A. Murugan

2018115059

IDE used: Netbeans IDE

Server used : Glassfish server

Database used : MySQL (XAMPP)

Web Services- SOAP

1. Creating and Testing a web service in a browser

Source Code :

<u>CalculatorWebService.java</u>

```
/*
 * 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 com.calculator;


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


/**
 *
```

```java
 * @author Murugan
 */
@WebService(serviceName = "CalculatorWebService")
public class CalculatorWebService {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "Hello " + txt + " !";
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "addition")
    public String addition(@WebParam(name = "num1") float num1,
@WebParam(name = "num2") float num2) {
        //TODO write your implementation code here:
        float res = num1+num2;
        return (Float.toString(res));
    }

    /**
```

```java
 * Web service operation

 */

@WebMethod(operationName = "subtraction")

public String subtraction(@WebParam(name = "num1") float num1,
@WebParam(name = "num2") float num2) {

    //TODO write your implementation code here:

  float res = num1-num2;

   return (Float.toString(res));

}


/**

 * Web service operation

 */

@WebMethod(operationName = "multiplication")

public String multiplication(@WebParam(name = "num1") float num1,
@WebParam(name = "num2") float num2) {

    //TODO write your implementation code here:

     float res = num1*num2;

   return (Float.toString(res));

}


/**

 * Web service operation

 */

@WebMethod(operationName = "division")
```

```java
    public String division(@WebParam(name = "num1") float num1,
@WebParam(name = "num2") float num2) {

        //TODO write your implementation code here:

        float res = num1/num2;

        return (Float.toString(res));

    }

}
```

Output:

hello Method invocation

Method parameter(s)

Type	Value
java.lang.String	murugan

Method returned

java.lang.String : **"Hello murugan !"**

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:hello xmlns:ns2="http://calculator.com/">
            <name>murugan</name>
        </ns2:hello>
    </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:helloResponse xmlns:ns2="http://calculator.com/">
            <return>Hello murugan !</return>
        </ns2:helloResponse>
    </S:Body>
</S:Envelope>
```

CalculatorWebService 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 com.calculator.CalculatorWebService.hello(java.lang.String)
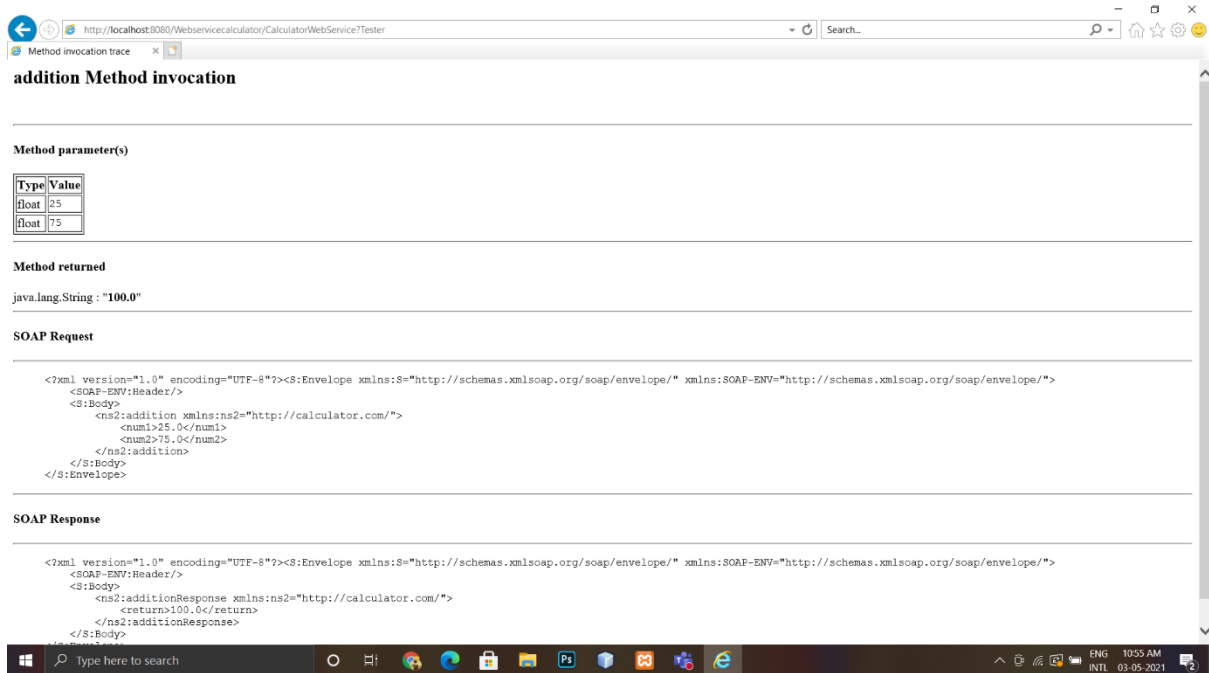[hello] ([murugan])

public abstract java.lang.String com.calculator.CalculatorWebService.multiplication(float,float)
[multiplication] ([____] , [____])

public abstract java.lang.String com.calculator.CalculatorWebService.subtraction(float,float)
[subtraction] ([____] , [____])

public abstract java.lang.String com.calculator.CalculatorWebService.division(float,float)
[division] ([____] , [____])

public abstract java.lang.String com.calculator.CalculatorWebService.addition(float,float)
[addition] ([25] , [75])

2. Creating a client side for the Web service and run it

Source Code :

index.jsp

```
<html>

    <head>

        <title>Arithmetic operations</title>

            <style>
```

input[type=text], select {

  width: 20%;

  padding: 12px 20px;

  margin: 8px 0;

  display: inline-block;

```css
    border: 1px solid #ccc;

    border-radius: 4px;

    box-sizing: border-box;

  }


input[type=submit] {

    width: 30%;

    background-color: #4CAF50;

    color: white;

    padding: 14px 20px;

    margin: 8px 0;

    border: none;

    border-radius: 4px;

    cursor: pointer;

  }


input[type=submit]:hover {

    background-color: #45a049;

  }


div {

    border-radius: 5px;

    background-color: #f2f2f2;

    padding: 20px;
```

```
        }
</style>
    </head>
    <body>
        <div>
    <center>
        <h1> Arithmetic operations </h1>
        <br>
        <br>
        <form action="action.jsp" method="post">
            <h1> Enter first number  :</h1> <input type="text" name ="n1"> <br/>


            <h1> Enter second number :</h1> <input type="text" name="n2"><br/>
            <br/>
            <br/>
                <input type="submit" value="calculate" />
        </form>
    </center>
        </div>
    </body>
</html>
```

action.jsp

```
<%--
    Document   : action
```

```jsp
    Created on : 3 May, 2021, 9:48:53 AM
    Author     : Murugan
--%>


<%@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>


    <%
      String n1 = request.getParameter("n1");
      String n2 = request.getParameter("n2");

      out.println("Arithmetic operations");
      out.println(" ");
      out.println("Given numbers are "+n1+"&"+n2+" ");


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

```
    try {

        com.calculator.CalculatorWebService_Service service = new
com.calculator.CalculatorWebService_Service();

        com.calculator.CalculatorWebService port =
service.getCalculatorWebServicePort();

         // TODO initialize WS operation arguments here

        float num1 = Float.parseFloat(n1);

        float num2 = Float.parseFloat(n2);

        // TODO process result here

        java.lang.String result = port.addition(num1, num2);

        out.println("Addition Result = "+result);

    } catch (Exception ex) {

        // TODO handle custom exceptions here

    }

    %>

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

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

    <%

    try {

        com.calculator.CalculatorWebService_Service service = new
com.calculator.CalculatorWebService_Service();

        com.calculator.CalculatorWebService port =
service.getCalculatorWebServicePort();

         // TODO initialize WS operation arguments here

      float num1 = Float.parseFloat(n1);

        float num2 = Float.parseFloat(n2);
```

```jsp
        // TODO process result here

        java.lang.String result = port.subtraction(num1, num2);

        out.println(" Subraction Result = "+result);

    } catch (Exception ex) {

        // TODO handle custom exceptions here

    }

    %>

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

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

    <%

    try {

        com.calculator.CalculatorWebService_Service service = new
com.calculator.CalculatorWebService_Service();

        com.calculator.CalculatorWebService port =
service.getCalculatorWebServicePort();

         // TODO initialize WS operation arguments here

       float num1 = Float.parseFloat(n1);

       float num2 = Float.parseFloat(n2);

       // TODO process result here

       java.lang.String result = port.multiplication(num1, num2);

       out.println("Multiplication Result = "+result);

    } catch (Exception ex) {

        // TODO handle custom exceptions here

    }

    %>
```

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

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

<%

try {

    com.calculator.CalculatorWebService_Service service = new
com.calculator.CalculatorWebService_Service();

    com.calculator.CalculatorWebService port =
service.getCalculatorWebServicePort();

     // TODO initialize WS operation arguments here

    float num1 = Float.parseFloat(n1);

    float num2 = Float.parseFloat(n2);

    // TODO process result here

    java.lang.String result = port.division(num1, num2);

    out.println("Division Result = "+result);

} catch (Exception ex) {

    // TODO handle custom exceptions here

}

%>

<%-- end web service invocation --%><hr/>
</html>
```
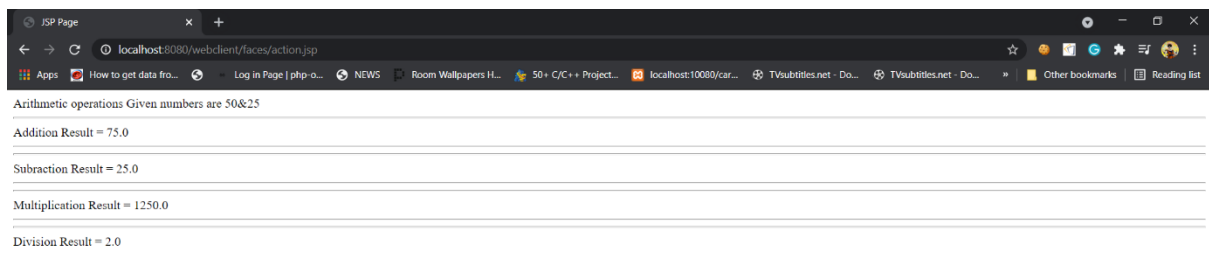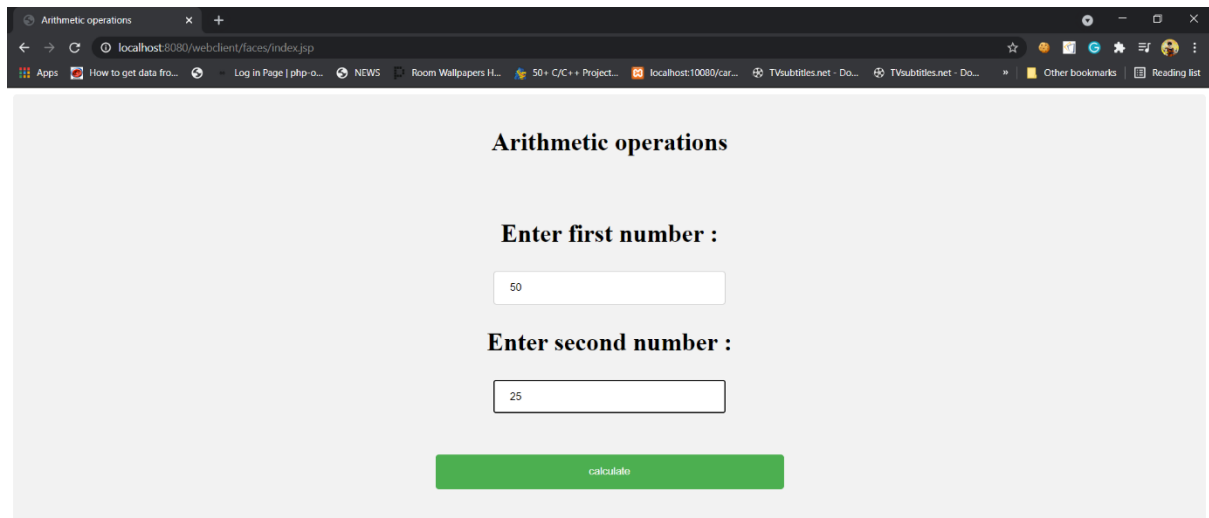
Output:

3. Creating a webservice for Car Rental system where input is given from the client and processed by the webservice.

Source Code:

Fare.jsp

```jsp
<%--

    Document   : fare

    Created on : 3 May, 2021, 10:22:12 AM

    Author     : Murugan

--%>


<html>

  <head>

    <title>Fare Calculation</title>

    <style>
input[type=date], select {

  width: 20%;

  padding: 12px 20px;

  margin: 8px 0;

  display: inline-block;

  border: 1px solid #ccc;

  border-radius: 4px;

  box-sizing: border-box;

}
input[type=text], select {

  width: 20%;

  padding: 12px 20px;
```

```css
  margin: 8px 0;

  display: inline-block;

  border: 1px solid #ccc;

  border-radius: 4px;

  box-sizing: border-box;

}


input[type=submit] {

  width: 30%;

  background-color: #4CAF50;

  color: white;

  padding: 14px 20px;

  margin: 8px 0;

  border: none;

  border-radius: 4px;

  cursor: pointer;

}


input[type=submit]:hover {

  background-color: #45a049;

}


div {

  border-radius: 5px;
```

```html
  background-color: #f2f2f2;

  padding: 20px;

}
</style>
  </head>
  <body>
    <div>
    <center>
      <h1>Car Rental Fare Calculation </h1>
      <br>
      <br>
      <form action="actioncar.jsp" method="post">
        <h1> Enter start date  :</h1> <input type="date" name ="d1"> <br/>


        <h1> Enter end date :</h1> <input type="date" name="d2"><br/>


        <h1> Enter fare amount per day :</h1> <input type="text"
name="amt"><br/>
        <br/>
        <br/>
          <input type="submit" value="calculate" />
      </form>
    </center>
      </div>
    </body>
```

actioncar.jsp

```jsp
<%--

    Document   : actioncar

    Created on : 3 May, 2021, 10:28:45 AM

    Author     : Murugan

--%>


<%@page import="java.util.Date"%>

<%@page import="java.text.SimpleDateFormat"%>

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

<!DOCTYPE html>

<html>

  <head>

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

    <title>Total fare JSP Page</title>

  </head>

  <body>


  <center>

    <h1>Car Rental Fare Calculation </h1>

    <br/>

    <br/>

    <br/>
```

```jsp
<%
   String n1 = request.getParameter("d1");

   String n2 = request.getParameter("d2");

  String fare = request.getParameter("amt");

 float nod=0;

   out.println("Car Rental Fare Calculation");

    %>

    <br/>

    <br/>

    <%

   Date date = new Date();

   out.print("Current date and time is "+"<h2 align = \"center\">"
+date.toString()+"</h2>");

      %>

      <br/>

      <br/>

      <%

   out.println("Given start and end dates are "+n1+" & "+n2+" ");

     %>

     <br/>

     <br/>

     <%

  out.println("Given Fare amount is  "+fare+" ");


   try {
```

```jsp
        SimpleDateFormat myFormat = new SimpleDateFormat("yyyy-mm-dd");

        Date dateBefore = myFormat.parse(n1);

        Date dateAfter = myFormat.parse(n2);

        long difference = dateAfter.getTime() - dateBefore.getTime();

        float daysBetween = (difference / (1000*60*60*24));

    nod = daysBetween;

    /* You can also convert the milliseconds to days using this method

     * float daysBetween =

     *       TimeUnit.DAYS.convert(difference, TimeUnit.MILLISECONDS)

     */

    %>

    <br/>

    <br/>

    <%

        out.println("Number of Days between starting and ending date is : "+daysBetween);

    } catch (Exception e) {

        e.printStackTrace();

    }

        %>


        <br/>
```

```jsp
            <br/>

            <br/>

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

    <%

    try {

        com.calculator.CalculatorWebService_Service service = new
com.calculator.CalculatorWebService_Service();

        com.calculator.CalculatorWebService port =
service.getCalculatorWebServicePort();

         // TODO initialize WS operation arguments here

        float num1 = Float.parseFloat(fare);

        float num2 = nod;

        // TODO process result here

        java.lang.String result = port.multiplication(num1, num2);

        out.println("Total fare = "+result);

    } catch (Exception ex) {

        // TODO handle custom exceptions here

    }

    %>

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

        </center>

    </body>

</html>
```
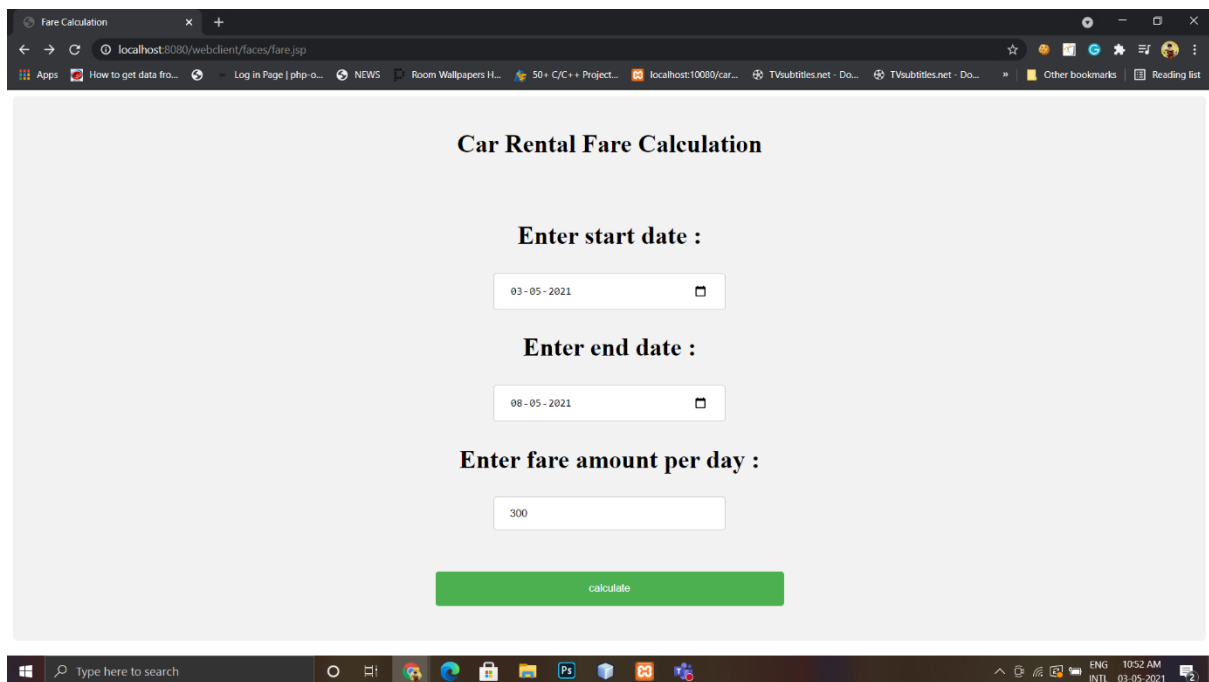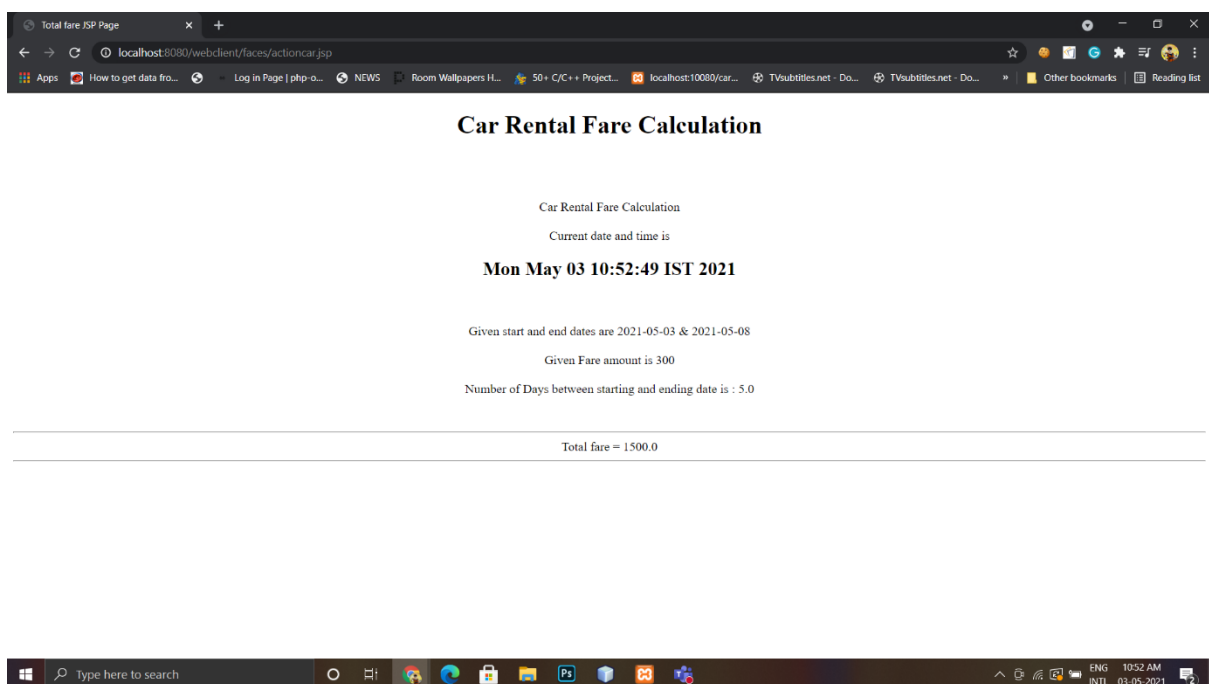
Output:

Here the start date and end date and fare per day is given as input and difference between the dates is calculated in jsp and the corresponding fare is calculated using the webservice. The service code used is the same code which is tested in (1) which is used to calculate the fare amount.