U19CS311 INTERNET PROGRAMMING LABORATORY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

SRI ESHWAR COLLEGE OF ENGINEERING
KINATHUKADAVU COIMBATORE - 641202

Sri Eshwar College of Engineering
An Autonomous Institution
Affiliated to Anna University, Chennai

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

BONAFIDE CERTIFICATE

Certified that this is the bonafide record of work done by

Name: Mr. /Ms. ...…….

Register No: ……... of 3rd Year

B.Tech – Artificial Intelligence and Data Science in the **U19CS311 – INTERNET PROGRAMMING LABORATORY** during the 5th **Semester** of the academic year **2022 – 2023 (Odd Semester).**

Signature of faculty In-charge Head of the department

Submitted for the practical examinations of Anna University, held on …….

Internal Examiner External Examiner

Contents

Average:

Average (in words) Signature of the Faculty

Ex. No. 1 a)	MODULE 1 – HTML AND CSS
Date:	**Create a web page with the following using HTML** **• To embed an image map in a web page** **• To fix the hot spots** **• Show all the related information when the hotspot is clicked**

AIM:

To create a webpage with the following using HTML

• To embed an image in web page

• To fix the hot spots.

• Show all the related information when a hot spot is clicked in the map

ALGORITHM:

Step 1:Create a html file with map tag.

Step 2 Set the source attribute of the img tag to the location of the image and also set the use map attribute.

Step 3 Specify an area with name, shape and href set of the appropriate value.

Step 4 Repeat step3 as many hot spots you want to include in the image

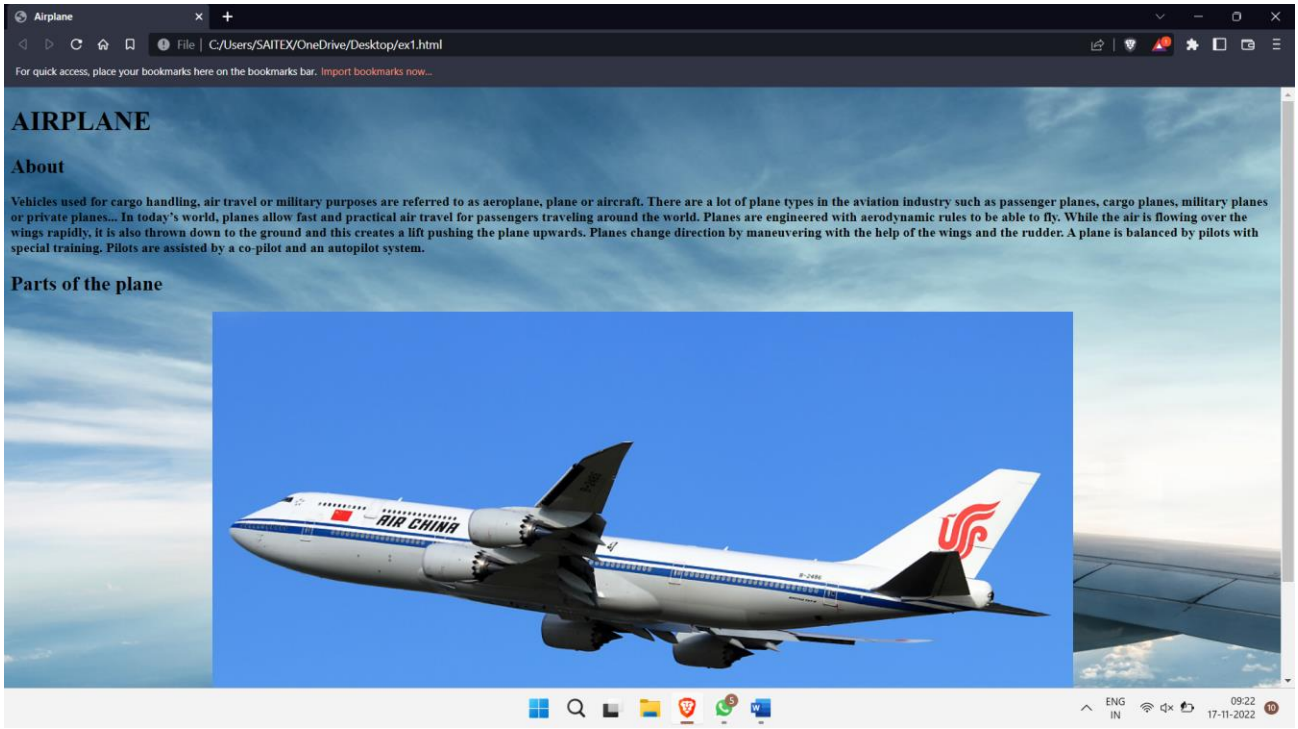
Step5 Create html file for each and every hot spots the user will select the particular location it shows information about it.

PROGRAM:

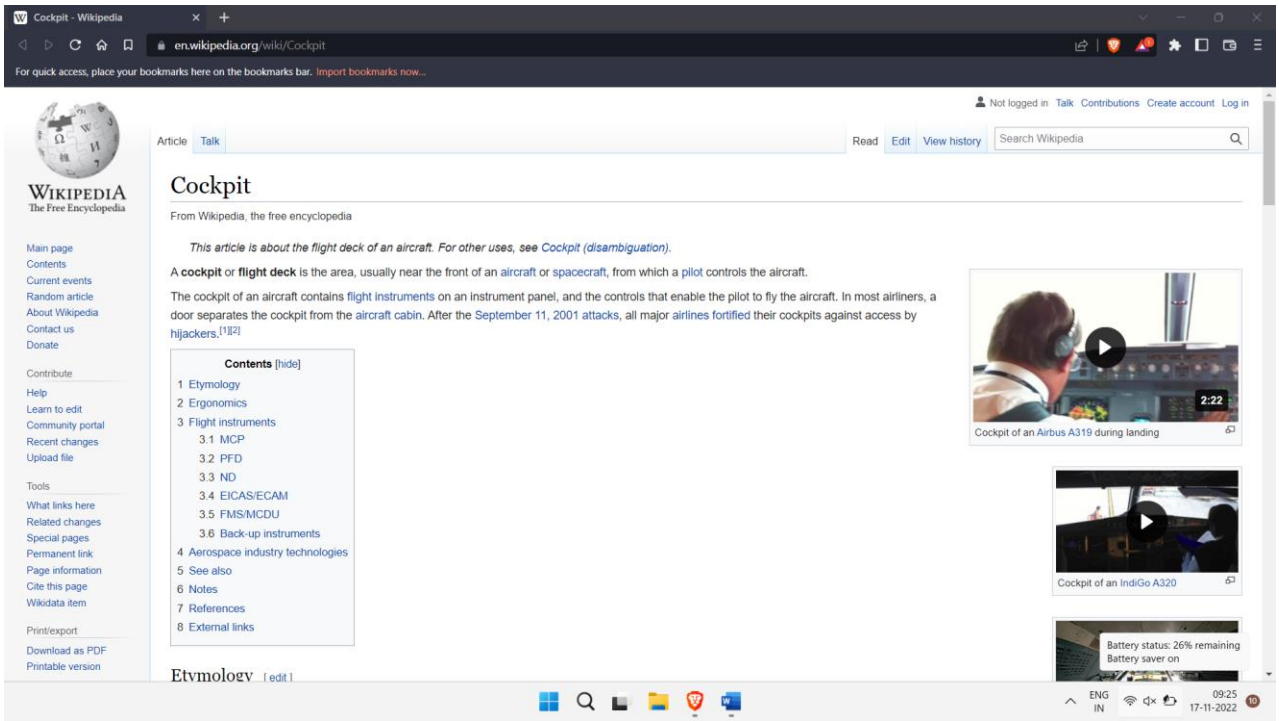
```
<!DOCTYPE html>
<html>
<head>
<title>Airplane</title>
<style>
body {
background-image: url('https://cdn.wallpapersafari.com/61/55/Q4rJpl.jpg');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
</style>
</head>
<body>
```

```
<h1>AIRPLANE</h1>

<h2>About</h2>

<p><b>Vehicles used for cargo handling, air travel or military purposes are referred to as aeroplane, plane or aircraft. There are a lot of plane types in the aviation industry such as passenger planes, cargo planes, military planes or private planes... In today’s world, planes allow fast and practical air travel for passengers traveling around the world.

Planes are engineered with aerodynamic rules to be able to fly. While the air is flowing over the wings rapidly, it is also thrown down to the ground and this creates a lift pushing the plane upwards. Planes change direction by maneuvering with the help of the wings and the rudder. A plane is balanced by pilots with special training. Pilots are assisted by a co-pilot and an autopilot system.</b></p>

<h2>Parts of the plane</h2>

<center>

<img src="https://news.cgtn.com/news/2022-06-15/China-aims-for-3-500-registered-general-aviation-airplanes-by-2025-1aSwozIm98Y/img/6fe12966f0bf4d6abe6a1bd99f1c42c3/6fe12966f0bf4d6abe6a1bd99f1c42c3.png" alt="parts" usemap="#Parts">

<map name="Parts">

  <area shape="rect" coords="81,84,131,340" alt="cockpit" href="https://en.wikipedia.org/wiki/Cockpit">


  <area shape="rect" coords="256,193,701,459" alt="Wheels" href="https://www.flyingmag.com/guides/how-many-wheels-do-airplanes-have/#:~:text=Most%20commercial%2C%20transport%2Dcategory%20airplanes,but%20counts%20as%20one%20assembly">


  <area shape="rect" coords="765,131,990,422" alt="turbine" href="https://www.lsptechnologies.com/resources/how-do-aircraft-engines-work/#:~:text=Turbine%3A%20The%20turbine%20section%20is,the%20front%20of%20the%20engine">


</map>

<center>

</body>

</html>
```

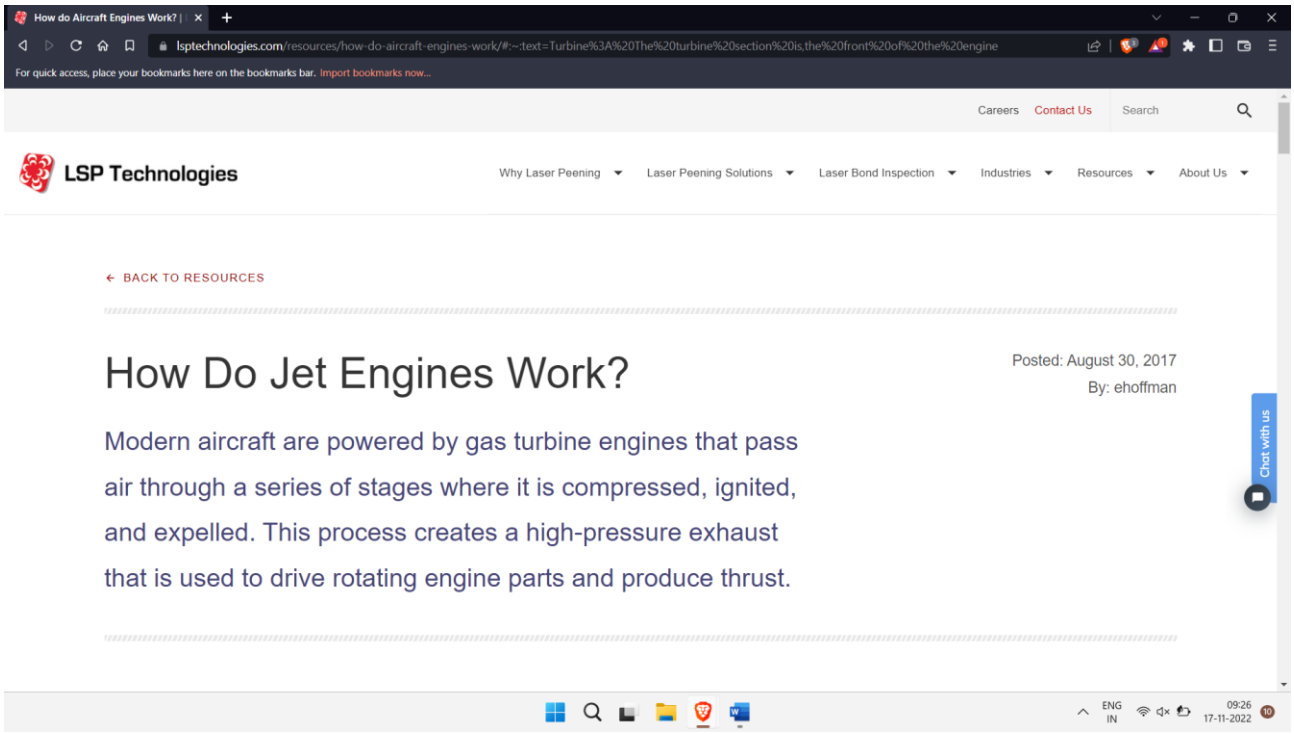
OUTPUT:



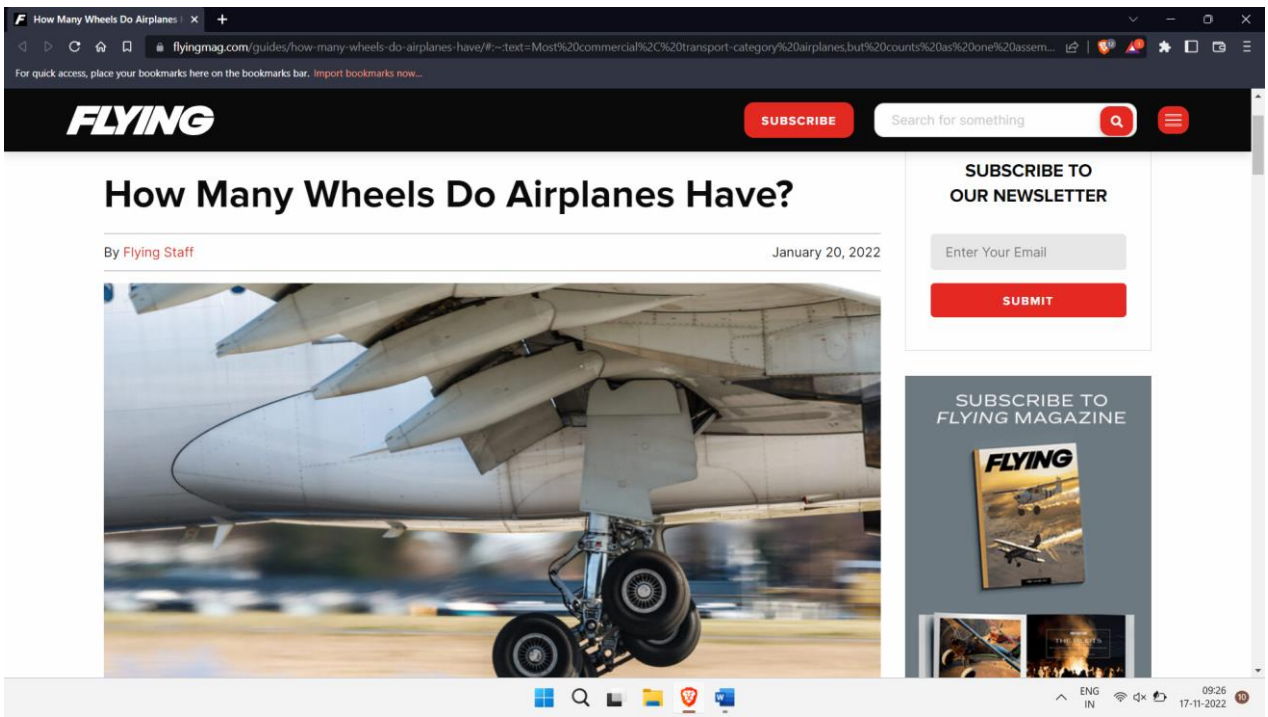
On clicking airplane's cockpit:



On clicking airplane’s Engines:



On clicking airplane’s Wheels:



RESULT:

Thus creation of a webpage using hotspots in the image has been developed successful

Ex. No. 1 b)	MODULE 1 – HTML AND CSS
Date:	Create a web page with all types of Cascading Style Sheets

AIM:

To create a webpage with the following using html to embedded the style sheet.

ALGORITHM:

Step1: Create html file with the style tag, inside head tag.

Step2: Set the style such as font-family, font-size, color, left etc, for the headingh1,h2,…h6 and respectively.

Step3: Close the head tag.

Step4: Specify the heading and information required inside the body tag.

Step5: Close the opened tag.

PROGRAM:

Ex2.html:

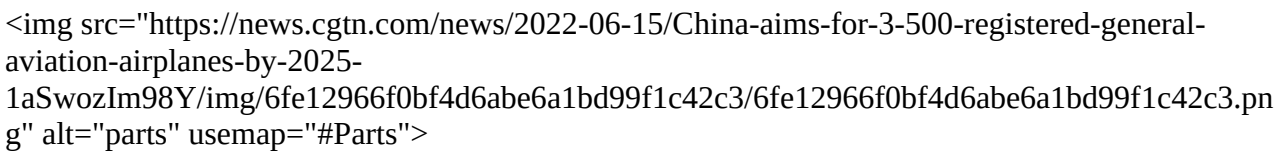
```
<!DOCTYPE html>
<html>
<head>
<title>Airplane</title>
<link rel="stylesheet" href ="C:\Users\USER\Downloads\ext.css">
<style>
body {
background-image: url('https://cdn.wallpapersafari.com/61/55/Q4rJpl.jpg');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}

h1 {text-align: center;}
h2{style=color:Black;}

p{
 font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>
<body>
<h1>AIRPLANE</h1>
```

<h2>About</h2>

Vehicles used for cargo handling, air travel or military purposes are referred to as aeroplane, plane or aircraft. There are a lot of plane types in the aviation industry such as passenger planes, cargo planes, military planes or private planes... In today’s world, planes allow fast and practical air travel for passengers traveling around the world.

Planes are engineered with aerodynamic rules to be able to fly. While the air is flowing over the wings rapidly, it is also thrown down to the ground and this creates a lift pushing the plane upwards. Planes change direction by maneuvering with the help of the wings and the rudder. A plane is balanced by pilots with special training. Pilots are assisted by a co-pilot and an autopilot system.

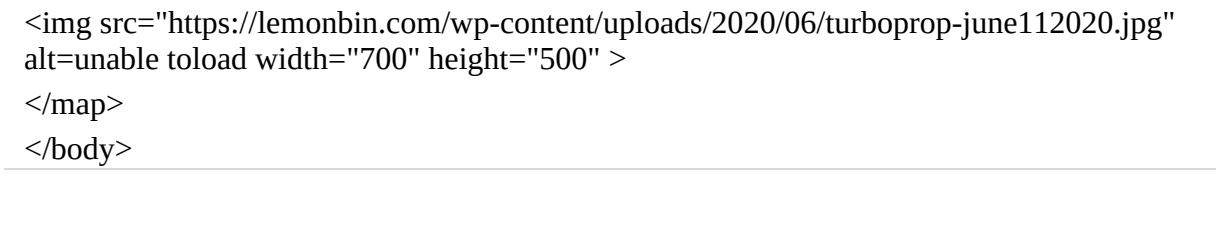
<h2>Parts of the plane</h2>



<h2>Types of Airplanes:</h2>

<h3 font-size:25px >Turboprop Aircraft:</h3>

The turboprop aircraft has one or more gas turbine engines. They are both connected to a gearbox which turns the propellers. This is contrary to the pistons or jet engines that are found in the other types of small aircraft. The turboprop airplanes are usually much bigger than the piston aircraft that they resemble and can fly at much higher altitudes of 35,000 feet. They are well suited to flying for 600 to 1000 miles in a single flight, and a much cheaper option to private jets



</map>

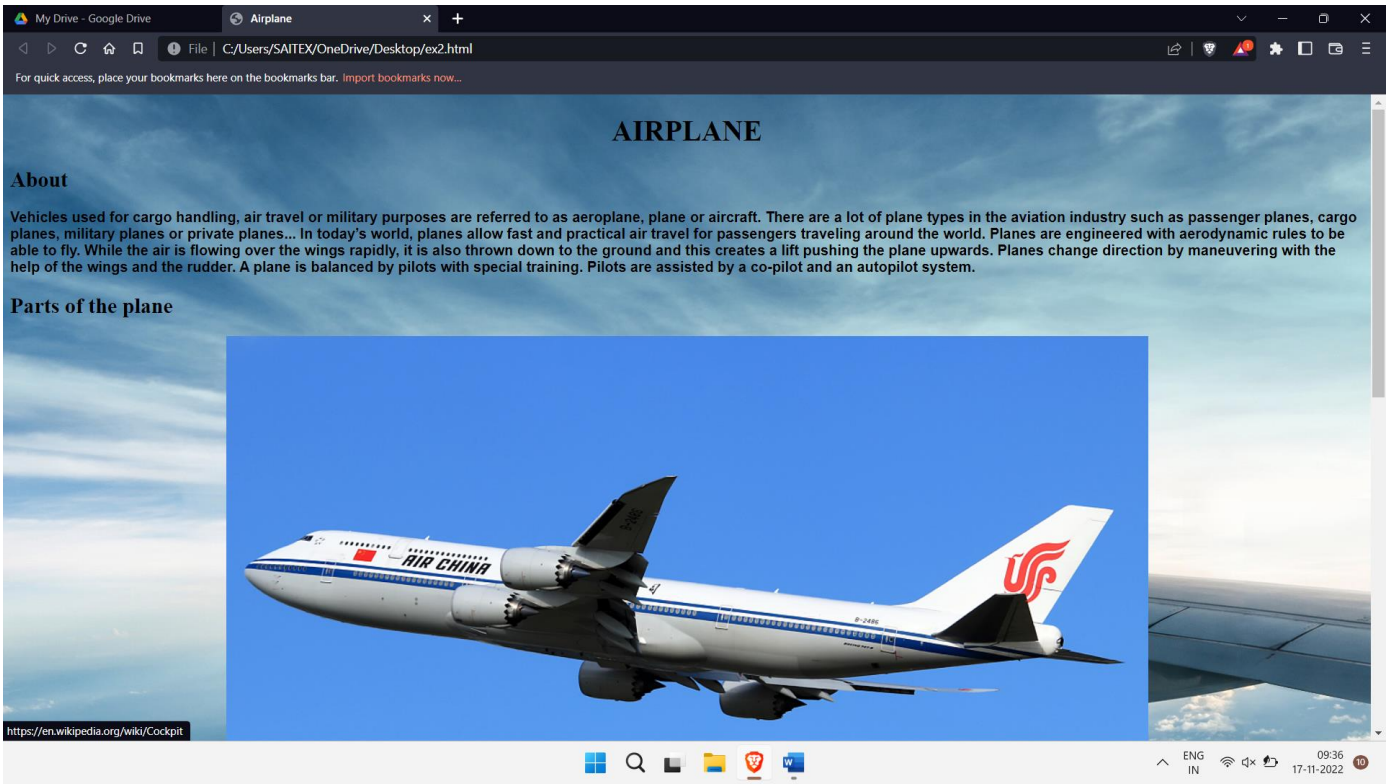
</body>

```
</html>
```

Ext.css

```
h1 {

font-size: 40px;

}
h2 {
font-size:30px;
}

p {

font-size: 20px;
}
```

OUTPUT:



RESULT:

Thus creation of a webpage using HTML and CSS has been developed successfully.

Ex. No. 2 a)	MODULE II CLIENT SIDE SCRIPTING
Date:	**Develop a java script page to validate the following fields in a registration page** **i)Name (should contains alphabets and the length should not be less than 6 characters).** **ii)Password(should not be less than 6 characters)** **iii)E-mail(should not contain invalid addresses)**

AIM

To Write JavaScript to validate the following fields of the above registration page.

ALGORITHM

Step 1**:** Name (Name should contain alphabets and the length should not be less than 6characters).

Step 2: Password (Password should not be less than 6 characters length).

Step 3: E-mail id (should not contain any invalid and must follow the standard patternname@domain.com)

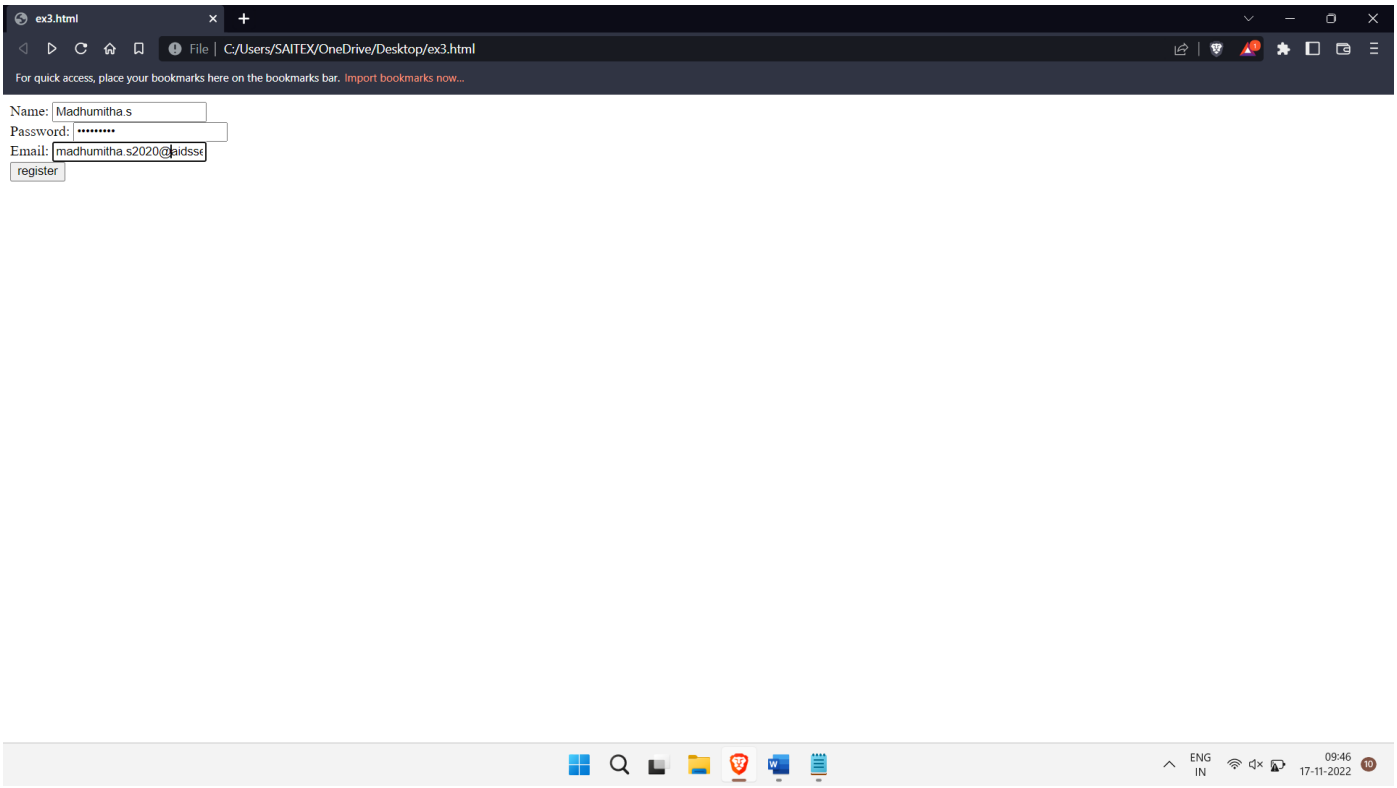
Step 4: Phone number (Phone number should contain 10 digits only).

PROGRAM

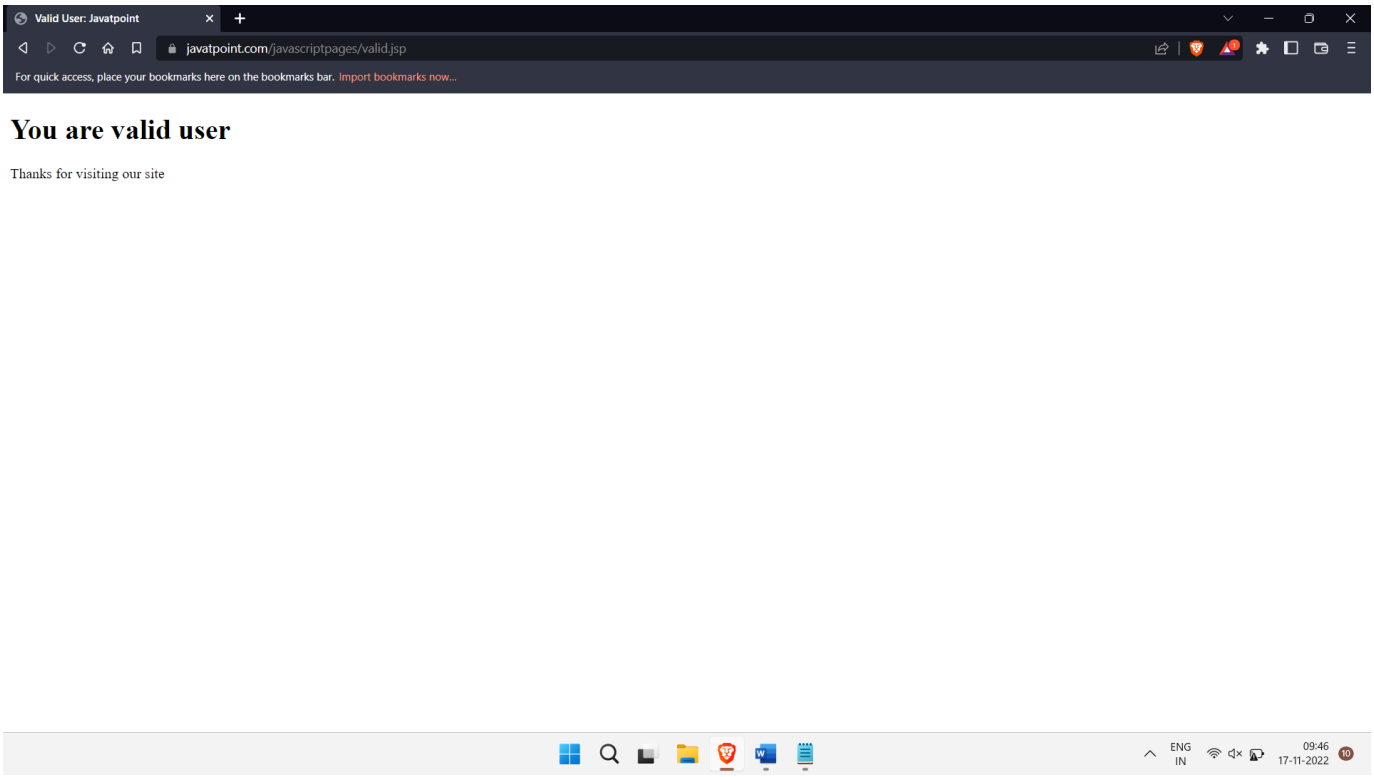
```
<html>
<body>
<script>
function validateform(){
var name=document.myform.name.value;
var password=document.myform.password.value;
var x=document.myform.email.value;
var atposition=x.indexOf("@");
var dotposition=x.lastIndexOf(".");
if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){
 alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n
dotposition:"+dotposition);
return false;
if (name==null || name==""){
alert("Name can't be blank");
 return false;
}else if(password.length<6){
alert("Password must be at least 6 characters long.");
return false;
}
}
</script>
<body>
<form name="myform" method="post"
action"http://www.javatpoint.com/javascriptpages/valid.jsp" onsubmit="return
```

```
validateform()" >
Name: <input type="text" name="name"><br/>
Password: <input type="password" name="password"><br/>
Email: <input type="text" name="email"><br/>
<input type="submit" value="register">
</form>
</body>
</html>
```

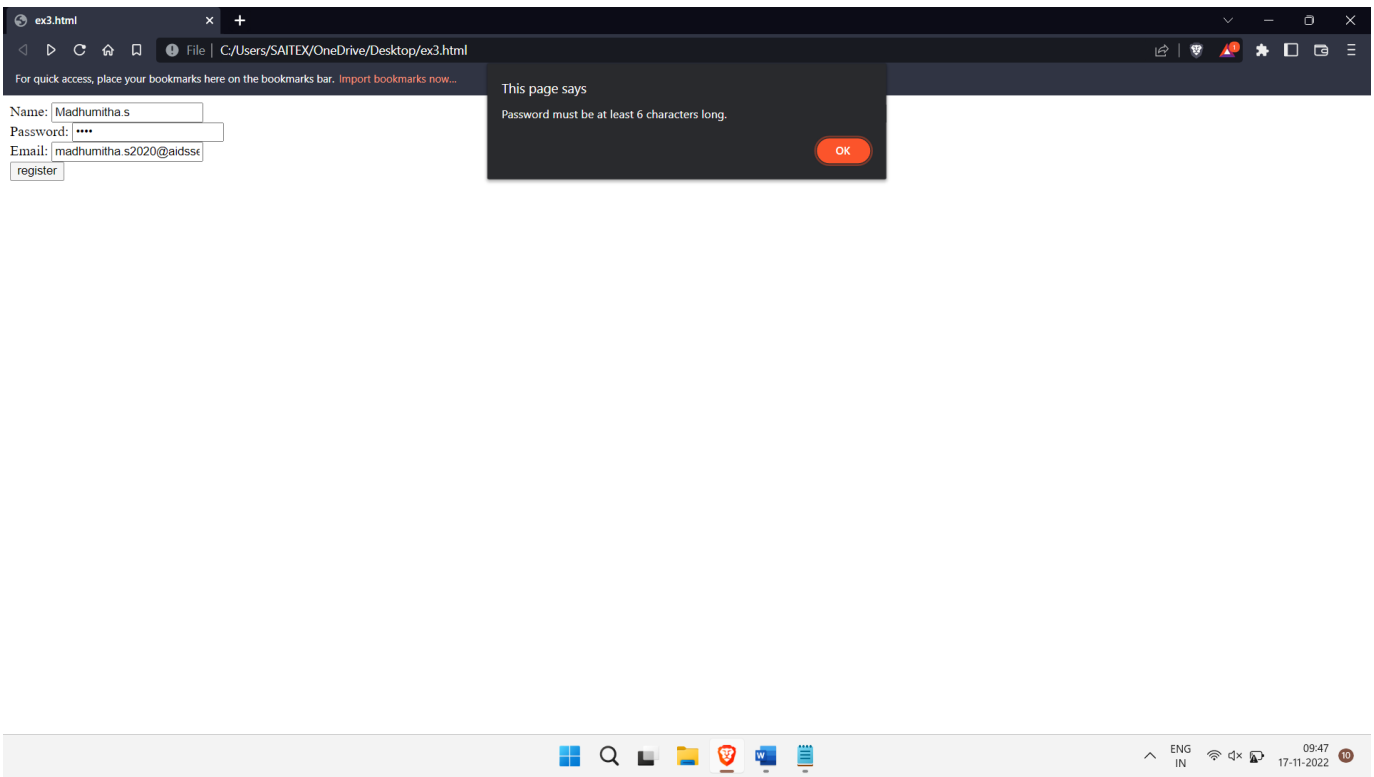
OUTPUT:



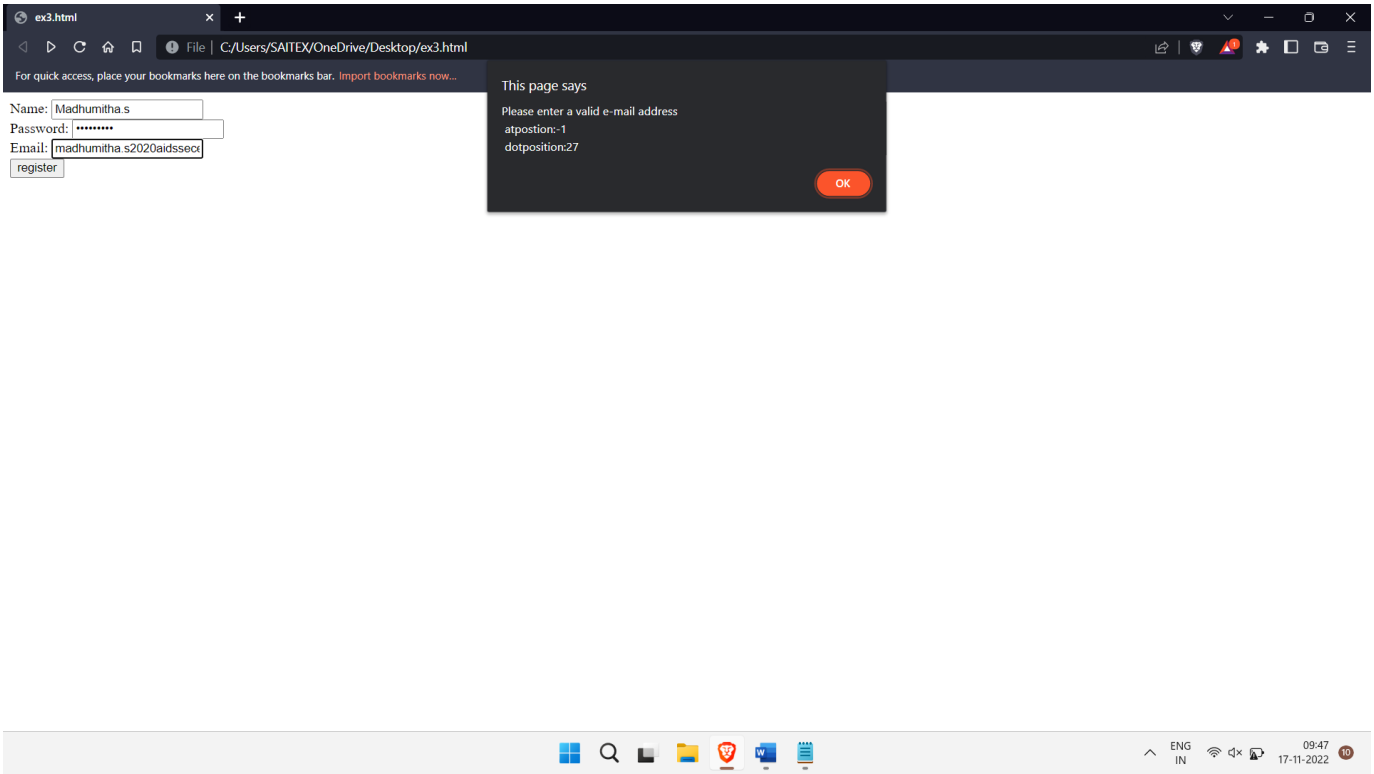
Valid Information:



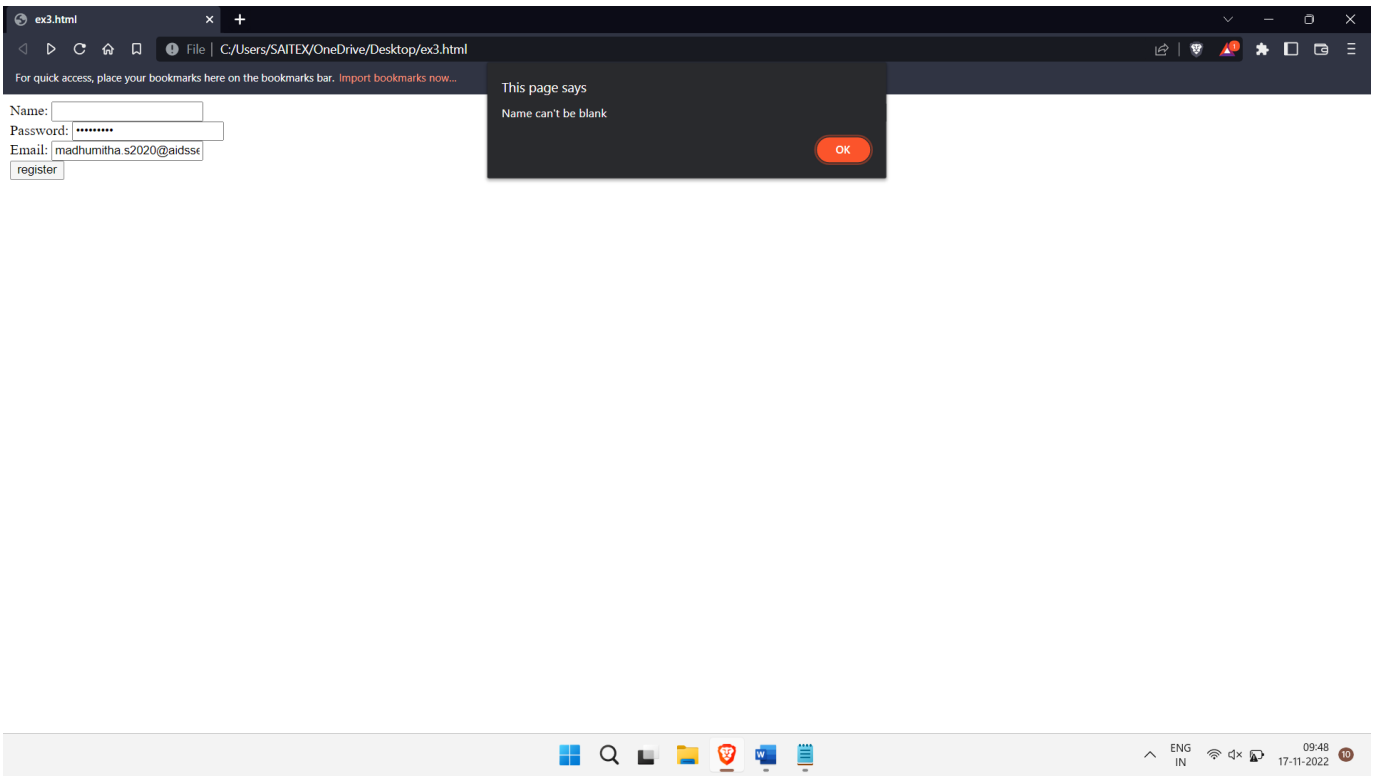
Invalid Password:



Invalid Email:



Invalid Name:



RESULT:

Thus the fields name, password and E-mail of the java script webpage has been validatedsuccessfully.

Ex. No. 2 b)	MODULE II CLIENT SIDE SCRIPTING
Date:	**Write a java script to get the nth largest element from an unsorted array.**

AIM

To develop a java script to get the nth largest element from an unsorted array.

ALGORITHM

Step 1:Create a function to find out the largest element

Step 2:Initialize the variables

Step 3:Using while loop for checking the largest element

Step 3:Print the largest element in the given array.

PROGRAM

```
var arr=[70,30,20,40,50];
let len=arr.length;
for(let i=0;i<len;i++){
   for(let j=0;j<len;j++){
      var temp=0;
      if(arr[i]>arr[j]){
         temp=arr[i];
         arr[i]=arr[j];
         arr[j]=temp;
      }
   }
}
document.write(arr);
let n=3;
document.write(arr[n-1]);
```

OUTPUT

main.js

Run

Output

```
var arr=[70,30,20,40,50];
let len=arr.length;
for(let i=0;i<len;i++){
    for(let j=0;j<len;j++){
        var temp=0;
        if(arr[i]>arr[j]){
            temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
        }
    }
}
document.write(arr);
let n=3;
document.write(arr[n-1]);
```

```
node /tmp/zt1NrqjG5P.js
[ 70, 50, 40, 30, 20 ]
40
```

RESULT

Thus the java script to get the nth largest element from an unsorted array is successfully executed.

Ex. No. 3 a)	MODULE III SERVER SIDE PROGRAMMING
Date:	**Write a programs in java using servlets** **i) To invoke servlets from HTML forms** **ii) To invoke servlets from Applets**

AIM

To invoke servlets from HTML Forms

ALGORITHM

- Start the Invoking Servlets from HTML FormsCreate the postparm.html
- Use the two input type for Employee name and phono. The save the postparm.html
- Create java serve let for invoke the html forms
- After creating java servlets extracting PostParam.war .

Then follow the step

Step 1: Open Web Browser and type

Step 2: http://localhost:8080

Step 3: Select Tomcat Manager

Step 4: Deploy the war file and Run

PROGRAM

PostParam.html

```
<HTML>
<BODY>
<CENTER>
<FORM name = "postparam" method = "post"
action="http://localhost:8080/PostParam/PostParam">
<TABLE>
<tr>
<td><B>Employee </B> </td>
<td><input type = "textbox" name="ename" size="25"
value=""></td>
</tr>
```

```
<tr>
<td><B>Phone </B> </td>
<td><input type = "textbox" name="phoneno" size="25"
value=""></td>
</tr>
</TABLE>
<INPUT type = "submit" value="Submit">
</body>
</html>
```

Invoking Servlets from HTML Forms source code java programming

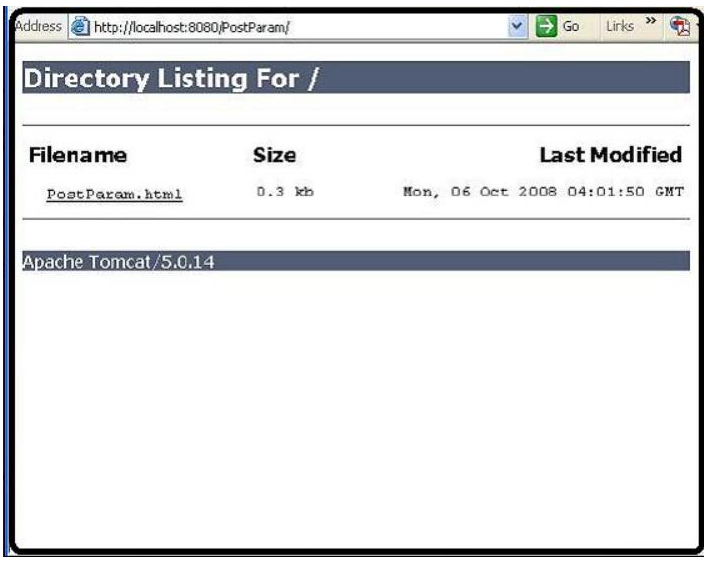
```
import java.io.*;
import java.util.*;
import javax.servlet.*;
public class PostParam extends GenericServlet {
public void service(ServletRequest
request,ServletResponse response) throws
ServletException, IOException {
PrintWriter pw = response.getWriter();
Enumeration e = request.getParameterNames();
while(e.hasMoreElements()) {
String pname = (String)e.nextElement();
pw.print(pname + " = ");
String pvalue = request.getParameter(pname);
pw.println(pvalue);
}
pw.close();
}
}
```

Invoking Servlets from HTML Forms source code xml programing

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
```

```
2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
<display-name>Welcome to Tomcat</display-name>
<description>
Welcome to Tomcat
</description>
<!-- JSPC servlet mappings start -->
<servlet>
<servlet-name>PostParam</servlet-name>
<servlet-class>PostParam</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>PostParam</servlet-name>
<url-pattern>/PostParam</url-pattern>
</servlet-mapping>
```

OUTPUT



RESULT

Thus the program

i) To invoke servlets from HTML forms.

ii) To invoke servlets from Applets is successfully executed.

Ex. No. 3 b)	MODULE III SERVER SIDE PROGRAMMING
Date:	**Write programs in Java to create three-tier applications using JSP and Databases for conducting on-line examination for displaying student mark list. Assume that student information is available in database which has been stored in a database server.**

AIM

To write a java program to create three-tier applications using JSP and Databases forconducting on-line examination for displaying student mark list.

ALGORITHM

Step 1:Go to Control Panel and Select Administrative Tools and then select Data Source ODBC icon.

(or)

Go to Run(Press Window+R) and type odbcad32.exe to go Data Source ODBC.

Step 2:Press Add Button.

Step 3:Choose the driver for Microsoft Access.

Step 4:Then,Press Finish Button.

Step 5:Type Data Source Name as ExamStudent.

Step 6:Then Press "Select Button" and choose the database file which is created already.

Step 7:Then, Press "OK" Button to complete the process.

Steps for Creating Database and table for this Program:

1.Create Ms Access File in any Name

2.Then Open it .

3.Create the table in the name of StudentTable With following columns Seat_no,Name,Marks

PROGRAM

ExamServer.jsp:

```
<%@page contentType="text/html" language="java" import="java.sql.*"%>
<html>
<head>
<title>Online Exam Server</title>
<style type="text/css">
  body{background-color:black;font-family:courier;color:blue}
</style>
</head>
<body>
<h2 style="text-align:center">ONLINE EXAMINATION</h2>
<p>
<a href="ExamClient.html">Back To Main Page</a>
</p>
<hr/>
```

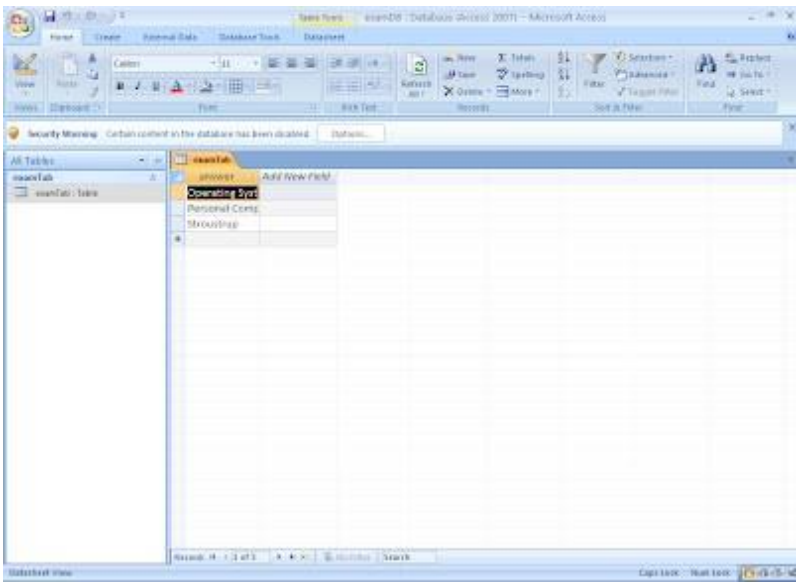
```
<%
String str1=request.getParameter("ans1");
String str2=request.getParameter("ans2");
String str3=request.getParameter("ans3");
int mark=0;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:examDS");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("SELECT * FROM examTab");
int i=1;
while(rs.next())
{
 if(i==1)
 {
  String dbans1=rs.getString(1);
  if(str1.equals(dbans1))
  {
   mark=mark+5;
  }
 }
 if(i==2)
 {
  String dbans2=rs.getString(1);
  if(str2.equals(dbans2))
  {
   mark=mark+5;
  }
 }
 if(i==3)
 {
  String dbans3=rs.getString(1);
  if(str3.equals(dbans3))
  {
   mark=mark+5;
  }
 }
 i++;
}
if(mark>=10)
{
 out.println("<h4>Your Mark Is : "+mark+"</h4>");
 out.println("<h3>Congratulations....! You Are Eligible For The Next Round...</h3>");
}
else
{
```

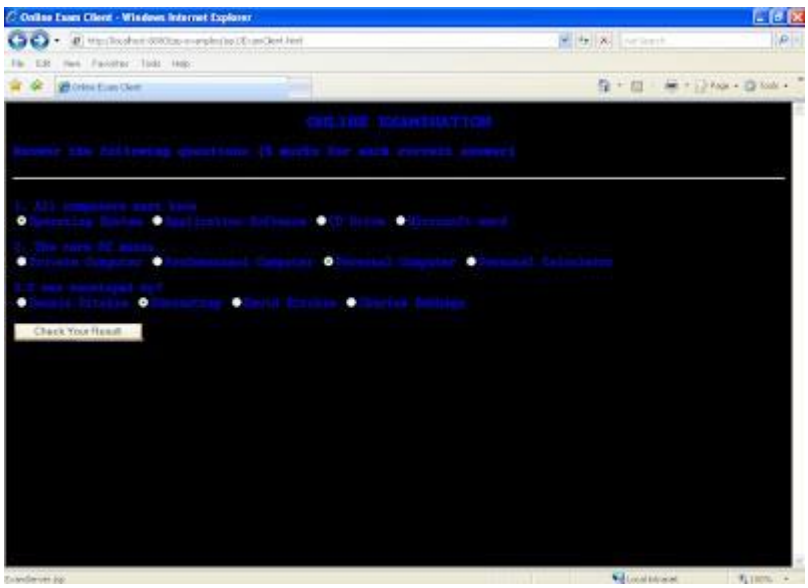
```
 out.println("<h4>Your Mark is : "+mark+"</h4>");
 out.println("<h3>Sorry....!! You Are Not Eligible For The Next Round...</h3>");
}
%>
</form>
</body>
</html>
```

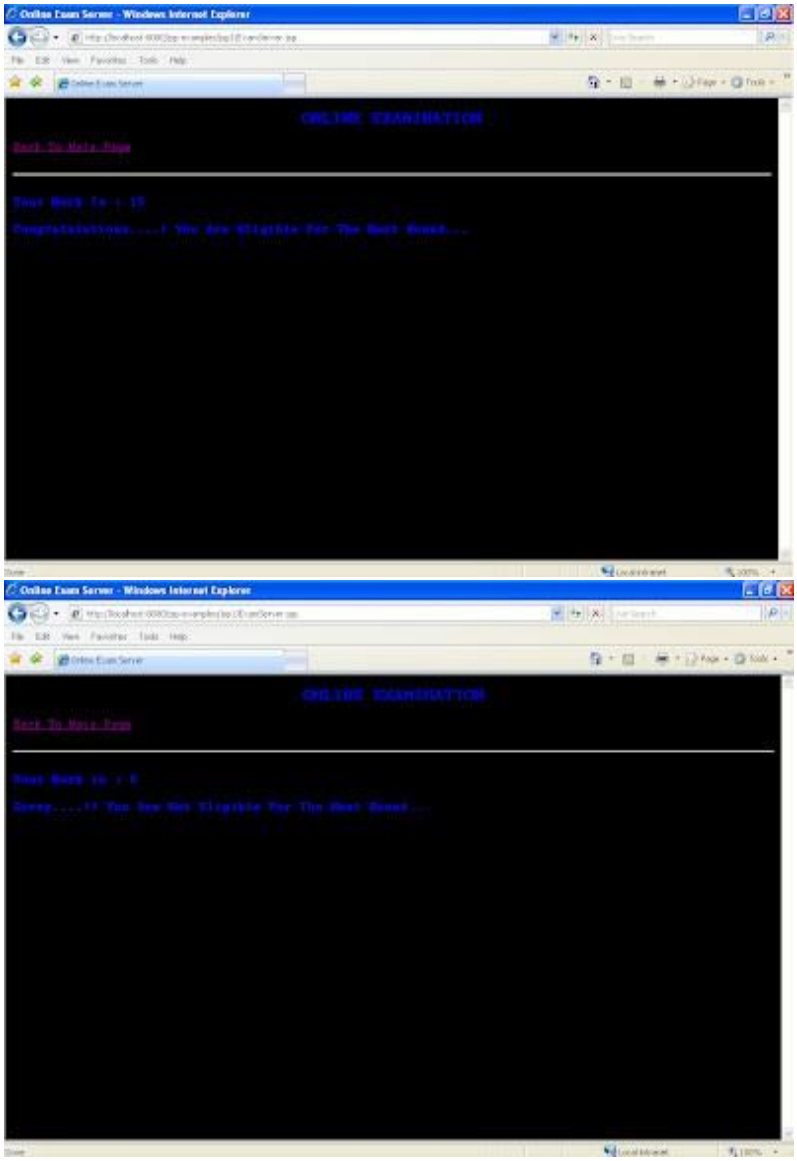
ExamClient.HTML:

```
<html>
<head>
<title>Online Exam Client</title>
<style type="text/css">
  body{background-color:black;font-family:courier;color:blue}
</style>
</head>
<body>
<h2 style="text-align:center">ONLINE EXAMINATION</h2>
<h3>Answer the following questions (5 marks for each correct answer)</h3>
<hr/>
<form name="examForm" method="post" action="ExamServer.jsp">
1. All computers must have <br/>
<input type="radio" name="ans1" value="Operating System">Operating System
<input type="radio" name="ans1" value="Application Software">Application Software
<input type="radio" name="ans1" value="CD Drive">CD Drive
<input type="radio" name="ans1" value="Microsoft word">Microsoft word
<br/><br/>
2. The term PC means <br/>
<input type="radio" name="ans2" value="Private Computer">Private Computer
<input type="radio" name="ans2" value="Professional Computer">Professional Computer
<input type="radio" name="ans2" value="Personal Computer">Personal Computer
<input type="radio" name="ans2" value="Personal Calculator">Personal Calculator
<br/><br/>
3.C was developed by?<br/>
<input type="radio" name="ans3" value="Dennis Ritchie">Dennis Ritchie
<input type="radio" name="ans3" value="Stroustrup">Stroustrup
<input type="radio" name="ans3" value="David Ritchie">David Ritchie
<input type="radio" name="ans3" value="Charles Babbage">Charles Babbage
<br/><br/>
<input type="submit" value="Check Your Result"/>
</form>
</body>
</html>
```

OUTPUT




Online Exam Client - Windows Internet Explorer
Check Your Result



RESULT

Thus the java program to create three-tier applications using JSP and Databases forconducting on-line examination for displaying student mark is successfully executed.

Ex. No. 4 a)	MODULE IV WEB DATA REPRESENTATION
Date:	**Create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document.**

AIM

Create and save an XML document at the server, which contains 10 users information. Write a program which takes User Id as input and returns the user details by taking the user information from the XML document.

ALGORITHM

Step 1:Write a xml document to fetch the user details

Step 2:Create a html code to pass the function to display the details.

Step 3:The function access the user details based on the given id.

PROGRAM

```
<?xml version=”1.0”?>
<userlist>
<user>
<userid>usr01</userid>
<username>Gouse</username>
<address>DSNR</address>
<phone>8801550101</phone>
<email>Gouse.sheikh@gmail.com</email>
</user>
<user>
<userid>usr02</userid>
<username>D Divakar</username>
<address>Ameerpet</address>
<phone>9888888888</phone>
<email>D Divakar@gmail.com</email>
</user>
<user>
<userid>usr03</userid>
<username>Rajinth</username>
<address>SR Nagar</address>
<phone>9866666666</phone>
<email>Rajinth@yahoo.com</email>
</user>
<user>
```

<userid>usr04</userid>
<username>M Vijaya</username>
<address>DESHMUKHI</address>
<phone>9835994445</phone>
<email>M Vijaya@yahoo.com</email>
</user>
<user>
<userid>usr05</userid>
<username>Kusuma</username>
<address>KOTI</address>
<phone>968877555</phone>
<email>Kusuma@yahoo.com</email>
</user>
<user>
<userid>usr06</userid>
<username>P Kalpana</username>
<address>KOTI</address>
<phone>968875554</phone>
<email>P Kalpana@yahoo.com</email>
</user>
<user>
<userid>usr07</userid>
<username>V Anitha</username>
<address>KPHB</address>
<phone>968888554</phone>
<email>V Anitha@yahoo.com</email>
</user>
<user>
<userid>usr08</userid>
<username>M Ramesh</username>
<address>IBP</address>
<phone>968878554</phone>
<email>M Ramesh@yahoo.com</email>
</user>
<user>
<userid>usr09</userid>
<username>Raswitha</username>
<address>DSNGR</address>
<phone>968899554</phone>
<email>Raswitha@yahoo.com</email>

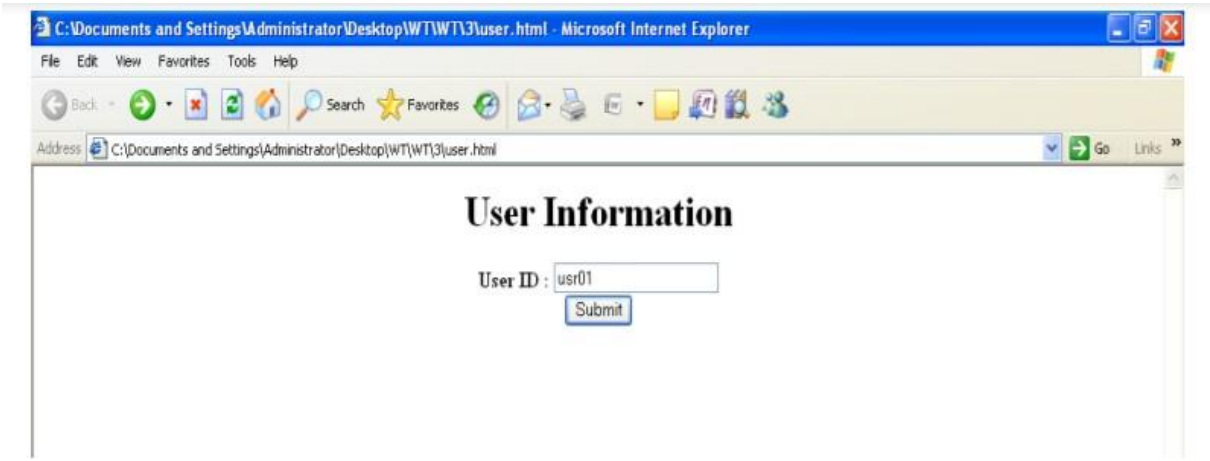
```
</user>
<user>
<userid>usr10</userid>
<username>A Prasanna</username>
<address>KPHB</address>
<phone>9088008554</phone>
<email>A Prasanna@yahoo.com</email>
</user>
</userlist>
```

User.html

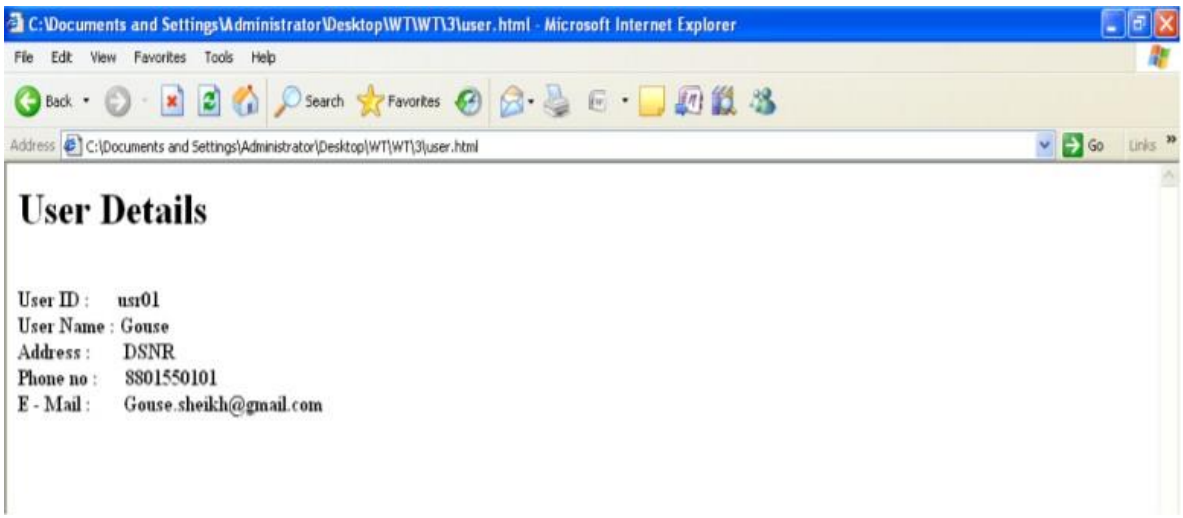
```
<html>
<head>
<script language="javascript">
function fncDisplayInfo()
{
var
xhttp=n
ull;var
flag=0;
var userid = document.frm.uname.value;
var xmlDoc = new
ActiveXObject("microsoft.xmldom");
xmlDoc.load("user.xml");
var noOfUsers =
xmlDoc.getElementsByTagName("userlist")[0].childNodes.length;for(var
i=0;i<parseInt(noOfUsers);i++)
{
var uid
=xmlDoc.getElementsByTagName("user")[i].childNodes[0].childNodes[0].node
Value;if(uid == userid)
{
 document.write("<h1> User
Details</h1>");var userName =
xmlDoc.getElementsByTagName("user")[i].childNodes[1].childNodes[0].node
Value;var Address
=xmlDoc.getElementsByTagName("user")[i].childNodes[2].childNodes[0].node
Value;var phone =
xmlDoc.getElementsByTagName("user")[i].childNodes[3].childNodes[0].nodeV
alue; var email=
```

```
        xmlDoc.getElementsByTagName("user")[i].childNodes[4].childNodes[0].node
    Value;document.write("<br><b>User ID
    :      "+uid) document.write("<br>User Name
    : "+userName);
    document.write("<br>Address :      
    "+Address);document.write("<br>Phone no :
         "+phone); document.write("<br>E - Mail :
          "+email); flag =1;
    break;
    }
    }
    if(flag==0)
    {
    alert("InValid User");
    }
    }
    </script>
    </head>
    <body>
<center>
    <h1><b>User Information</h1>
     <form name="frm">
    User ID : <input type="text" name="uname"><br>
    <input type="button" name="btn" value="Submit" onclick="fncDisplayInfo()">
    </form>
    </center>
    </body>
    </html>
```

OUTPUT



OUTPUT



RESULT

Thus an XML document at the server, which contains 10 users Information and takes user Id as an input and returns the User details by taking the user information from the XML document, was executed successfully.

Ex. No. 4 b)	MODULE IV WEB DATA REPRESENTATION
Date:	**Develop an XSLT code convert an XML document to a HTML table**

AIM

To develop an XSLT code convert an XML document to a HTML table.

ALGORITHM

Step 1: Create gradle or maven based project in Eclipse. The name of the project is java-xslt-xml-to-html

Step 2: Put the XML file books.xml under src/main/resources/xml directory.

Step 3: create the XSLT file called Xslt2Html.xsl and put it under src/main/resources/xslt directory.

Step 4: Keep everything inside the tag <xsl:stylesheet/>.

Step 5: Specify the namespace xmlns:xsl="http://www.w3.org/1999/XSL/Transform" for the XSLT.

Step 6: Write the Java class to transform the XML file data to HTML using XSLT file.

Step 7: Put both XML and XSLT files under classpath and finaly transforms the XML data into HTML output.

Step 8: Write the output to the HTML file called books.html under the project's root directory.

PROGRAM

If you are creating gradle based project in Eclipse then you can use the below build.gradle script:

```
plugins {
   id 'java-library'
}

sourceCompatibility = 12
targetCompatibility = 12

repositories {
   jcenter()
}

dependencies {
}
```

If you are creating maven based project then you can use below pom.xml file:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
      <modelVersion>4.0.0</modelVersion>
```

```
        <groupId>com.roytuts</groupId>
        <artifactId>java-xslt-xml-to-html</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <packaging>jar</packaging>

        <properties>
          <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
          <java.version>at least 1.8</java.version>
        </properties>

        <dependencies>
        </dependencies>
        <build>
          <plugins>
                <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-compiler-plugin</artifactId>
                        <version>3.8.1</version>
            <configuration>
                                <source>${java.version}</source>
                                <target>${java.version}</target>
                        </configuration>
                </plugin>
          </plugins>
        </build>
</project>
```

BOOKS.XML

```
<?xml version="1.0"?>
<catalog>
        <book id="bk101">
          <author>Gambardella, Matthew</author>
          <title>XML Developer's Guide</title>
          <genre>Computer</genre>
          <price>44.95</price>
          <publish_date>2000-10-01</publish_date>
          <description>An in-depth look at creating applications
                with XML.
          </description>
        </book>
        <book id="bk102">
          <author>Ralls, Kim</author>
          <title>Midnight Rain</title>
          <genre>Fantasy</genre>
          <price>5.95</price>
          <publish_date>2000-12-16</publish_date>
```

```
    <description>A former architect battles corporate zombies,
          an evil sorceress, and her own childhood to become queen
          of the world.
    </description>
  </book>
  <book id="bk103">
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-11-17</publish_date>
    <description>After the collapse of a nanotechnology
          society in England, the young survivors lay the
          foundation for a new society.
    </description>
  </book>
  <book id="bk104">
    <author>Corets, Eva</author>
    <title>Oberon's Legacy</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-03-10</publish_date>
    <description>In post-apocalypse England, the mysterious
          agent known only as Oberon helps to create a new life
          for the inhabitants of London. Sequel to Maeve
          Ascendant.
    </description>
  </book>
  <book id="bk105">
    <author>Corets, Eva</author>
    <title>The Sundered Grail</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-09-10</publish_date>
    <description>The two daughters of Maeve, half-sisters,
          battle one another for control of England. Sequel to
          Oberon's Legacy.
    </description>
  </book>
  <book id="bk106">
    <author>Randall, Cynthia</author>
    <title>Lover Birds</title>
    <genre>Romance</genre>
    <price>4.95</price>
    <publish_date>2000-09-02</publish_date>
    <description>When Carla meets Paul at an ornithology
          conference, tempers fly as feathers get ruffled.
    </description>
```

```
</book>

<book id="bk107">
   <author>Thurman, Paula</author>
   <title>Splish Splash</title>
   <genre>Romance</genre>
   <price>4.95</price>
   <publish_date>2000-11-02</publish_date>
   <description>A deep sea diver finds true love twenty
         thousand leagues beneath the sea.
   </description>
</book>
<book id="bk108">
   <author>Knorr, Stefan</author>
   <title>Creepy Crawlies</title>
   <genre>Horror</genre>
   <price>4.95</price>
   <publish_date>2000-12-06</publish_date>
   <description>An anthology of horror stories about roaches,
         centipedes, scorpions and other insects.
   </description>
</book>
<book id="bk109">
   <author>Kress, Peter</author>
   <title>Paradox Lost</title>
   <genre>Science Fiction</genre>
   <price>6.95</price>
   <publish_date>2000-11-02</publish_date>
   <description>After an inadvertant trip through a Heisenberg
         Uncertainty Device, James Salway discovers the problems
         of being quantum.
   </description>
</book>
<book id="bk110">
   <author>O'Brien, Tim</author>
   <title>Microsoft .NET: The Programming Bible</title>
   <genre>Computer</genre>
   <price>36.95</price>
   <publish_date>2000-12-09</publish_date>
   <description>Microsoft's .NET initiative is explored in
         detail in this deep programmer's reference.
   </description>
</book>
<book id="bk111">
   <author>O'Brien, Tim</author>
   <title>MSXML3: A Comprehensive Guide</title>
   <genre>Computer</genre>
   <price>36.95</price>
```

```
    <publish_date>2000-12-01</publish_date>
    <description>The Microsoft MSXML3 parser is covered in
        detail, with attention to XML DOM interfaces, XSLT processing,
        SAX and more.
    </description>
  </book>
  <book id="bk112">
    <author>Galos, Mike</author>
    <title>Visual Studio 7: A Comprehensive Guide</title>
    <genre>Computer</genre>
    <price>49.95</price>
    <publish_date>2001-04-16</publish_date>
    <description>Microsoft Visual Studio 7 is explored in depth,
        looking at how Visual Basic, Visual C++, C#, and ASP+ are
        integrated into a comprehensive development
        environment.
    </description>
  </book>
</catalog>
```

Xslt2Html.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/catalog">
    <html>
      <head>
        <style>
          table {
          font-family: arial, sans-serif;
          border-collapse: collapse;
          width:
          100%;
          }
          td, th {
          border: 1px solid #dddddd;
          text-align: left;
          padding: 8px;
          }
          tr:nth-child(even) {
          background-color: #dddddd;
          }
        </style>
      </head>
      <body>
        <h2>Books</h2>
```

```
                        <table>
                            <tr>
                            <th>Id</th>
                            <th>Author</th>
                            <th>Title</th>
                            <th>Genre</th>
                            <th>Price</th>
                            <th>Publish Date</th>
                            <th>Description</th>
                            </tr>
                                <xsl:for-each select="book">
                                    <tr>

                                    </tr>
  <td>

  </td>
  <td>

  </td>
  <td>

  </td>
  <td>

  </td>
  <td>

  </td>
  <td>

  </td>
  <td>
  </td>
  <xsl:value-of select="book/@id" />
<xsl:value-of select="author" />
<xsl:value-of select="title" />
<xsl:value-of select="genre" />
<xsl:value-of select="price" />
<xsl:value-of select="publish_date" />
<xsl:value-of select="description" />
                                    </xsl:for-each>
                            </table>
                    </body>
                </html>
            </xsl:template>
        </xsl:stylesheet>
```

```
package com.roytuts.java.xslt.xml.to.html;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter; import java.io.IOException;import
java.io.InputStream;import java.io.StringWriter;import java.net.URL;
import javax.xml.parsers.DocumentBuilderFactory; import
javax.xml.parsers.ParserConfigurationException;import
javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;


import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerFactoryConfigurationError;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;import
javax.xml.transform.stream.StreamSource;

import org.w3c.dom.Document; import org.xml.sax.SAXException;

public class XmlToHtmlTransformer {

      public static void main(String[] args) throws
            SAXException, IOException,
ParserConfigurationException,
             TransformerFactoryConfigurationError,
        TransformerException {transform("xml/books.xml",
        "xslt/Xslt2Html.xsl");
      }
      public static void transform(final String xml, final String xslt)
throws SAXException,IOException,
             ParserConfigurationException,
                                                TransformerFact
oryConfigurationError,TransformerException {

        ClassLoader classloader =
        XmlToHtmlTransformer.class.getClassLoader();InputStream
        xmlData = classloader.getResourceAsStream(xml);
        URL xsltURL = classloader.getResource(xslt);

        DocumentxmlDocument=
DocumentBuilderFactory.newInstance().newDocumentBuilder().parse(
xmlData);
        Source stylesource = new
                StreamSource(xsltURL.openStream(),
xsltURL.toExternalForm());
        Transformer transformer=
TransformerFactory.newInstance().newTransformer(stylesource);
```

```
        StringWriter stringWriter = new StringWriter();
        transformer.transform(new
                                                DOMSource(xmlDocu
ment),                                          new
StreamResult(stringWriter));

        // write to file
        File file = new File("books.html");if (!file.exists()) {
                file.createNewFile();
        }

        FileWriter fw = new FileWriter(file); BufferedWriter bw = new
        BufferedWriter(fw);bw.write(stringWriter.toString());
        bw.close();
      }

}
```

OUTPUT

Books

Id	Author	Title	Genre	Price	Publish Date	Description
	Gambardella, Matthew	XML Developer's Guide	Computer	44.95	2000-10-01	An in-depth look at creating applications with XML.
	Ralls, Kim	Midnight Rain	Fantasy	5.95	2000-12-16	A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.
	Corets, Eva	Maeve Ascendant	Fantasy	5.95	2000-11-17	After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.
	Corets, Eva	Oberon's Legacy	Fantasy	5.95	2001-03-10	In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.
	Corets, Eva	The Sundered Grail	Fantasy	5.95	2001-09-10	The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.
	Randall, Cynthia	Lover Birds	Romance	4.95	2000-09-02	When Carla meets Paul at an ornithology conference, tempers fly as feathers get ruffled.
	Thurman, Paula	Splish Splash	Romance	4.95	2000-11-02	A deep sea diver finds true love twenty thousand leagues beneath the sea.
	Knorr, Stefan	Creepy Crawlies	Horror	4.95	2000-12-06	An anthology of horror stories about roaches, centipedes, scorpions and other insects.
			Science		2000-11-	After an inadvertant trip through a Heisenberg Uncertainty Device, James Salway discovers the

RESULT

Thus an XML document is converted to a HTML table using XSLT code.

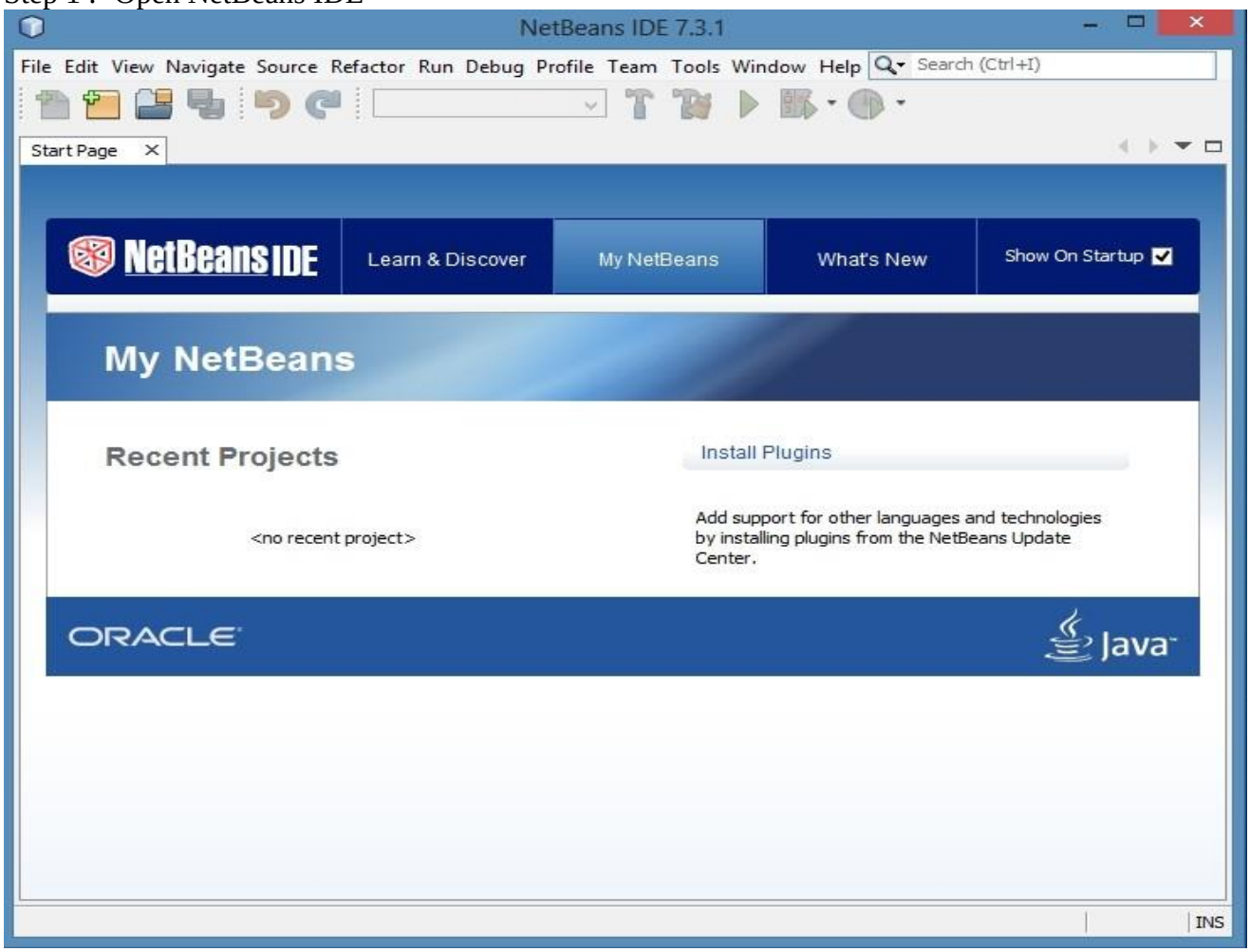
Ex. No. 5 a)	MODULE V WEB SERVICES
Date:	**Create a SOAP based web service for a simple Java Calculator class with operations add and subtract then create a web service client which then consumes the web service and displays the result of the invoked web service.**

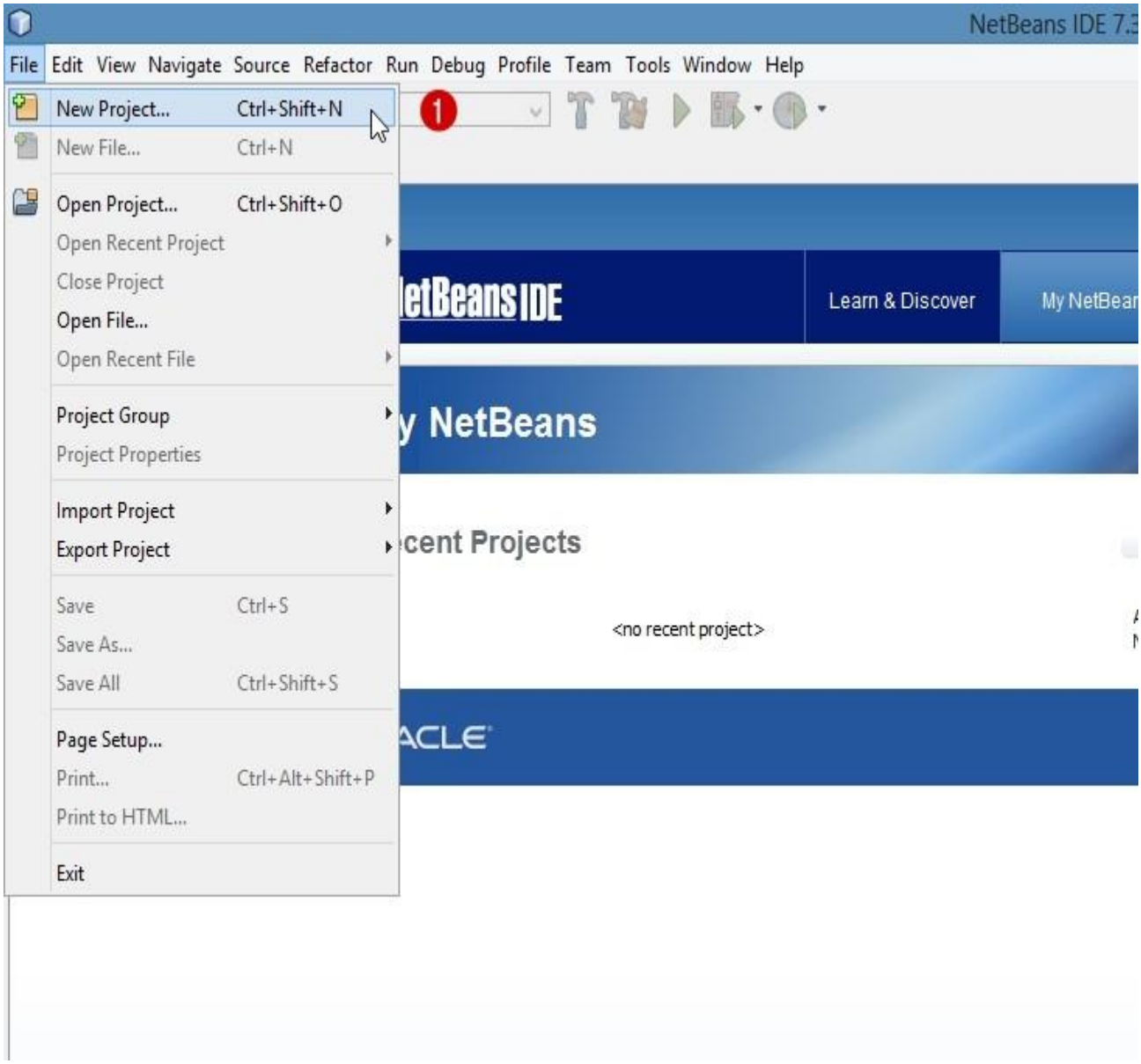
AIM

To Create a SOAP based web service for a simple Java Calculator class with operations add and subtract then create a web service client which then consumes the web service and displays the result of the invoked web service.

ALGORITHM

Step 1 :- Open NetBeans IDE

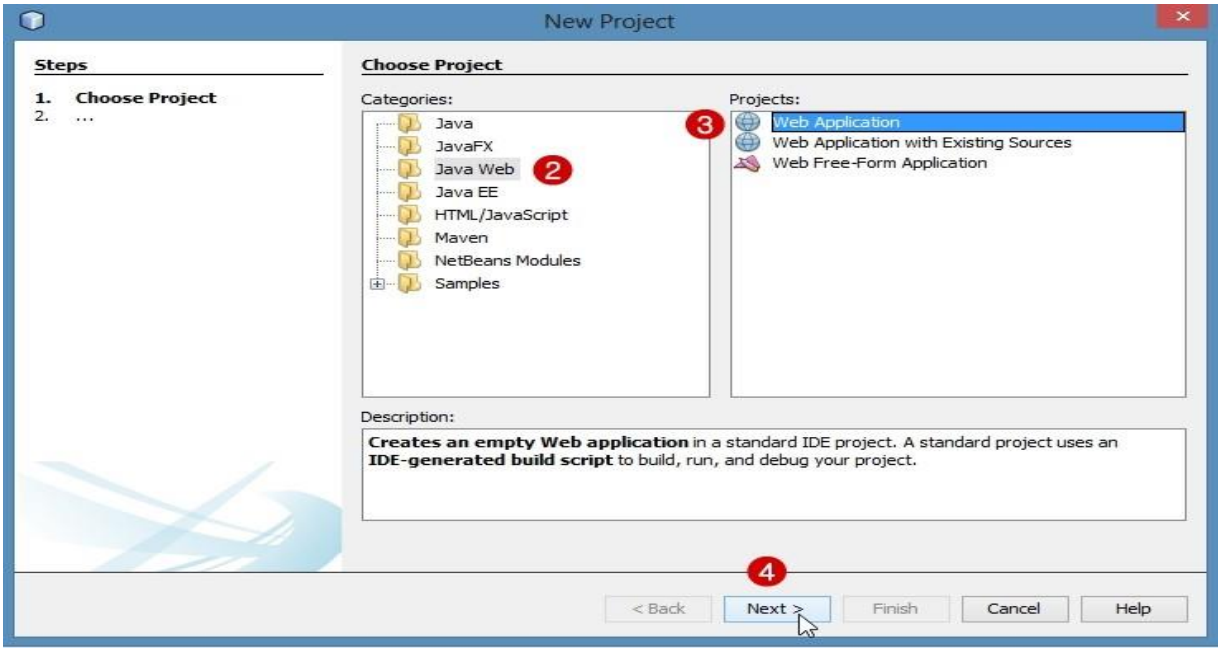




Step 2 :- Under Categories: select Java Web.

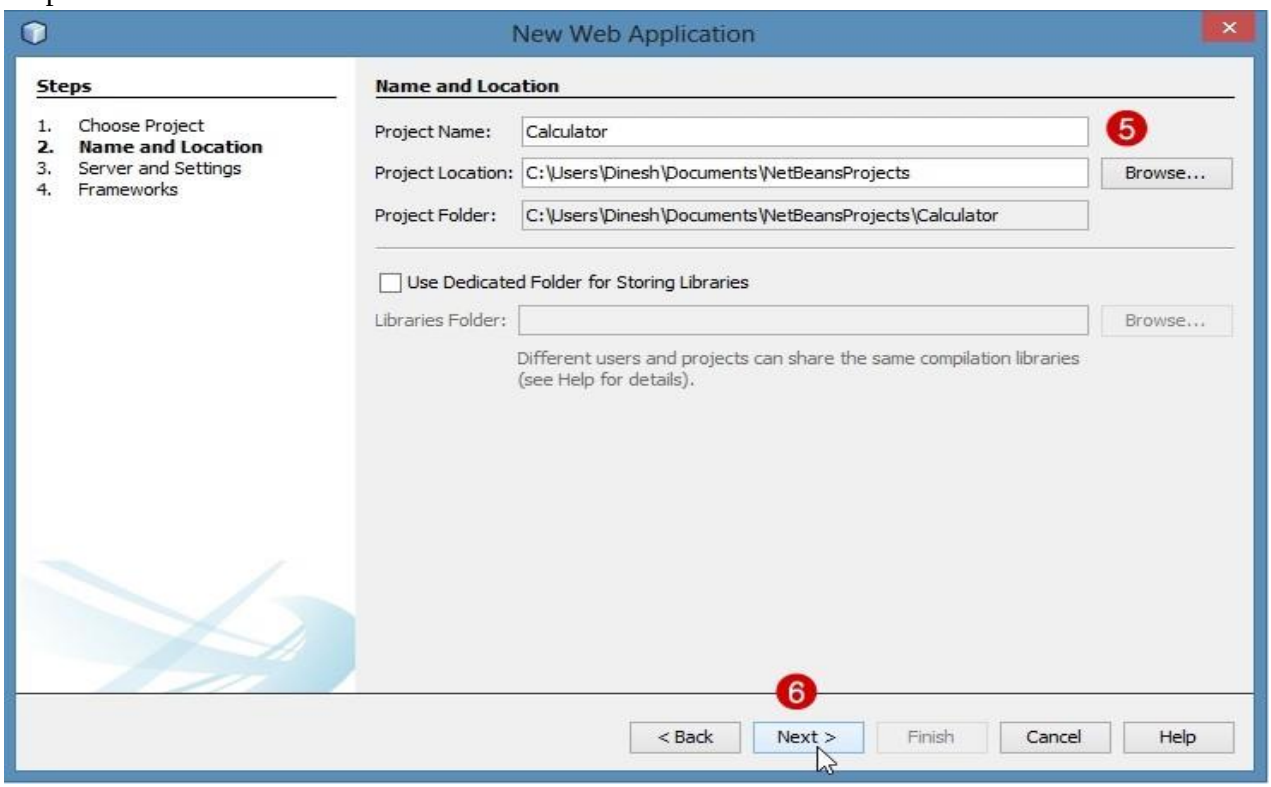
Step 3 :- Under Projects: select Web Application.

Step 4 :- Click Next >



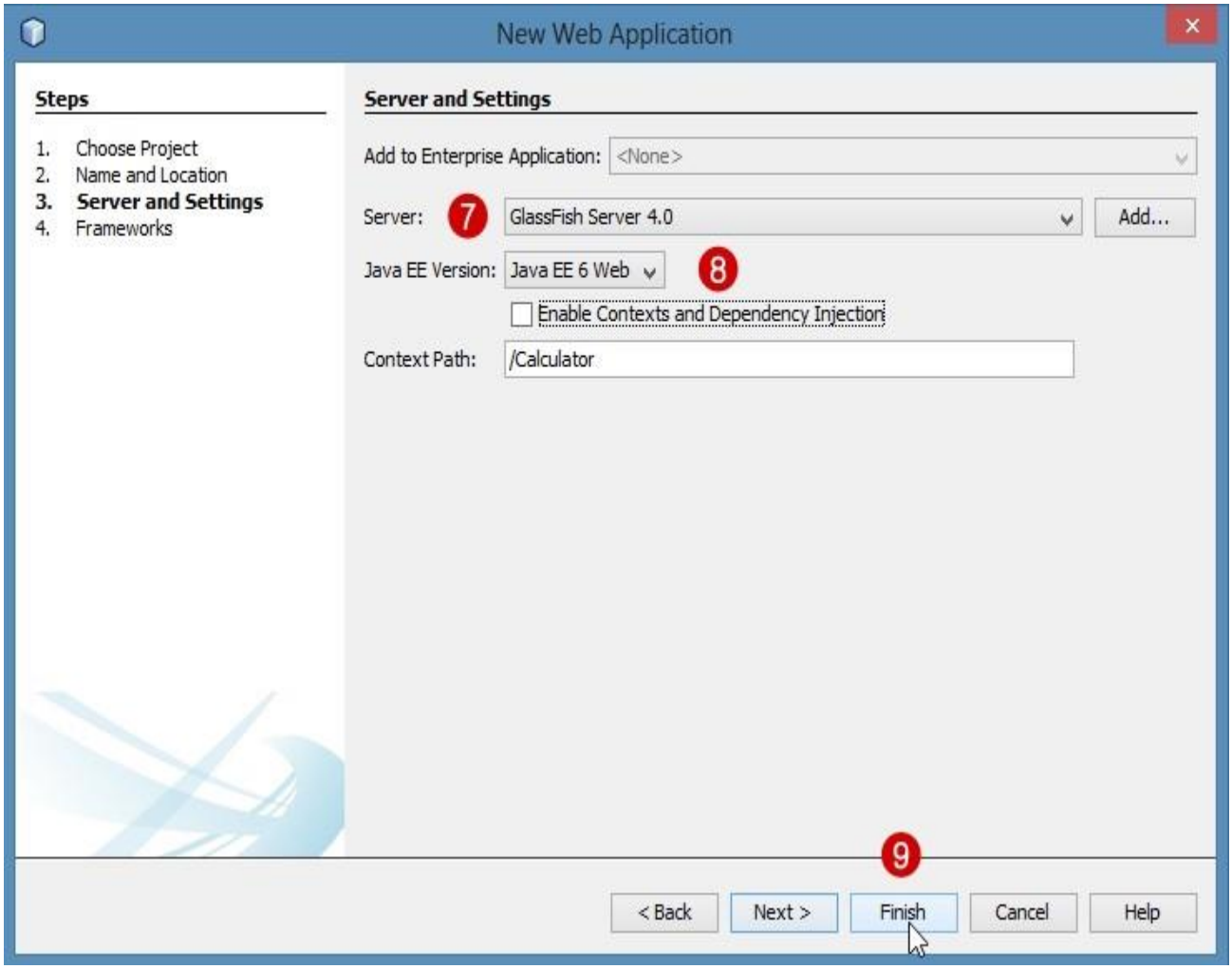
Step 5 :- Under Name and Location tab, enter Project Name: as "Calculator".

Step 6 :- Click Next >



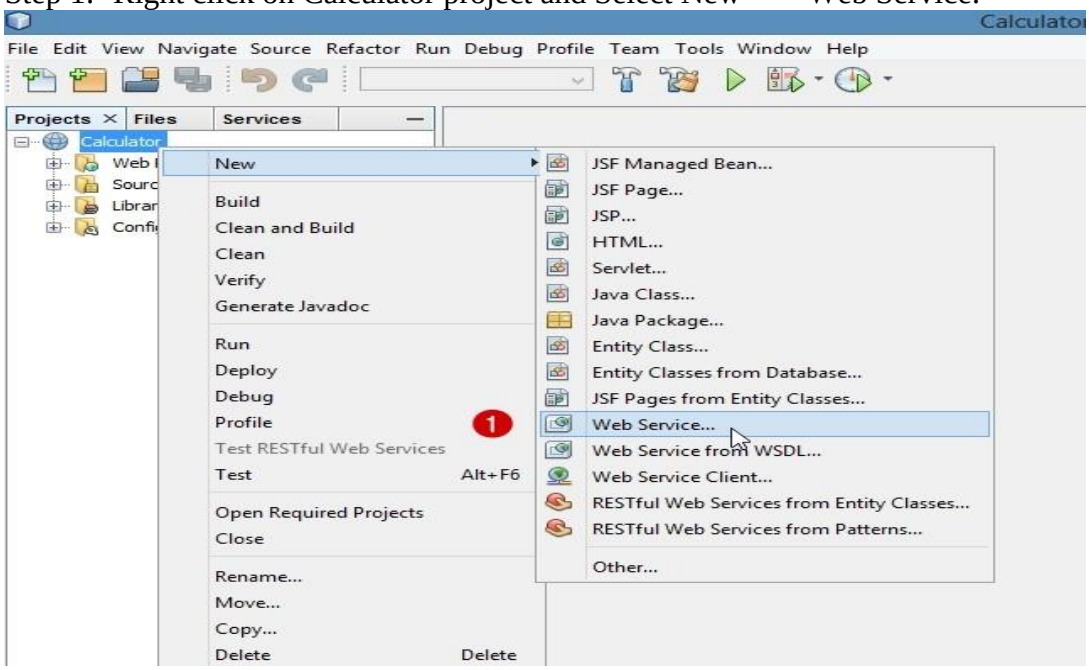
Step 7:- Choose Server: as "GlassFish Server 4.0". You can also choose "GlassFish v3 Domain" as server if you are using old version of NetBeans other than 7.3.1, which comes bundled with "GlassFish v3 Domain".

Step 8:- Choose Java EE 6 Web as Java EE Version:. Keep rest as default. Step 9:- Click Finish.


New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Server and Settings
Add to Enterprise Application: <None>
Server: 7 GlassFish Server 4.0
Add...
Java EE Version: Java EE 6 Web 8
Enable Contexts and Dependency Injection
Context Path: /Calculator
9
< Back
Next >
Finish
Cancel
Help

Creating a SOAP Web Service called as "CalculatorService"

Step 1:- Right click on Calculator project and Select New ---> Web Service.

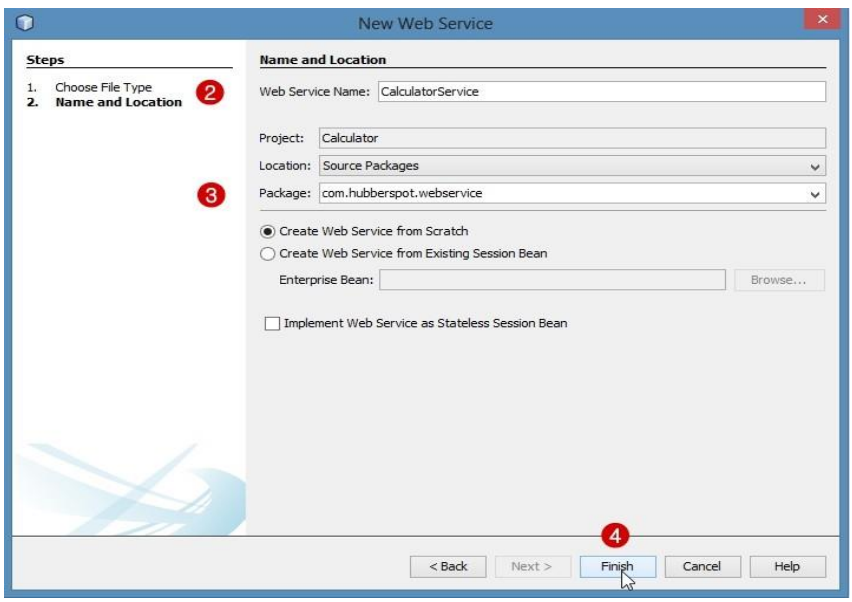


Step 2:- New Web Service dialog box gets open. In the Web Service Name:

textfield enter name as "CalculatorService".

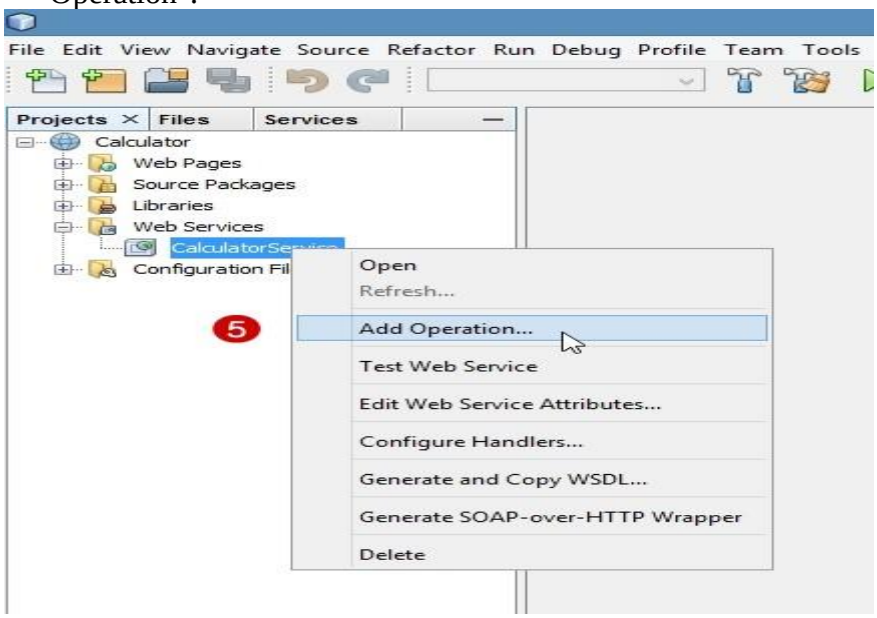
Step 3:- Enter the package name for the CalculatorService Web Service.

Step 4:- Click Finish



Creating a simple operation called as "sum".

Step 5:- After creating Web Service by name "CalculatorService". Under Web Services directory of the project, right click on the CalculatorService created and click on "Add Operation".

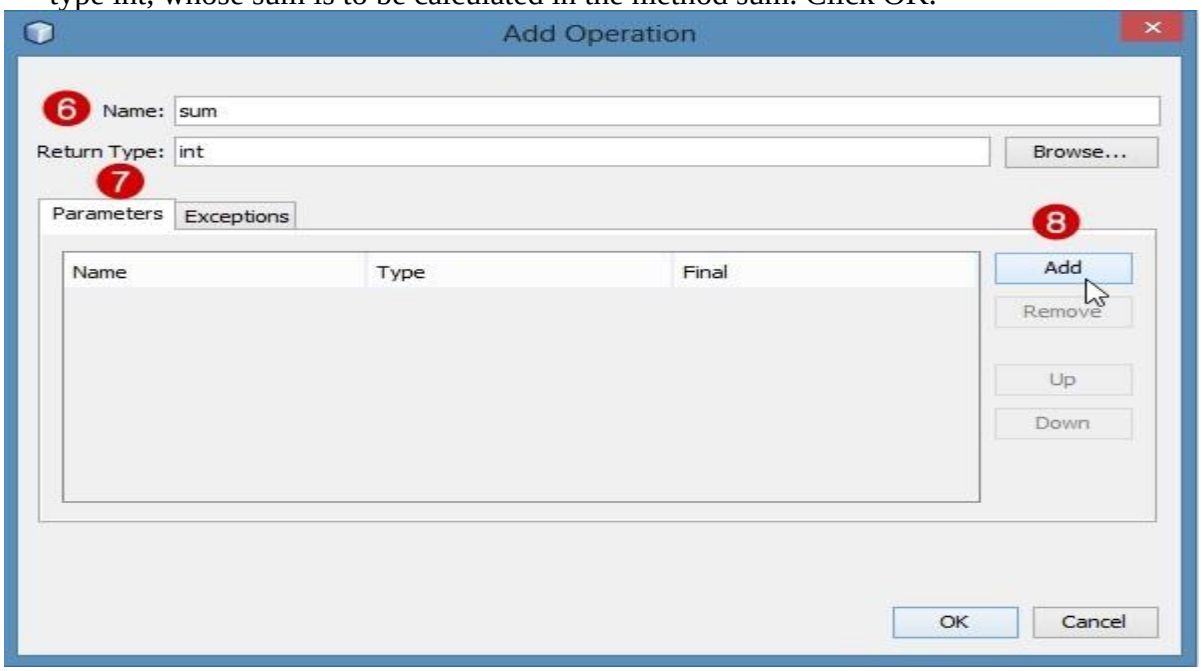


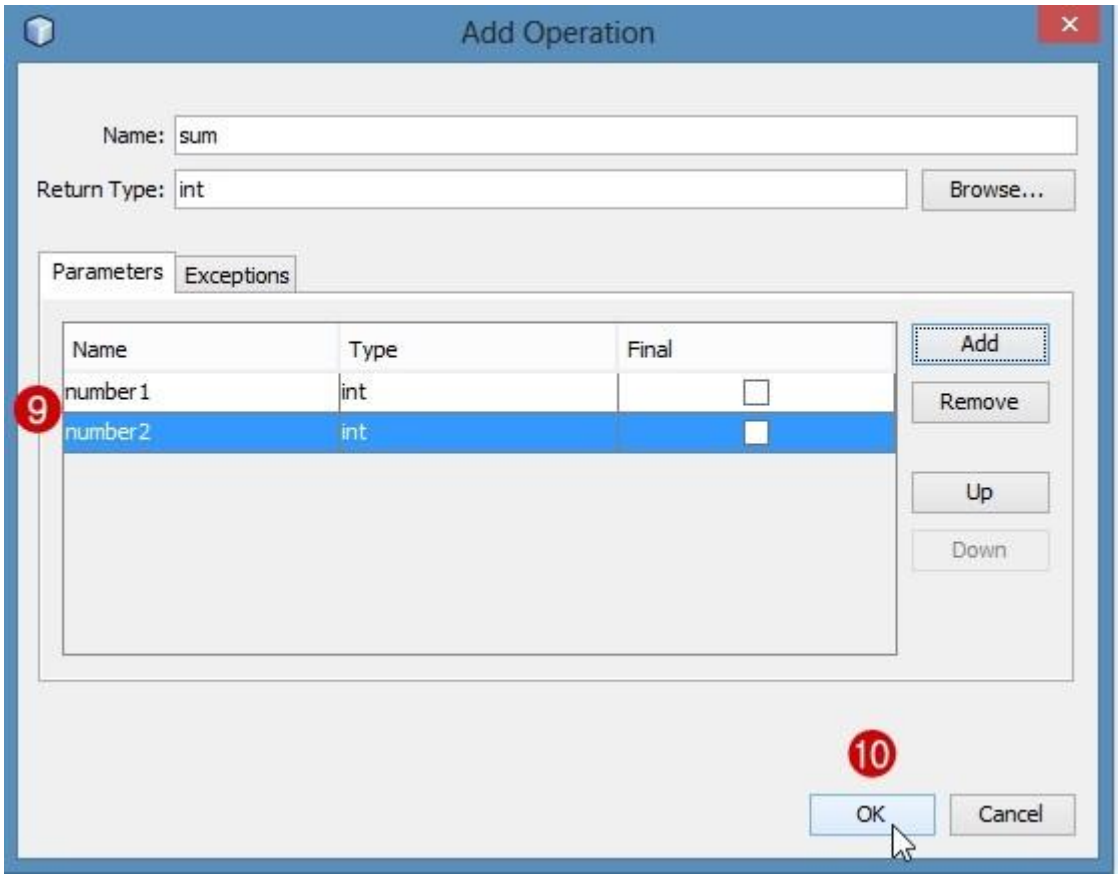
Add Operation dialog box gets open.

Step 6:- Enter name of the operation to expose as Web Service method. Here provideas "sum".

Step 7:- Enter the return type for the method. Here sum method will calculate sum oftwo numbers "number1" and "number2" and return it as a int value.

Step 8, 9 and 10:- Enter two parameters by clicking add button as "number1" and"number2" of type int, whose sum is to be calculated in the method sum. Click OK.



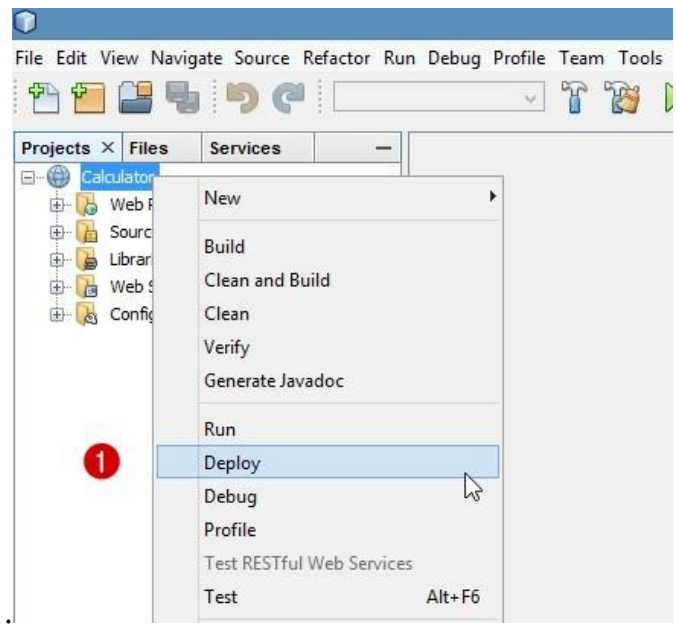

Add Operation
Name: sum
Return Type: int
Browse...
Parameters
Exceptions
Name
Type
Final
number1
int
number2
int
Add
Remove
Up
Down
9
10
OK
Cancel

PROGRAM

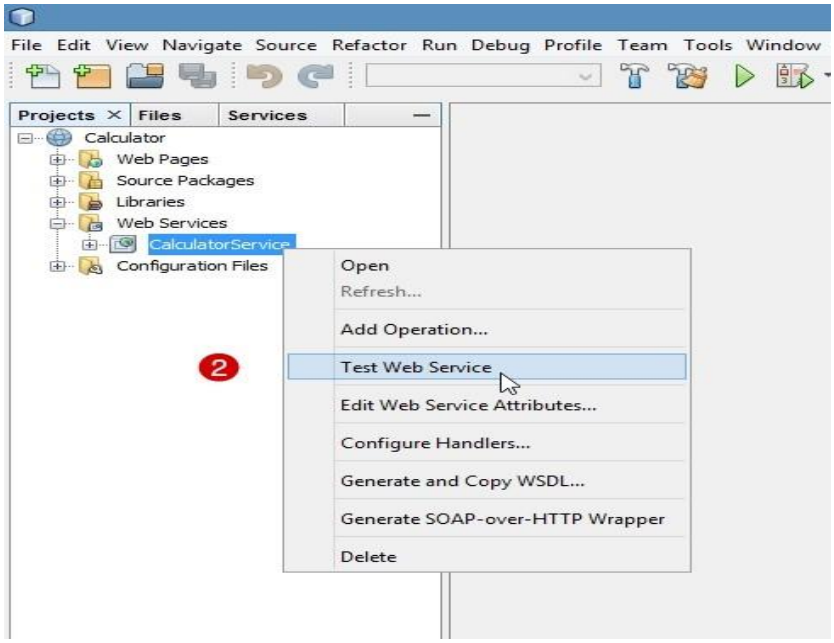
```
package com.hubberspot.webservice;
import  javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

// @WebService annotation makes a class a Web Service
@WebService(serviceName = "CalculatorService")
public class CalculatorService {

   // @WebMethod annotation expose a method as a service.
   @WebMethod(operationName = "sum")
   // @WebParam annotation indicates parameters to method coming from SOAP request.
   public int sum(@WebParam(name = "number1") int number1,
                  @WebParam(name = "number2") int number2) {

      int sum = 0;

      sum = number1 + number2;

      return sum;
   }
}
```

Deploy and Test the Web Service.

Step 1:- Right click on the "Calculator" project directory and click "Deploy". The Web Service gets deployed on the GlassFish server.



Step 2:- Under Web Services directory of the project, right click on the CalculatorService created and click on "Test Web Service". It opens a browser window to test the CalculatorService Web Service.



Step 3:- In order to test sum method exposed as Web Service. Enter number1 parameter value on the Tester client say 5.

Step 4:- Enter number2 parameter value on the Tester client say 10.

Step 5:- After the entering the value for number1 and number2 variables click sum button.

RESULT

Thus a SOAP based web service has been created for a simple Java Calculator class with operations add and subtract, then create a web service client which then consumes the web service and displays the result of the invoked web service.

Ex. No. 5 b)	MODULE V WEB SERVICES
Date:	**Write a web service for finding what people think by asking 500 people's opinion for any consumer product.**

AIM:

To write a web services for finding what people think by asking 500people's opinion for any consumer product

PROCEDURE:

Open the home page.

- Enter the login ID and type the comments then submit.
- Retrieve comments with post id
- Display the comments.

PROGRAM:

Index.php

```
<!doctype html>
<html lang="en">
<head>
        <meta charset="UTF-8" />
        <title>jQuery Ajax Comment System - Demo</title>
        <link rel="stylesheet" href="css/style.css">
        <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
        <script src="js/script.js"></script>
</head>
<body>
        <div class="wrap">
                <h1> Maggy Noodles Comment System</h1>
        <?php
                // retrive post
                include('config.php');
                include ('function.php');
                dbConnect();

                $query = mysql_query(
                        'SELECT *
                        FROM post
                          WHERE post_id = 1');
        ?>          $row = mysql_fetch_array($query);

                <div class="post">
                        <h2><?php echo $row['post_title']?></h2>
```

```
                <p><?php echo $row['post_body']?></p>
            </div>

        <?php
            // retrive comments with post id
            $comment_query = mysql_query(
                "SELECT *
                FROM comment
                WHERE post_id = {$row['post_id']}
                ORDER BY comment_id DESC LIMIT
                15");
        ?>

            <h2>Comments.....</h2>
            <div class="comment-block">
            <?php while($comment = mysql_fetch_array($comment_query)):
?>
                <div class="comment-item">
                    <div class="comment-avatar">
                        <img src="<?php echo
avatar($comment['mail']) ?>" alt="avatar">
                    </div>
                    <div class="comment-post">
                        <h3><?php echo $comment['name'] ?>
<span>said. .....</span></h3>
                        <p><?php echo $comment['comment']?></p>
                    </div>
                </div>
            <?php endwhile?>
            </div>

            <h2>Submit new comment</h2>
            <!--comment form -->
            <form id="form" method="post">
                <!-- need to supply post id with hidden fild -->
                <input type="hidden" name="postid" value="<?php echo
$row['post_id']?>">
                <label>
                    <span>Name *</span>
                    <input type="text" name="name" id="comment-name"
placeholder="Your name here. ....." required>
                </label>
                <label>
```

```
                                <span>Email *</span>
                                <input type="email" name="mail" id="comment-mail"
placeholder="Your mail here....." required>
                        </label>
                        <label>
                                <span>Your comment *</span>
                                <textarea name="comment" id="comment" cols="30"
rows="10" placeholder="Type your comment here. ....." required></textarea>
                        </label>
                        <input type="submit" id="submit" value="Submit
Comment">
                </form>
        </div>
</body>
</html>
```

Ajax_Comment.php

```
<?php
if (isset( $_SERVER['HTTP_X_REQUESTED_WITH'] )):
        include('config.php');
        include('function.php');
        dbConnect();

        if (!empty($_POST['name']) AND !empty($_POST['mail']) AND
!empty($_POST['comment']) AND !empty($_POST['postid'])) {
                $name = mysql_real_escape_string($_POST['name']);
                $mail = mysql_real_escape_string($_POST['mail']);
                $comment = mysql_real_escape_string($_POST['comment']);
                $postId = mysql_real_escape_string($_POST['postid']);

                mysql_query("
                        INSERT INTO comment (name,
                        mail, comment, post_id)
                        VALUES('{$name}', '{$mail}', '{$comment}',
 '{$postId}')");
        }
?>

<div class="comment-item">
        <div class="comment-avatar">
                <img src="<?php echo avatar($mail) ?>" alt="avatar">
        </div>
        <div class="comment-post">
```

```
                <h3><?php echo $name ?> <span>said. .....</span></h3>
                <p><?php echo $comment?></p>
        </div>
</div>

<?php
        dbConnect(0);
endif?>
```

Config.php

```
<?php
# db configuration
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', 'root');
define('DB_NAME', 'dbname');
?>
```

Function.php

```
<?php
/**
 * Connect to mysql server
 * @param bool
 * @use true to connect false to close
 */
function dbConnect($close=true){

        if (!$close) {
                mysql_close($link);
                return true;
        }

        $link = mysql_connect(DB_HOST, DB_USER, DB_PASS) or die('Couldnot connect
to MySQL DB ') . mysql_error();
        if (!mysql_select_db(DB_NAME, $link))return
                false;
}

/**
 * gravatar Image
 * @see http://en.gravatar.com/site/implement/images/
 */
function avatar($mail, $size = 60){
```

```
        $url = "http://www.gravatar.com/avatar/";
        $url .= md5( strtolower( trim( $mail ) ) );
        // $url .= "?d=" . urlencode( $default );
        $url .= "&s=" . $size;
        return $url;
}
?>
```

Style.CSS

```
/* general styling */
*{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        -webkit-box-sizing: border-box;
        -moz-box-sizing: border-box;
        -webkit-font-smoothing: antialiased;
        -moz-font-smoothing: antialiased;
        -o-font-smoothing: antialiased;font-
        smoothing: antialiased;
}       text-rendering: optimizeLegibility;
body{

        font: 12px Arial,Tahoma,Helvetica,FreeSans,sans-serif;text-
        transform: inherit;
        color: #333; background:
        #e7edee;width: 100%;
        text-shadow: 0 1px 1px rgba(0, 0, 0, 0.2)
}
.wrap{
        width: 720px; margin:
        15px auto; padding:
        15px 20px;background:
        white;
        border: 2px solid #DBDBDB;
        -webkit-border-radius: 5px;
        -moz-border-radius: 5px;
        border-radius: 5px;
        overflow: hidden;
}

a{ text-decoration: none; color: #333}
```

```
h1{
	font-family: Georgia, "Times New Roman", Times, serif;font-size:
	2.8em;
	text-align: center;
	margin: 25px 0;
}
h2{font-size: 1.5em; margin: 8px 0}h3{
	font-size: 1.2em;
	margin: 5px 0;
}
h3 span{
	font-weight: normal;
	font-size: 1em;
}
.item{
	clear: both;
	margin:0; padding:
	10px; overflow:
	hidden;
	border-top: 1px solid #DBDBDB;
}
```

```
.item:last-child{border-bottom:1px solid #DBDBDB}
.item:hover{background: #f9f9f9}
.post{
	padding: 10px 0;
	border-bottom: 1px solid #E6E6E6;
}
.comment-block{
	margin: 20px 0 20px 20px;
}
.comment-item{
	overflow: hidden;
	width: 500px; clear:
	both; padding: 10px;
	border: 1px solid #E6E6E6;
	border-radius: 5px;
	margin: 5px;
}
.comment-avatar{
	width: 60px;
	float: left;
}
.comment-avatar img{ width:
	60px; height: 60px;
	border-radius: 5px;
}
.comment-post{
	width: 400px;
	float: left;
	padding: 0 5px 0 10px;
}
#form{
	clear: both;
	margin: 10px;
	width: 500px;
}

/* form styling */
input[type="text"],
input[type="email"],
input[type="tel"],
input[type="url"],
textarea {
	width:100%; background:
	#fff; border: 1px solid
	#ddd;font-size: 13px;
	line-height: 20px;
	margin:  0; padding:
	7px 10px;
	box-shadow: inset 0 1px 2px #eee;
	border:1px solid #CCC;
	margin:0 0 5px;
```

```
        border-radius:5px;
}

textarea {
        height:100px; max-
        width:100%;
}
input[type="submit"] {
        cursor:pointer;
        width:100%;
```

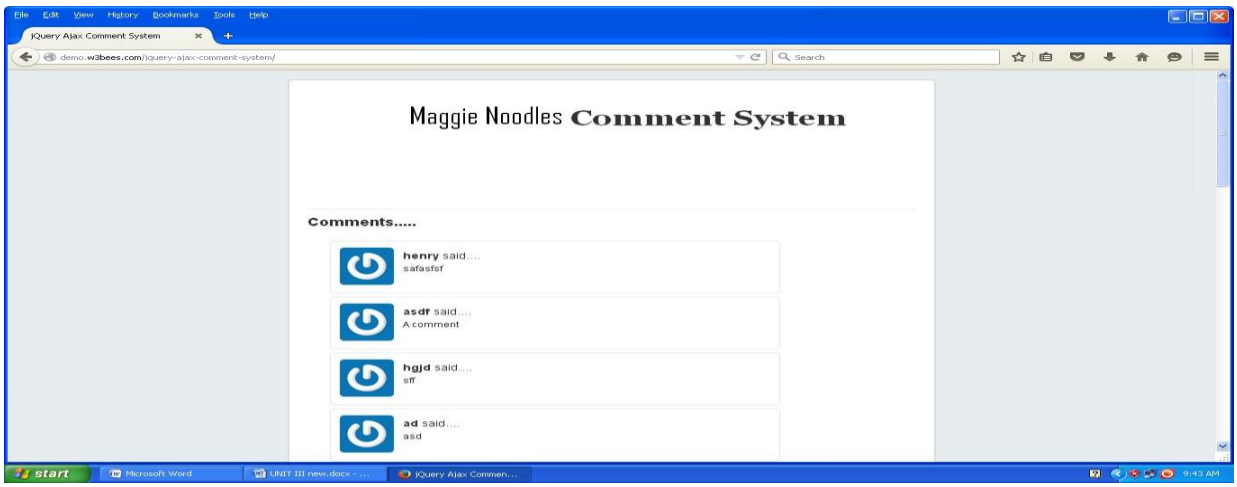
```
	border:none;
	background:#991D57;
	background-image:linear-gradient(bottom, #8C1C50 0%, #991D57 52%);
	background-image:-moz-linear-gradient(bottom, #8C1C50 0%, #991D57
52%);
	background-image:-webkit-linear-gradient(bottom, #8C1C50 0%,
#991D57 52%);
	color:#FFF; margin:0
	0 5px; padding:10px;
	border-radius:5px;
}
input[type="submit"]:hover {
	background-image:linear-gradient(bottom, #9C215A 0%, #A82767
52%);
	background-image:-moz-linear-gradient(bottom, #9C215A 0%, #A82767
52%);
	background-image:-webkit-linear-gradient(bottom, #9C215A 0%,
#A82767 52%);
	-webkit-transition:background 0.3s ease-in-out;
	-moz-transition:background 0.3s ease-in-out;
	transition:background-color 0.3s ease-in-out;
}
input[type="submit"]:active {
	box-shadow:inset 0 1px 3px rgba(0,0,0,0.5);
}
input:focus,
textarea:focus {
	outline:0;
	border:1px solid #999;
}
label{
	display: block;
	margin: 5px 0;
	font-weight: 900;
	cursor: pointer;
}

.alert{
	display: none;
	padding: 8px 35px 8px 14px;
	margin: 20px 0;
	text-shadow: 0 1px 0 rgba(255, 255, 255, 0.5);
```

```
        color: #468847; background-
        color: #dff0d8;border-color:
        #d6e9c6;
        -webkit-border-radius: 4px;
        -moz-border-radius: 4px;
        border-radius: 4px;
}
```

Script.js

```
$(document).ready(function(){var
        form = $('form');
        var submit = $('#submit');

        form.on('submit', function(e) {
                // prevent default action
                e.preventDefault();
                // send ajax request
                $.ajax({
                        url: 'ajax_comment.php',
                        type: 'POST',
                        cache: false,
                        data: form.serialize(), //form serizlize data
                        beforeSend: function(){
                                // change submit button value text and disabled it
                                submit.val('Submitting...').attr('disabled', 'disabled');
                        },
                        success: function(data){
                                // Append with fadeIn see
http://stackoverflow.com/a/978731
                                var item = $(data).hide().fadeIn(800);
                                $('.comment-block').append(item);


                                // reset form and button
                                form.trigger('reset');
                                submit.val('Submit Comment').removeAttr('disabled');
                        },
                        error: function(e){
                                alert(e);
                        }
                });
        });
});
```

OUTPUT



RESULT:

Thus a web services for finding what people think by asking 500 people'sopinion for any consumer product has been executed successfully.

Ex. No. 11	CONTENT BEYOND SYLLABUS
Date:	DEVELOPMENT OF WEB APPLICATION OF ANGULARJS

AIM

To develop a web application using AngularJS framework.

DESCRIPTION

AngularJS is a client side JavaScript MVC framework to develop a dynamic web application. AngularJS was originally started as a project in Google but now, it is open source framework.

AngularJS Extends HTML

- AngularJS extends HTML with ng-directives.
- The ng-app directive defines an AngularJS application.
- The ng-model directive binds the value of HTML controls (input, select, textarea) to application data.
- The ng-bind directive binds application data to the HTML view.

SETUP ANGULARJS DEVELOPMENT ENVIRONMENT

We need the following tools to setup a development environment for AngularJS:

1. AngularJS Library
2. Editor/IDE
3. Browser
4. Web server

AngularJS Library

- To download AngularJS library, go to angularjs.org -> click download button, which will open the following popup.

- Select the required version from the popup and click on download button in the popup.

CDN: You can include AngularJS library from CDN url –
https://ajax.googleapis.com/ajax/libs/angularjs/1.3.16/angular.min.js

Editor

- AngularJS is eventually HTML and JavaScript code. So you can install any good editor/IDE as per your choice.

The following editors are recommended:

- Sublime Text
- Aptana Studio 3
- Ultra Edit
- Eclipse
- Visual Studio

Web server

- Use any web server such as IIS, apache etc., locally for development purpose.

Browser

- You can install any browser of your choice as AngularJS supports cross-browser compatibility. However, it is recommended to use Google Chrome while developing an application.

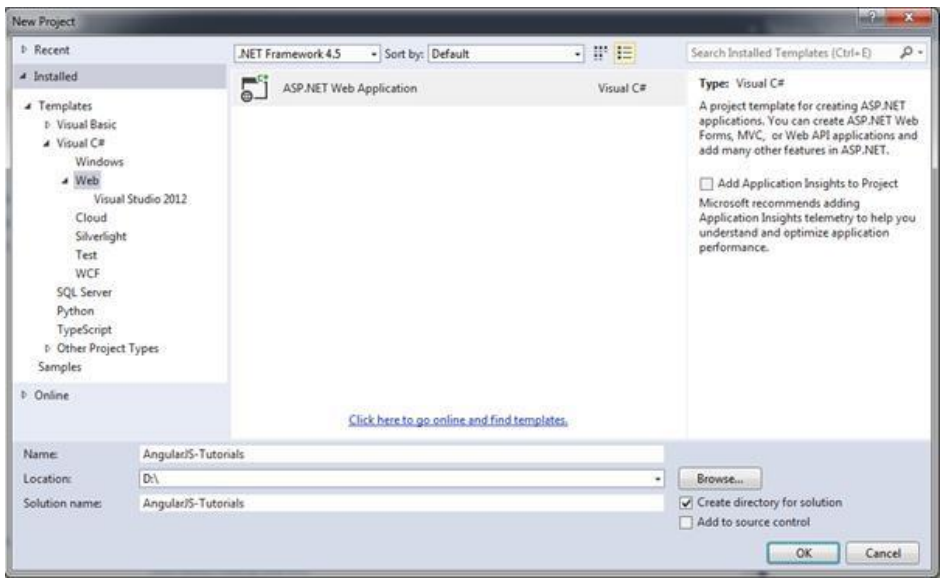
Angular Seed

- Use Angular seed project to quickly get started on AngularJS application. The Angular-seed is an application skeleton for a typical AngularJS web application. You can use it to quickly bootstrap your angular webapp projects and development environment for your project.

Let's setup Angular project in Visual Studio 2013 for web.

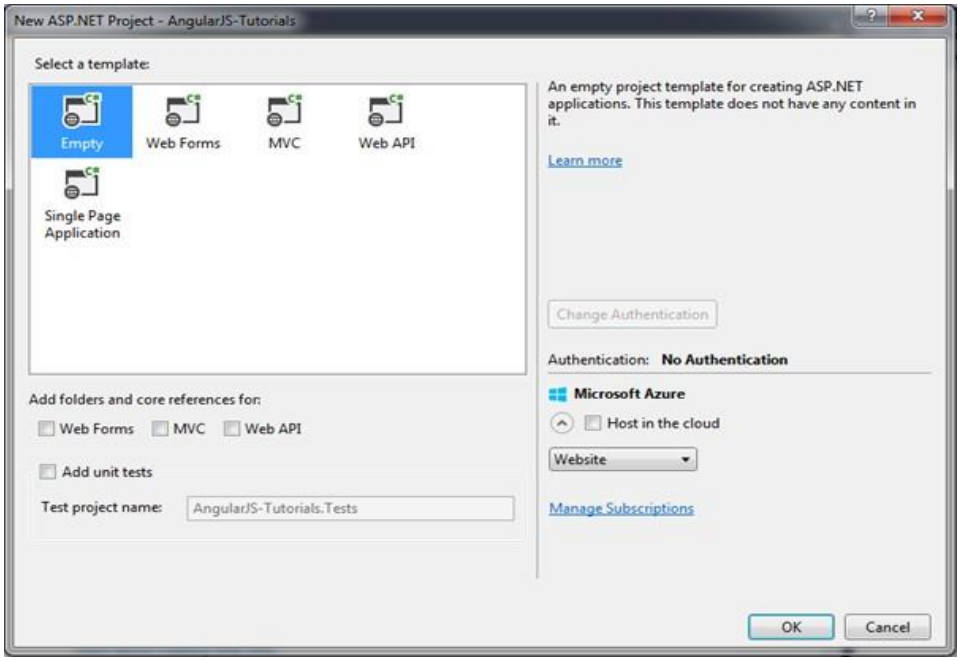
Setup AngularJS Project in Visual Studio

- You can create AngularJS application in any version of Visual Studio. Here, we will use Visual Studio 2013 for web.
- First, create new project by clicking on New Project link on start page. This will open New Project dialog box, as shown below.

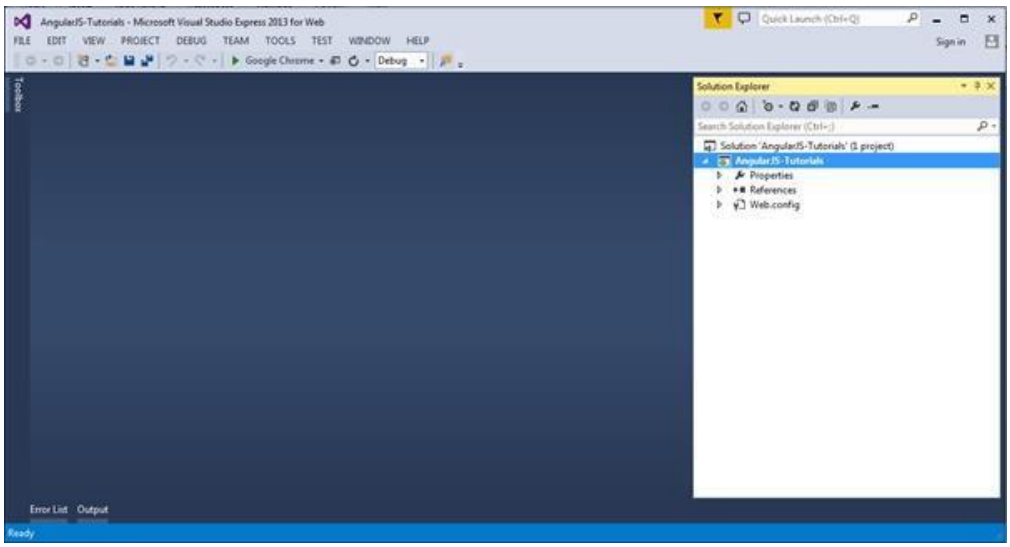


AngularJS in Visual Studio

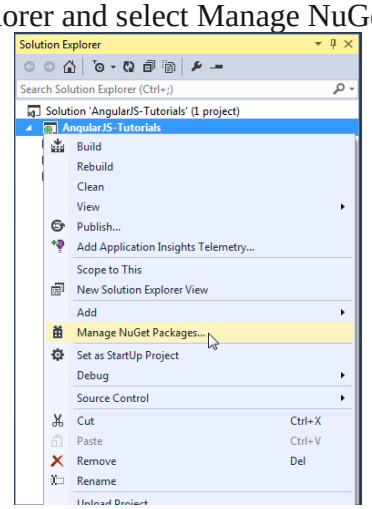
- Select Web in the left pane and ASP.NET Web Application in the middle pane and then click OK.
- In the New ASP.NET Project dialog box, select Empty template and then click OK.



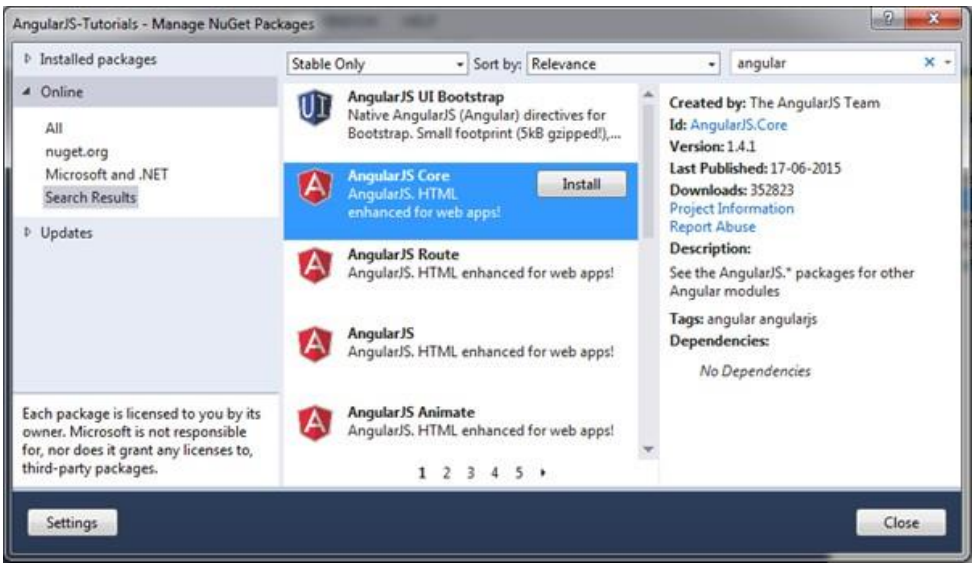
- This will create an empty website project in Visual Studio.



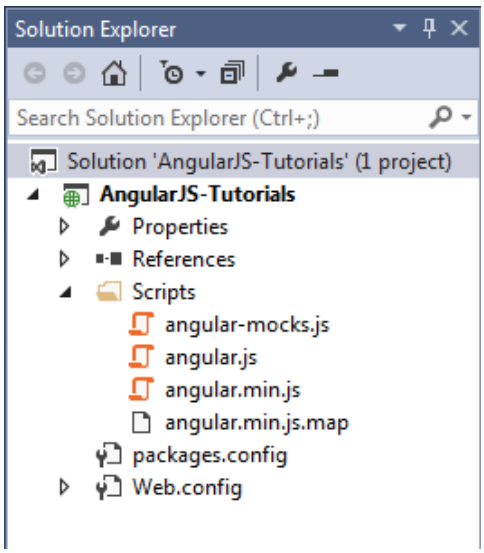
- Now, install AngularJS library from NuGet package manager. Right click on the project in Solution Explorer and select Manage NuGet Packages.



- Search for "angular" in the Manage NuGet Packages dialog box and install AngularJS Core.



- This will add AngularJS files into Scripts folder such as angular.js, angular.min.js, and angular-mocks.js, as shown below.



CODE

Let's create a simple AngularJS web application step by step and understand the basic building blocks of AngularJS.

1. First, create an HTML document with <head> and <body> elements, as show below.

Example: HTML Template

```
<!DOCTYPE html>
<html>
<head>
</head>
```

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

2. Include angular.js file in the head section (you have learned how to download angular library in the previous section). You can take a reference from the CDN also. (all the examples in this tutorials will use CDN reference.)

Example: Include AngularJS Library

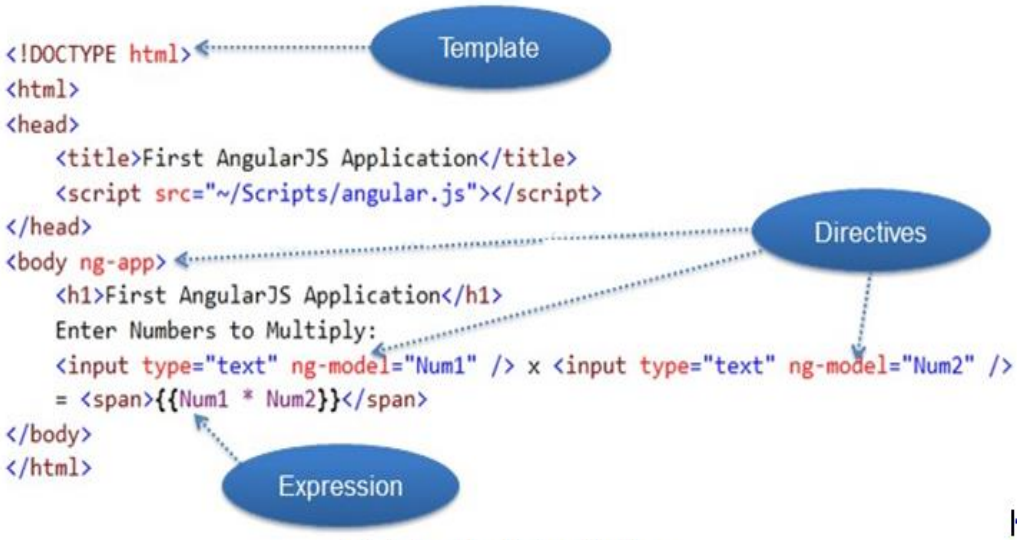
```
<!DOCTYPE html>
<html>
<head>
<title>First AngularJS Application</title>
<script src= "~/Scripts/angular.js"></script>
</head>
<body>
</body>
</html>
```

3. Here, we will be creating a simple multiplier application which will multiply two numbers and display the result. User will enter two numbers in two separate textboxes and the result will be displayed immediately, as shown below.

- The following is the HTML code with AngularJS for the above multiplier example. Example: First AngularJS Application.

```
<!DOCTYPE html>
<html>
<head>
<title>First AngularJS Application</title>
<script src= "~/Scripts/angular.js"></script>
</head>
<body ng-app >
<h1>First AngularJS Application</h1>
Enter Numbers to Multiply:
<input type="text" ng-model="Num1" /> x <input type="text" ng-model="Num2" />
 <span>{{Num1 * Num2}}</span>
</body>
</html>
```

- The above example is looks like HTML code with some strange attributes and braces such as ng-app, ng-model, and {{ }}. These built-in attributes in AngularJS are called directives.
- The following figure illustrates the AngularJS building blocks in the above example



First AngularJS Application

Template

- In AngularJS, a template is HTML with additional markups. AngularJS compiles templates and renders the resultant HTML.

Directive

- Directives are markers (attributes) on a DOM element that tell AngularJS to attach a specific behavior to that DOM element or even transform the DOM element and its children.
- Most of the directives in AngularJS are starting with ng. It stands for Angular. We have applied ng-app and ng-model directive in the above example.
- ng-app: The ng-app directive is a starting point. If AngularJS framework finds ng-app directive anywhere in the HTML document then it bootstraps (initializes) itself and compiles the HTML template.
- ng-model: The ng-model directive binds HTML element to a property on the $scope object. You will learn about this model later but for now let us consider this as a model property.
- In the above example, we have included ng-model directive to both the textboxes with different names Num1 and Num2. AngularJS framework will create two properties called Num1 and Num2 in the scope and will assign a value that we type into textboxes.

Expression

- An expression is like JavaScript code which is usually wrapped inside double curly braces such as {{ expression }}. AngularJS framework evaluates the expression and produces a result. In the above example, {{ Num1 * Num2}} will simply display the product of Num1 and Num2.
- The following table lists all the important concepts in AngularJS.

Concept	Description
Template	HTML with additional markup
Directives	Extends the HTML with custom attributes and elements
Concept	Description
Model	The data shown to the user in the view and with which the user interacts
Scope	A context where the model is stored so that controllers, directives and expressions can access it
Expressions	Executes JavaScript code inside brackets {{ }}.
Compiler	Parses the template and instantiates directives and expressions
Filter	Formats the value of an expression for display to the user
View	what the user sees (the DOM)
Data Binding	Sync data between the model and the view
Controller	Maintains the application data and business logic
Module	a container for different parts of an app including controllers, services, filters, directives which configure the Injector
Service	Reusable business logic, independent of views
Dependency Injection	Creates and wires objects and functions
Injector	Dependency injection container

RESULT

Thus a web application had been created using Angular framework.