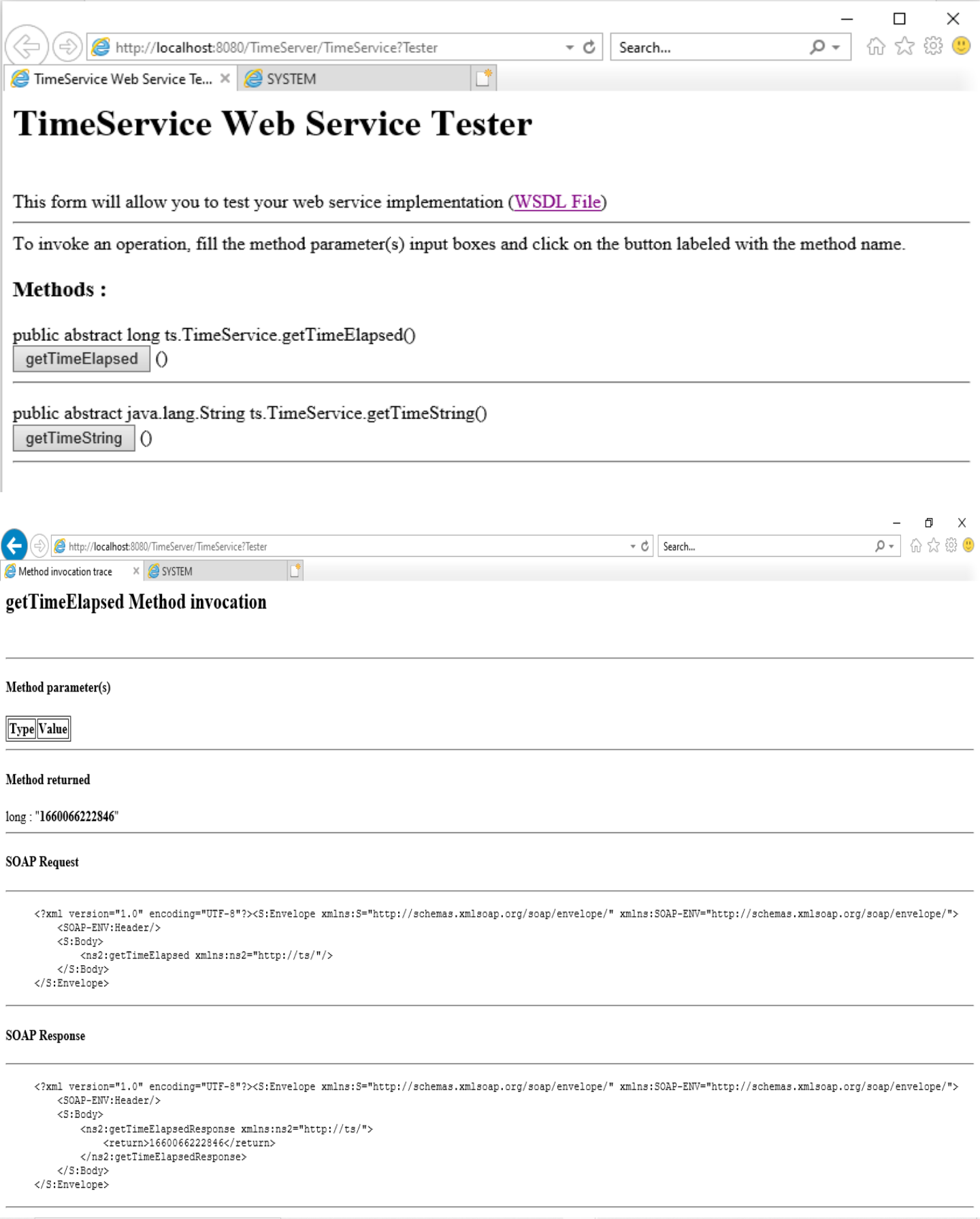
$t2=$client->getTimeAsString();

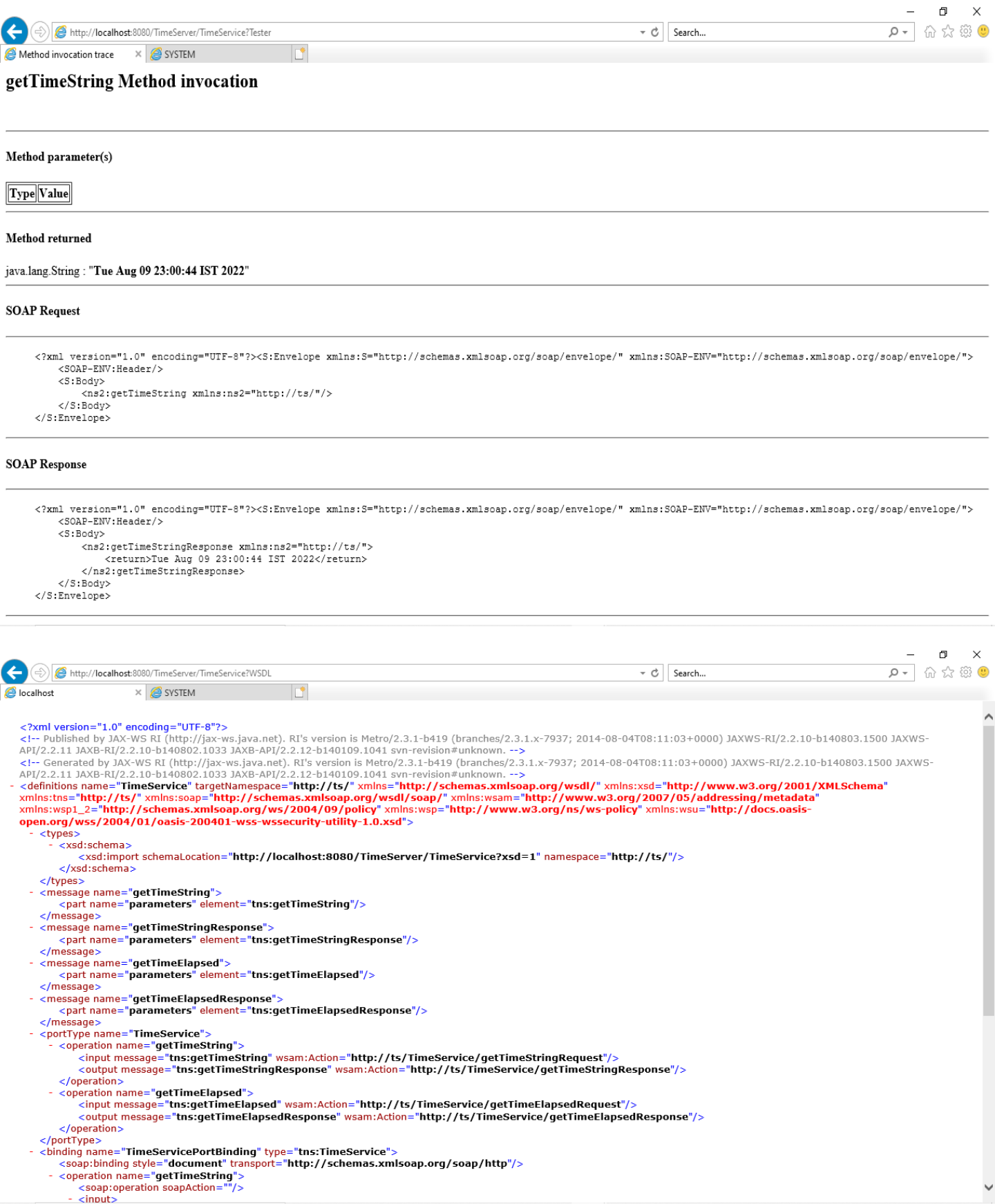
echo "<br>Time as String is",$t2->return;

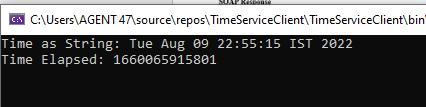
?>

OUTPUT:









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

}

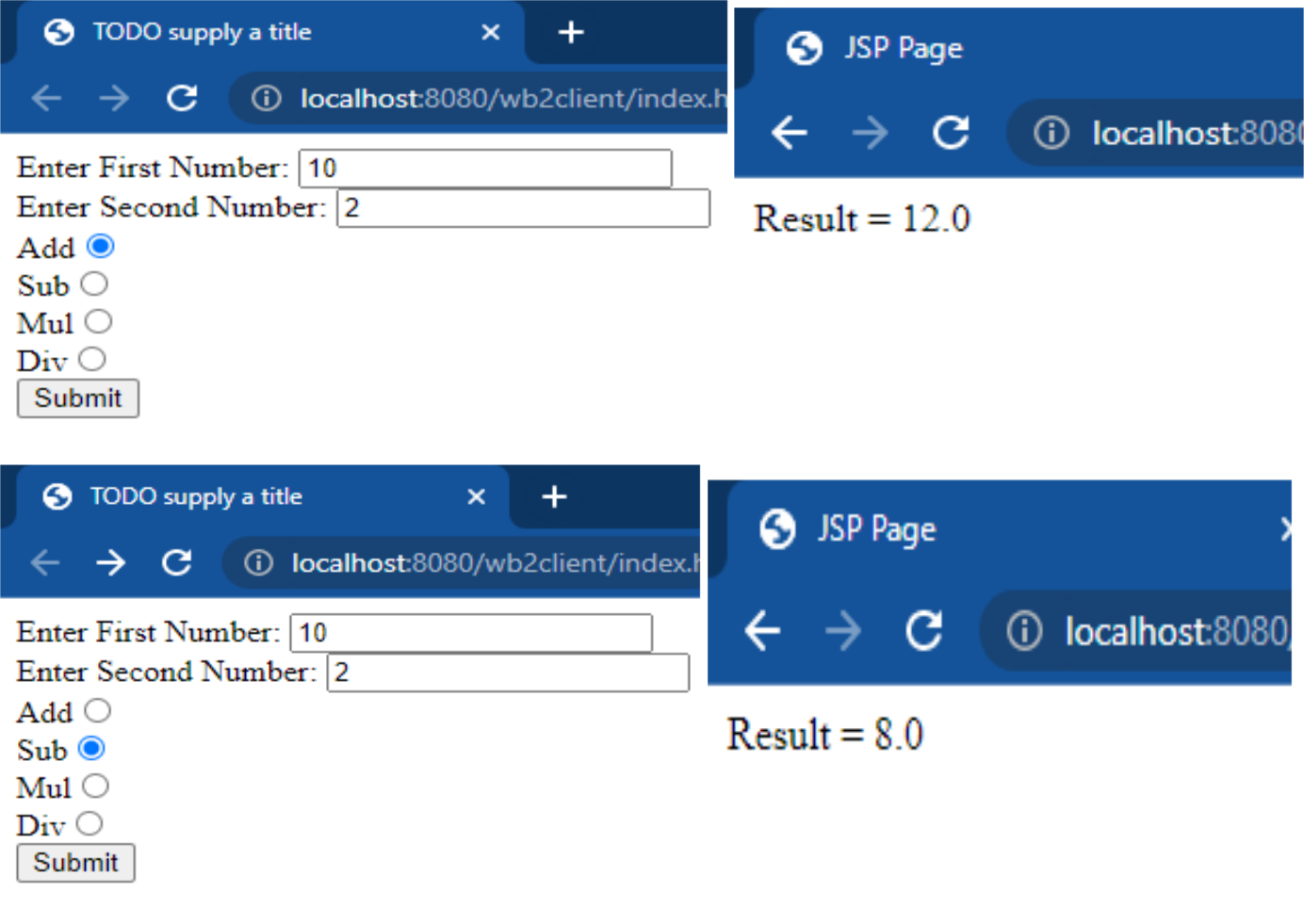
}

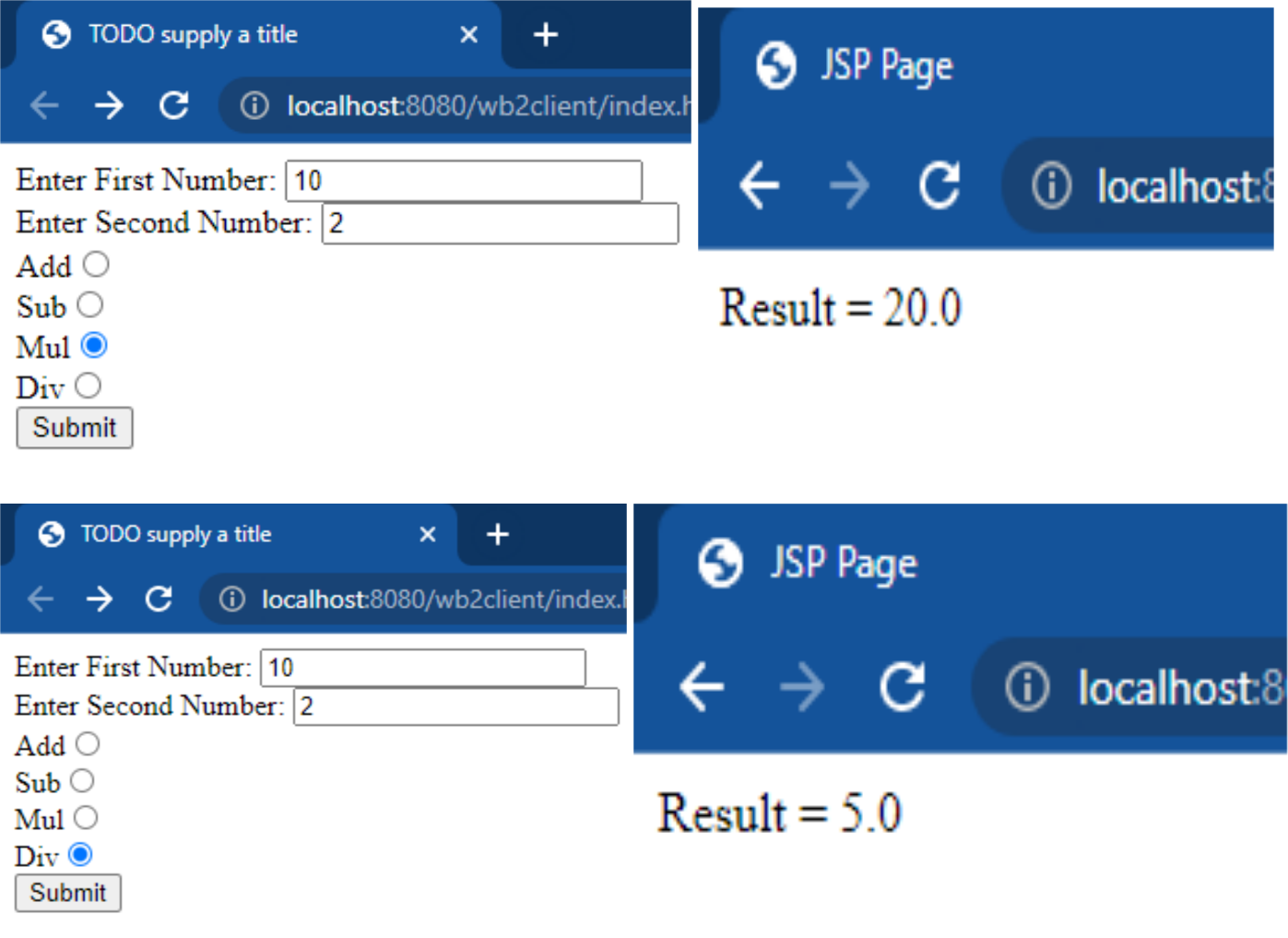
%>

</body>

</html>

OUTPUT:





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

}

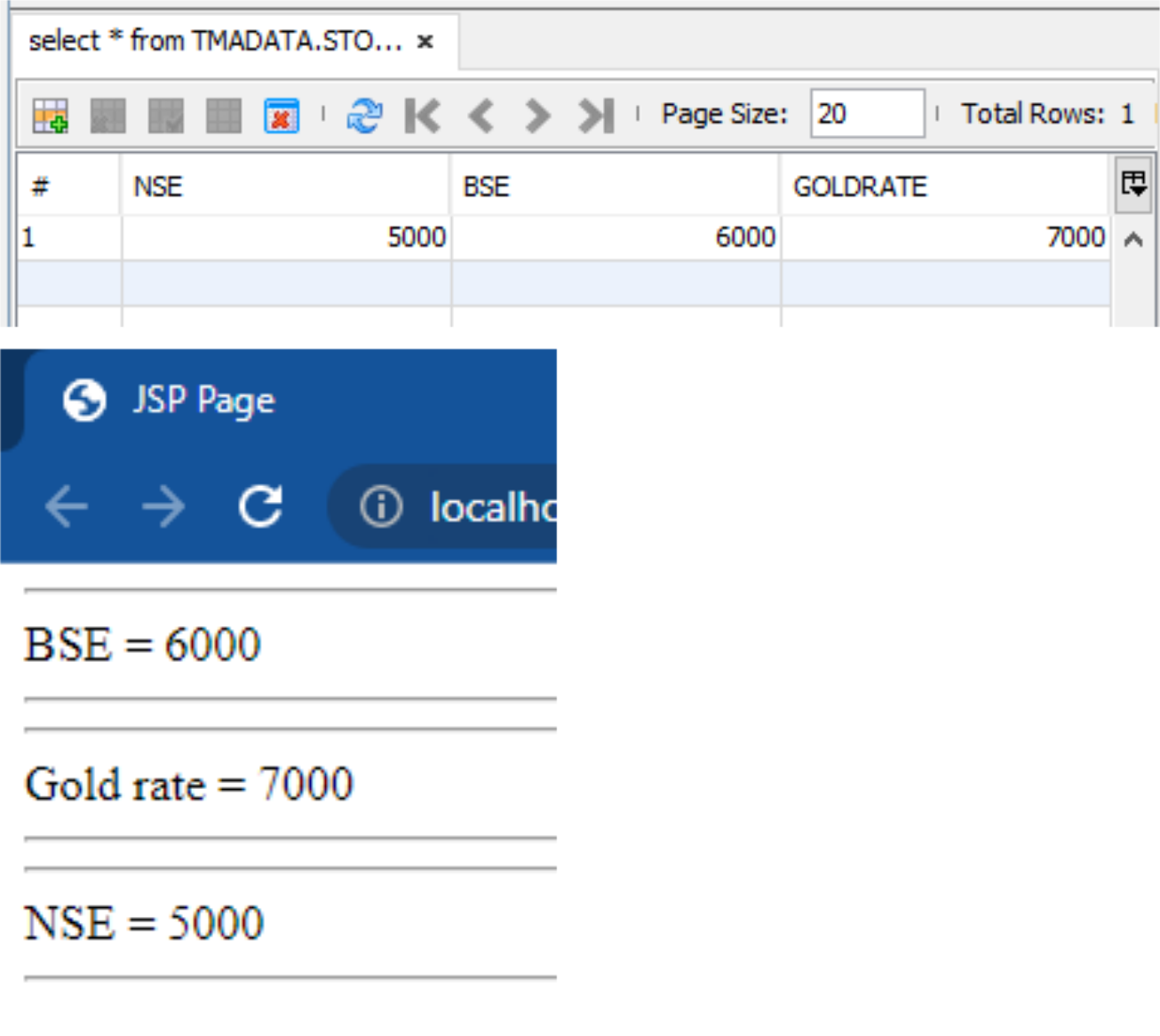
%>

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

</body>

</html>

OUTPUT:



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">

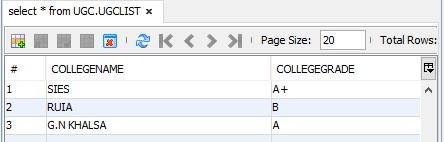
Enter College Name: <input type="text" name="value1"> <input type="submit" >

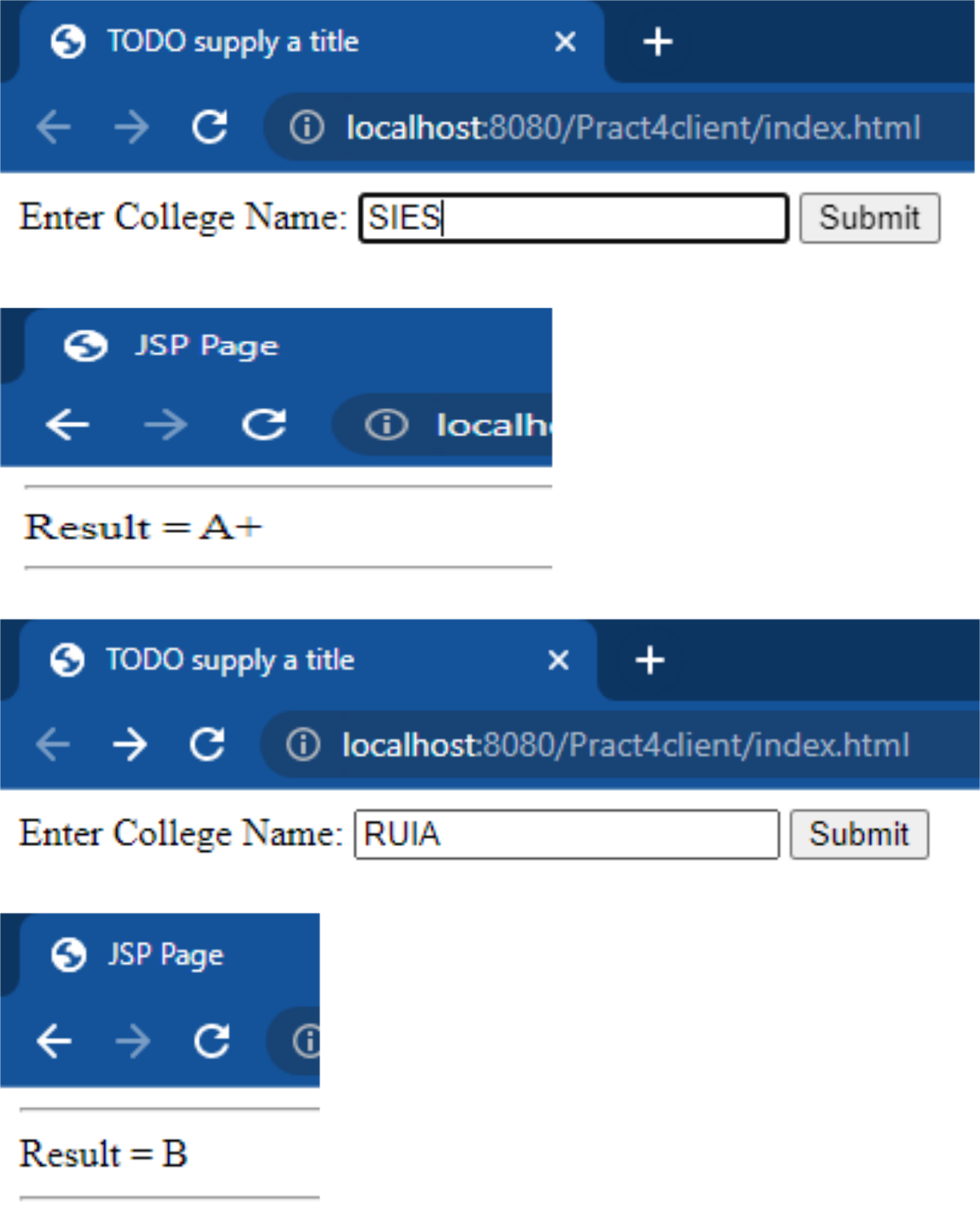
</form>

</body>

</html>

OUTPUT:





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=UTF8"> <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

}

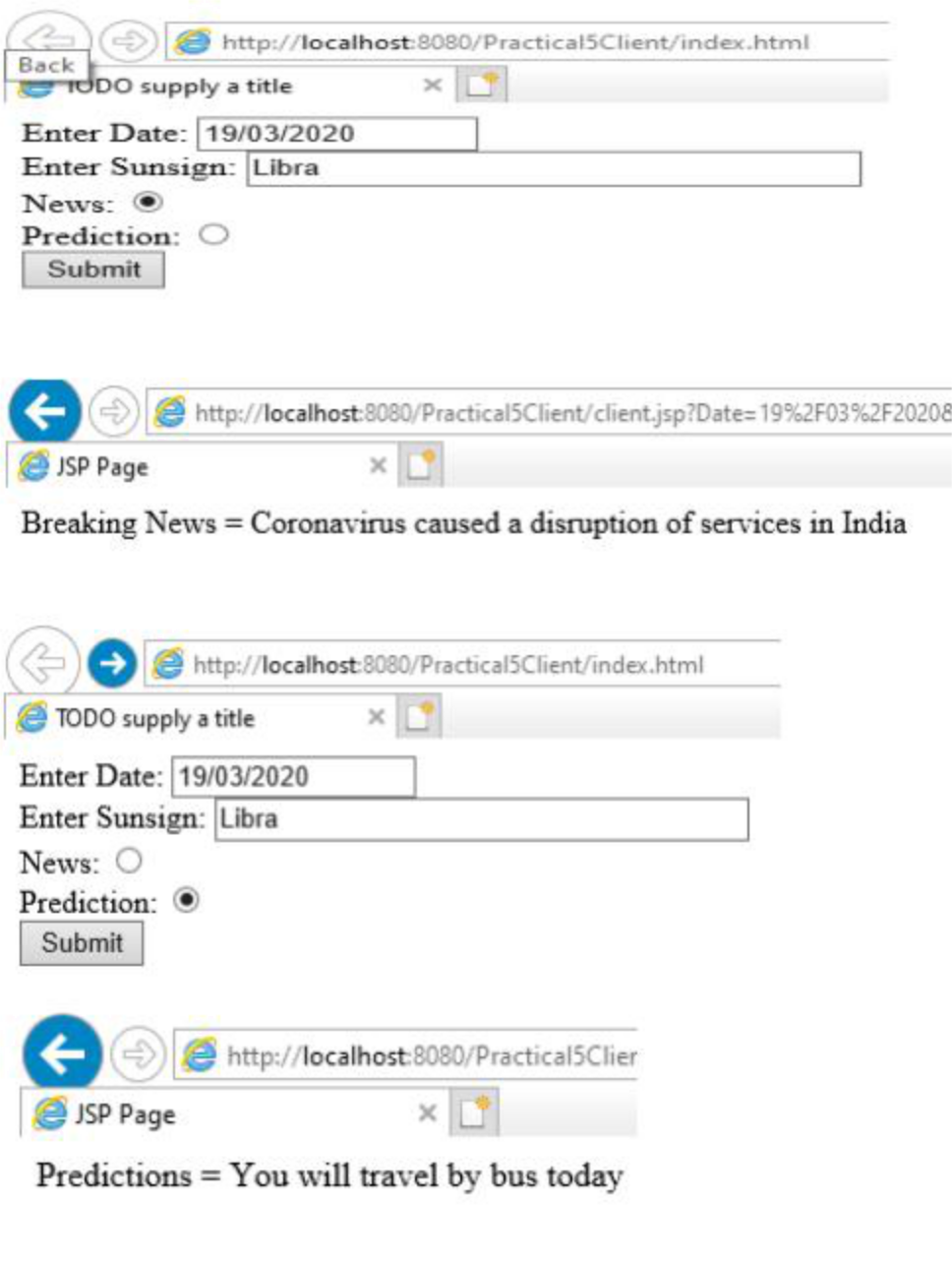
}

%>

</body>

</html>

OUTPUT:



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

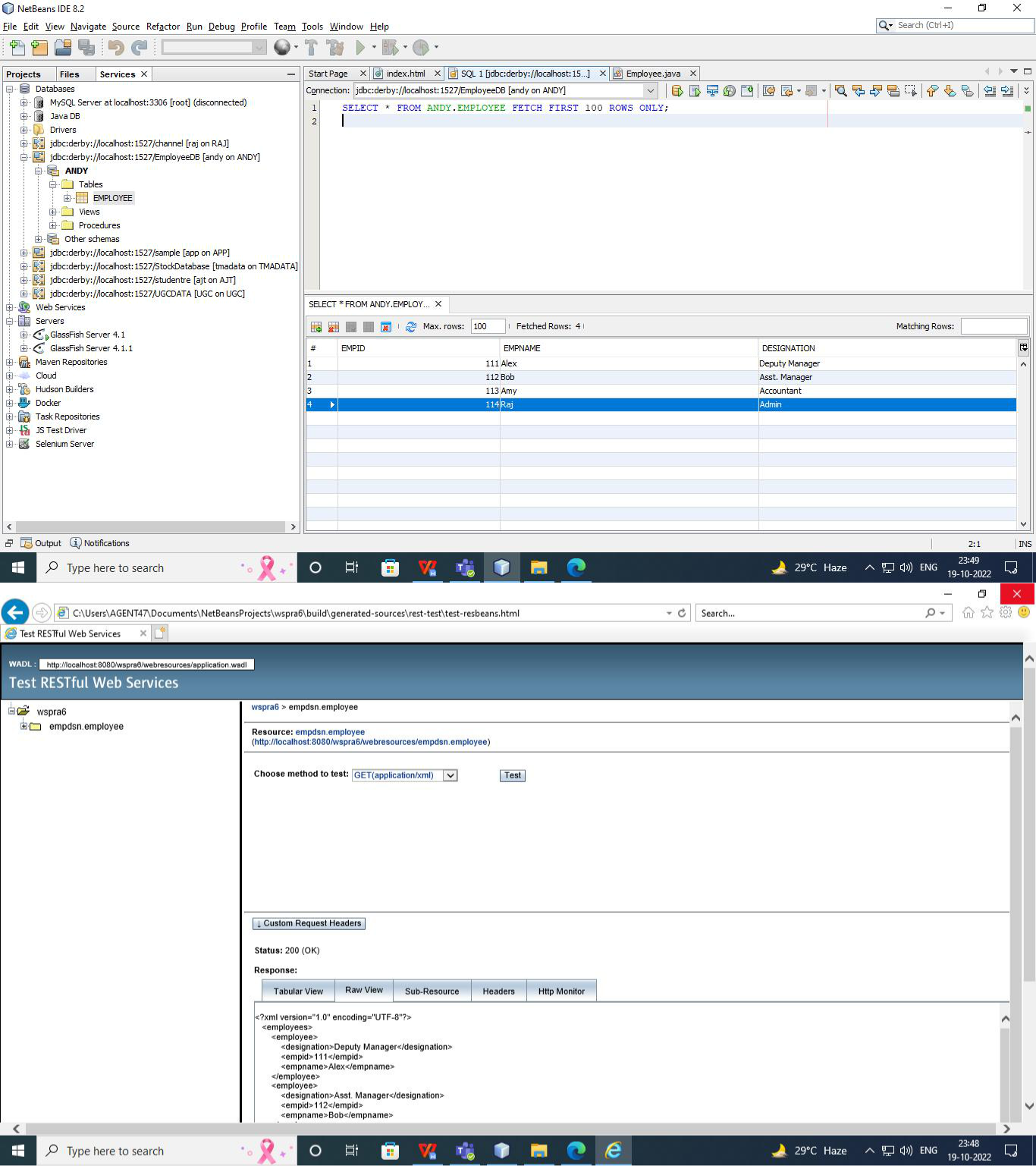
public String toString() {

return "empdsn.Employee[ empid=" + empid + " ]";

}

}

OUTPUT:



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

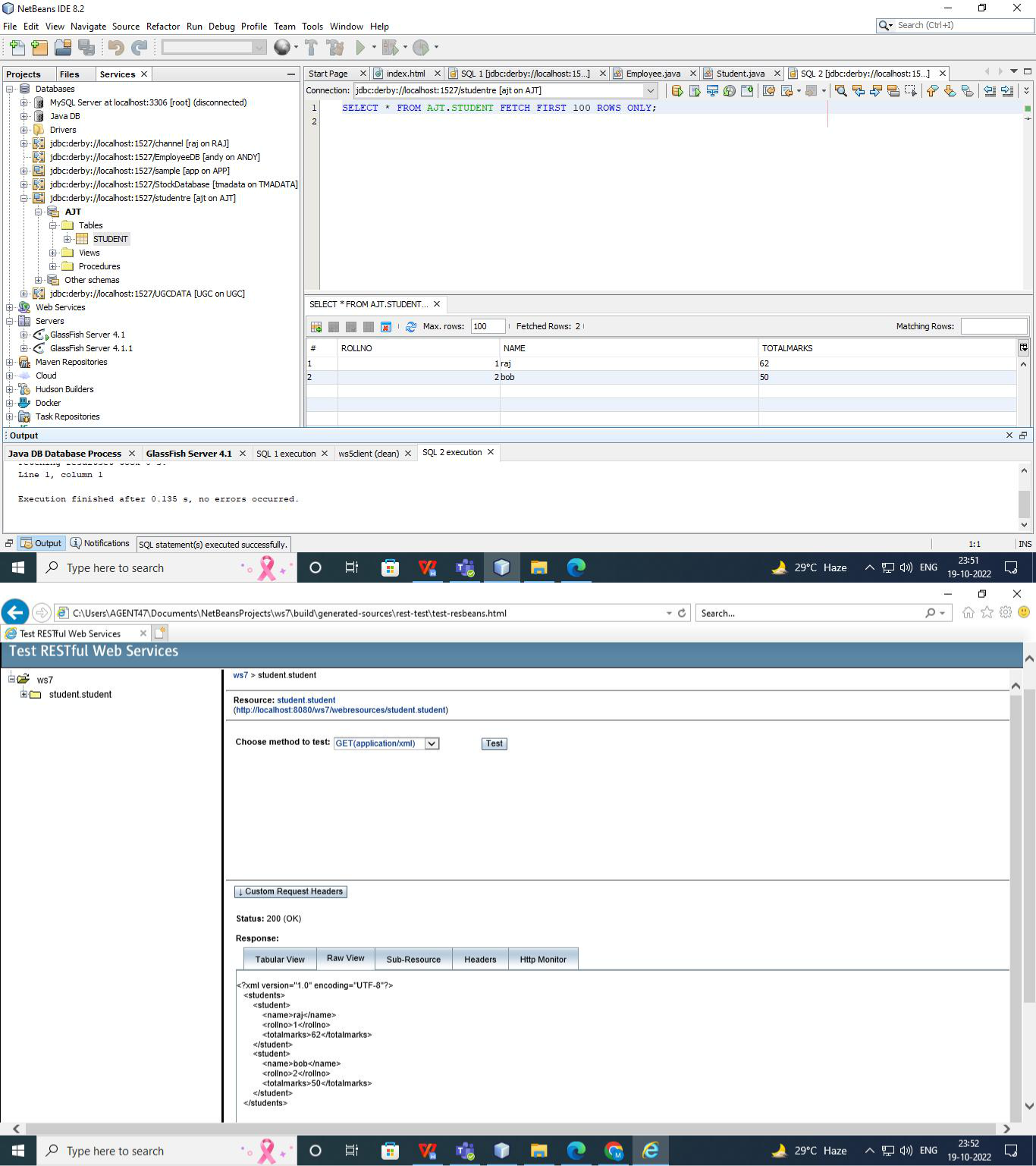
public String toString() {

return "student.Student[ rollno=" + rollno + " ]";

}

}

OUTPUT:



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);

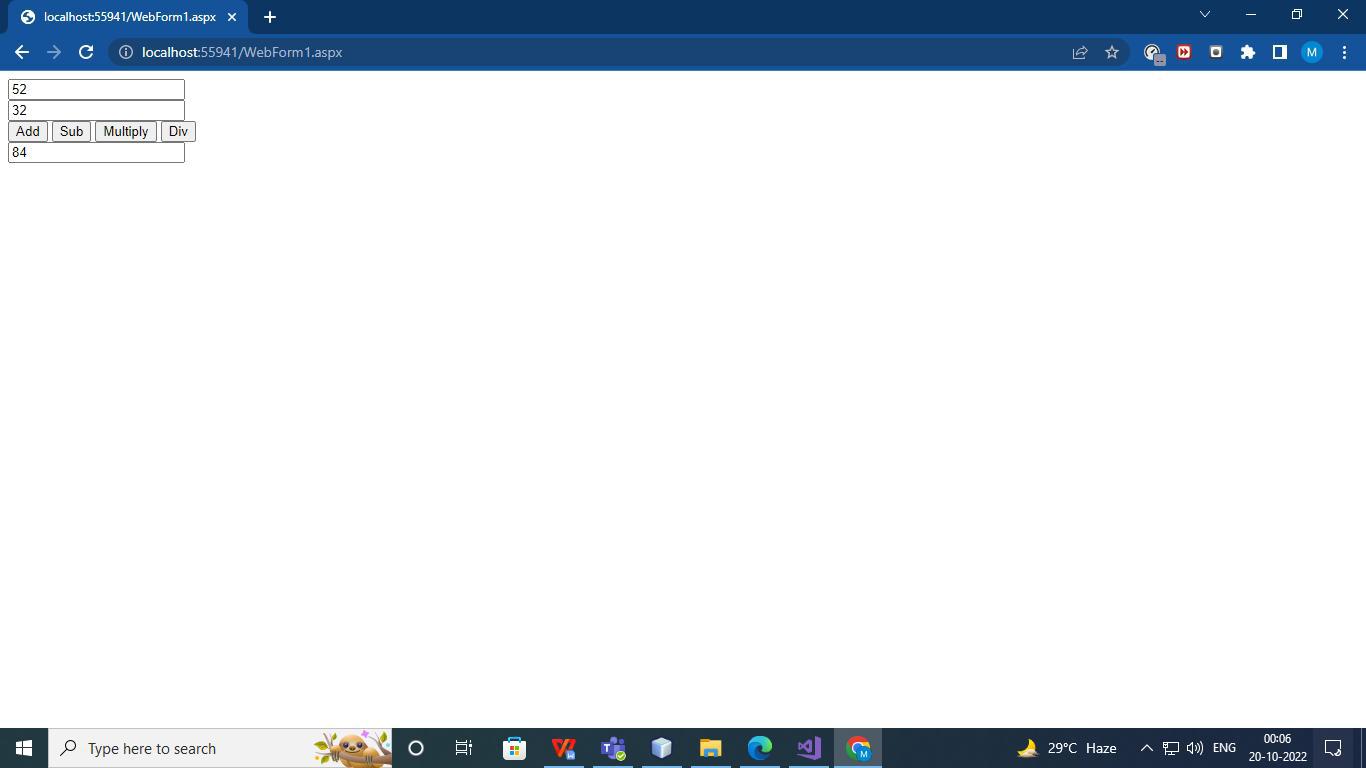
[OperationContract]

double Div(double a, double b);

}

}

OUTPUT:



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"/> <!--

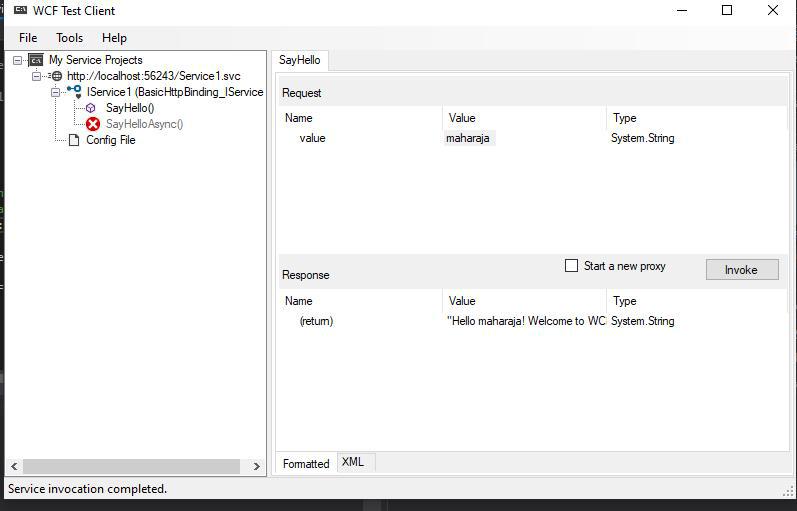
To browse web app root directory during debugging, set the value below to true. Set to false before deployment to avoid disclosing web app folder information. -->

<directoryBrowse enabled="true"/>

</system.webServer>

</configuration>

OUTPUT:



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:

