KGiSL INSTITUTE OF TECHNOLOGY

(Approved by AICTE & Affiliated to Anna University, Chennai)
(Accredited by NBA)

DEPARTMENT OF INFORMATION TECHNOLOGY

NAME :

REG. No :

COURSE : B.TECH - IT

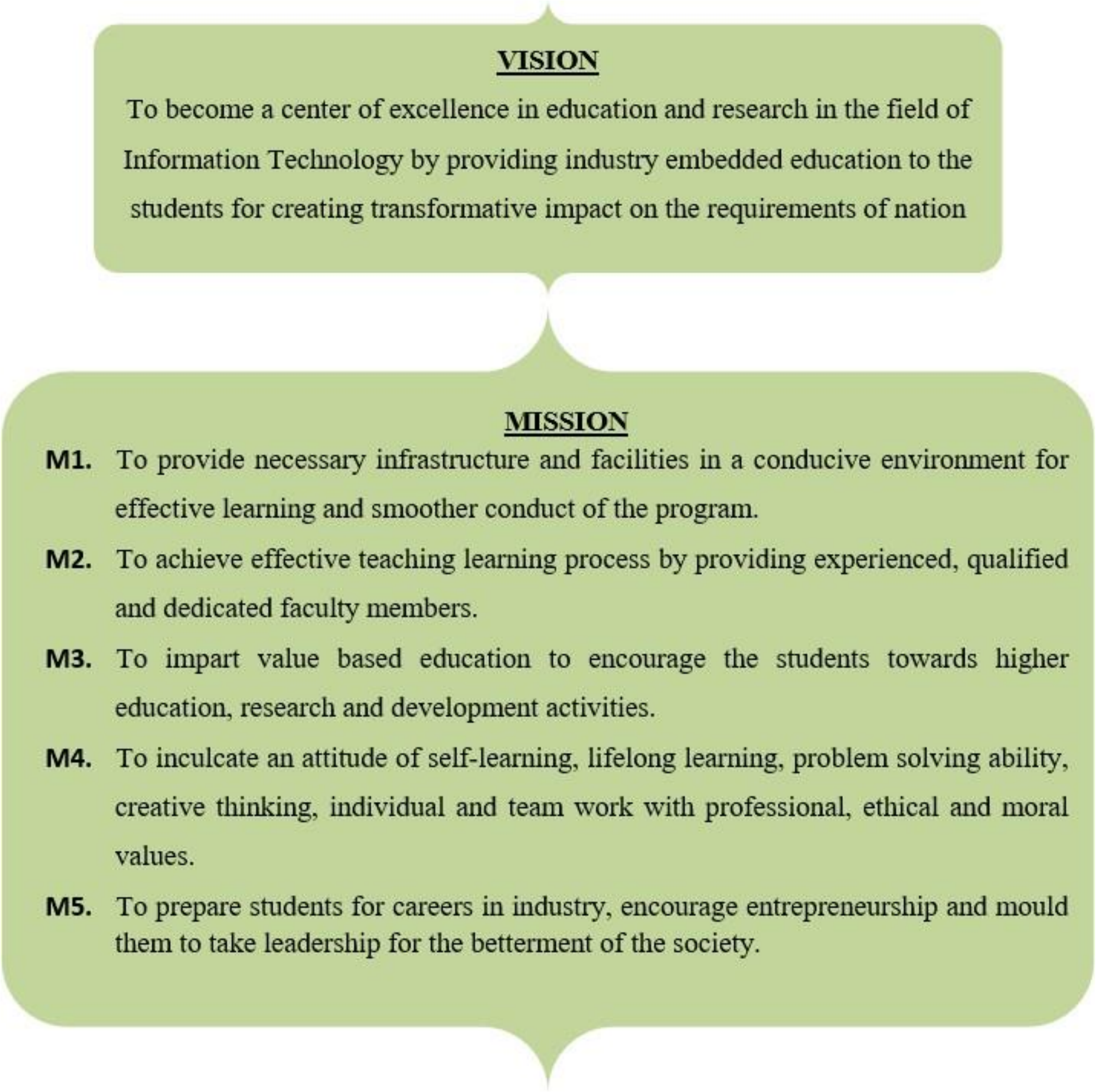
SUB.CODE : IT8211

SUBJECT : WEB TECHNOLOGY LABORATORY

YEAR & SEM : THIRD YEAR & V SEMESTER

BATCH : 2020-2024

DEPARTMENT OF INFORMATION TECHNOLOGY



VISION

To become a center of excellence in education and research in the field of Information Technology by providing industry embedded education to the students for creating transformative impact on the requirements of nation

MISSION

M1. To provide necessary infrastructure and facilities in a conducive environment for effective learning and smoother conduct of the program.

M2. To achieve effective teaching learning process by providing experienced, qualified and dedicated faculty members.

M3. To impart value based education to encourage the students towards higher education, research and development activities.

M4. To inculcate an attitude of self-learning, lifelong learning, problem solving ability, creative thinking, individual and team work with professional, ethical and moral values.

M5. To prepare students for careers in industry, encourage entrepreneurship and mould them to take leadership for the betterment of the society.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: To enable graduates to think logically, pursue lifelong learning and will have the capacity to understand technical issues related to computing systems and to design optimal solutions.

PEO 2: To enable graduates to develop hardware and software systems by understanding the importance of social, business and environmental needs in the human context.

PEO 3: To enable graduates to gain employment in organizations and establish themselves as professionals by applying their technical skills to solve real world problems and meet the diversified needs of industry, academia and research.

PROGRAMME OUTCOMES (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one‘s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OBJECTIVES (PSOs)

1. To create, select, and apply appropriate techniques, resources, modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
2. To manage complex IT projects with consideration of the human, financial, ethical and environmental factors and an understanding of risk management processes, and operational and policy implications.

\

KGiSL INSTITUTE OF TECHNO

(Approved by AICTE & Affiliated to Anna University, C

(Accredited by NBA)

NAME :

CLASS :

REGISTER NUMBER :

Certified that, this is a bonafide record of practical work done by_____________________of **INFORMATION TECHNOLOGY** in **WEB TECHNOLOGY** Laboratory, during **V** semester of academic year 2022-2023.

Faculty In-Charge **Head of the Department**

Submitted during Anna University Practical Examination held on_____________at KGiSL Institute of Technology, Coimbatore – 641 035.

Internal Examiner **External Examiner**

INDEX

EX NO:1 CREATE A WEB PAGE USING HTML

Aim:

To create a web page with the following using HTML

1) To embed an image map in web page.

2) To fix hot spots.

3) Show all the related information when the hot spots are clicked.

Procedure :

Step 1: Start with basic opening tags

Step 2: Using image tag, India map source is obtained and specified.

Step 3: Using map tag different states of India is mapped.

Step 4: In the map tag the area shape of states with different quardinates are specified.

Step 5: Using "**href"** attribute the wikipedia page of each states are linked.

Step 6: Repeat the process for all the other states

Step 7: Place the html file and india map image in a folder.

Step 8: On selecting a particular state the wikipedia page for that state will open.

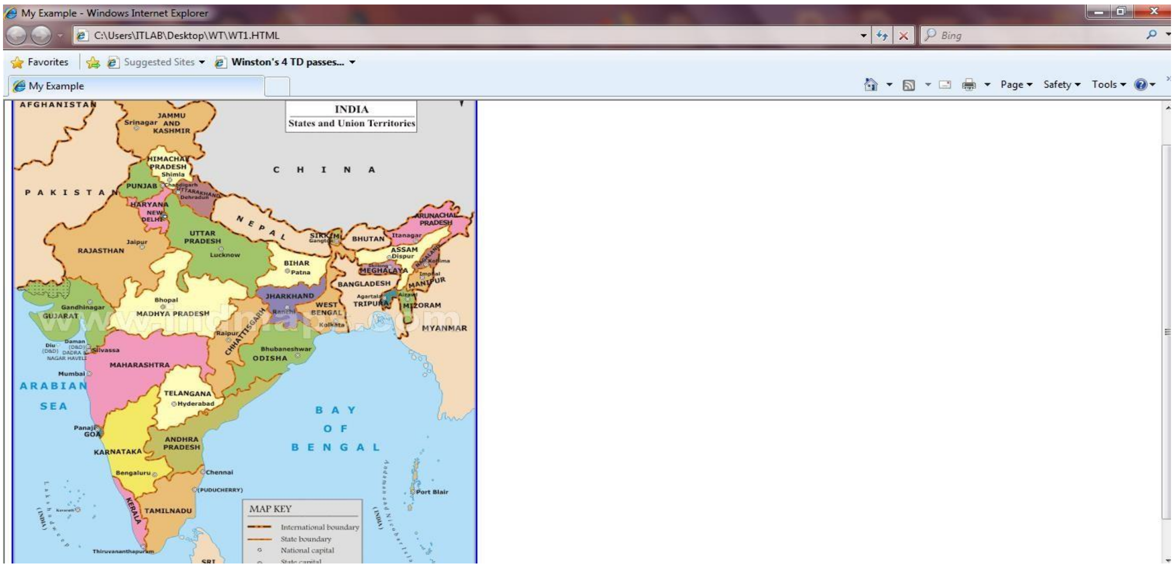
Source code:

```
<!DOCTYPE html>
<title>My Example</title>

<!-- Image -->
<img usemap="#India" src="india.gif"
alt="Map of India">

<!-- Map -->
<map name="India">
   <area shape="poly"
coords="384,1399,462,1367,526,1302,536,1200,499,1176,415,1225,365,1275"
href="https://en.wikipedia.org/wiki/Tamil_Nadu" target="_blank" alt="tamilnadu">
   <area shape="poly" coords="290,1196,377,1396,385,1314,354,1244"
href="https://en.wikipedia.org/wiki/Kerela" target="_blank" alt="kerela">
   <area shape="poly"
coords="278,1180,273,1023,370,959,425,918,398,1070,395,1236,458,1174"
href="https://en.wikipedia.org/wiki/Karnataka" target="_blank" alt="karnataka">
   <area shape="poly" coords="475,1184,407,1044,453,870,543,907,621,949,667,969"
href="https://en.wikipedia.org/wiki/Andhra_Pradesh" target="_blank" alt="andhra pradesh">
```

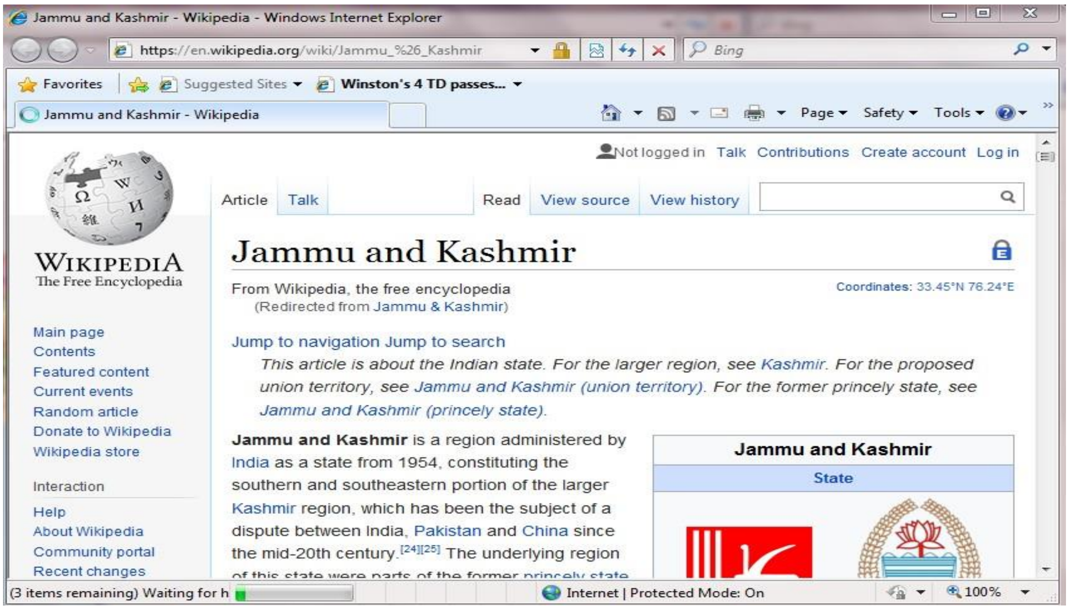
```
    <area shape="poly" coords="256,1040,204,856,310,754,458,770,556,781,558,840"
href="https://en.wikipedia.org/wiki/Maharastra" target="_blank" alt="maharastra">
    <area shape="poly" coords="60,730,153,600,271,675,258,763,151,782"
href="https://en.wikipedia.org/wiki/Gujarat" target="_blank" alt="gujarat">
    <area shape="poly"
coords="344,769,292,680,387,631,456,599,511,751,613,712,590,612,458,774"
href="https://en.wikipedia.org/wiki/Madhya_Pradesh" target="_blank" alt="madhya
pradesh">
    <area shape="poly" coords="553,895,597,730,687,657,691,754,617,908"
href="https://en.wikipedia.org/wiki/Chhattisgarh" target="_blank" alt="chhattisgarh">
    <area shape="poly" coords="594,944,679,784,837,735,844,824,761,883"
href="https://en.wikipedia.org/wiki/Orissa" target="_blank" alt="orissa">
    <area shape="poly" coords="894,501,833,668,880,755,938,728"
href="https://en.wikipedia.org/wiki/West_Bengal" target="_blank" alt="west bengal">
    <area shape="poly" coords="95,464,243,643,376,571,395,451,262,374"
href="https://en.wikipedia.org/wiki/Rajasthan" target="_blank" alt="rajasthan">
    <area shape="poly" coords="445,335,464,510,580,600,681,602,720,521"
href="https://en.wikipedia.org/wiki/Uttar_Pradesh" target="_blank" alt="uttar pradesh">
    <area shape="poly" coords="713,489,849,536,866,590,769,624,709,619"
href="https://en.wikipedia.org/wiki/Bihar" target="_blank" alt="bihar">
    <area shape="poly" coords="282,342,393,344,364,263"
href="https://en.wikipedia.org/wiki/Punjab" target="_blank" alt="punjab">
    <area shape="poly" coords="388,290,394,214,487,256,446,313"
href="https://en.wikipedia.org/wiki/Himanchal_Pradesh" target="_blank" alt="himanchal
pradesh">
    <area shape="poly" coords="362,236,308,107,325,15,464,91,546,134,504,203"
href="https://en.wikipedia.org/wiki/Jammu & Kashmir" target="_blank" alt="jammu &
kashmir">
    <area shape="poly" coords="964,515,1204,453,1114,580"
href="https://en.wikipedia.org/wiki/Assam" target="_blank" alt="assam">
    <area shape="poly" coords="1041,460,1215,378,1266,457,1114,495"
href="https://en.wikipedia.org/wiki/Arunachal_Pradesh" target="_blank" alt="arunachal
pradesh">
    <area shape="poly" coords="308,358,408,334,431,450,400,450"
href="https://en.wikipedia.org/wiki/Haryana" target="_blank" alt="haryana">
    <area shape="poly" coords="485,517,558,433,437,350,423,467"
href="https://en.wikipedia.org/wiki/Delhi" target="_blank" alt="delhi">
    <area shape="poly" coords="483,375,454,302,554,325,547,403"
href="https://en.wikipedia.org/wiki/Uttaranchal" target="_blank" alt="uttaranchal">
    <area shape="poly" coords="888,492,909,455,922,495"
href="https://en.wikipedia.org/wiki/Sikkim" target="_blank" alt="sikkim">
    <area shape="poly" coords="688,634,853,598,822,730,729,728"
href="https://en.wikipedia.org/wiki/Jharkhand" target="_blank" alt="jharkhand">
    <area shape="poly" coords="969,554,1075,550,1074,596,979,587"
href="https://en.wikipedia.org/wiki/Meghalaya" target="_blank" alt="meghalaya">
    <area shape="poly" coords="1044,631,1033,676,1054,696,1070,660,1074,638"
href="https://en.wikipedia.org/wiki/Tripura" target="_blank" alt="tripura">
    <area shape="poly" coords="1090,629,1092,700,1121,735,1129,660"
href="https://en.wikipedia.org/wiki/Mizoram" target="_blank" alt="mizoram">
```

```
    <area shape="poly" coords="1118,638,1119,577,1166,566,1172,604,1158,644"
href="https://en.wikipedia.org/wiki/Manipur" target="_blank" alt="Manipur">
    <area shape="poly" coords="1117,567,1169,504,1195,501,1185,557"
href="https://en.wikipedia.org/wiki/Nagaland" target="_blank" alt="nagaland">
    <area shape="poly" coords="237,1040,261,1055,251,1073,258,1078"
href="https://en.wikipedia.org/wiki/Goa" target="_blank" alt="goa">
</map>
```

Output:



On selecting Jammu & Kashmir:



Jammu and Kashmir

From Wikipedia, the free encyclopedia
(Redirected from Jammu & Kashmir)

Coordinates: 33.45°N 76.24°E

Jump to navigation Jump to search

This article is about the Indian state. For the larger region, see Kashmir. For the proposed union territory, see Jammu and Kashmir (union territory). For the former princely state, see Jammu and Kashmir (princely state).

Jammu and Kashmir is a region administered by India as a state from 1954, constituting the southern and southeastern portion of the larger Kashmir region, which has been the subject of a dispute between India, Pakistan and China since the mid-20th century.[24][25] The underlying region of this state were parts of the former princely state

Jammu and Kashmir

State

Result:

Thus the html web page has been successfully created.

EX NO:2 CREATE A WEB PAGE WITH CASCADING STYLE SHEETS

2A.CREATE A WEB PAGE WITH EXTERNAL CSS

Aim:

To create a webpage with the following css method.

- External

Algorithm:

Step 1: Start the process.

Step 2: Write the html code to perform external css method with an extension Of .html.

Step 2.1: write a css program contains the styles to the html with the Extension of .css.

Step 2.2: Insert a linking statement that links the html with the css.

Step 3: Display the output in any of the browser.

Step 4: Stop the process.

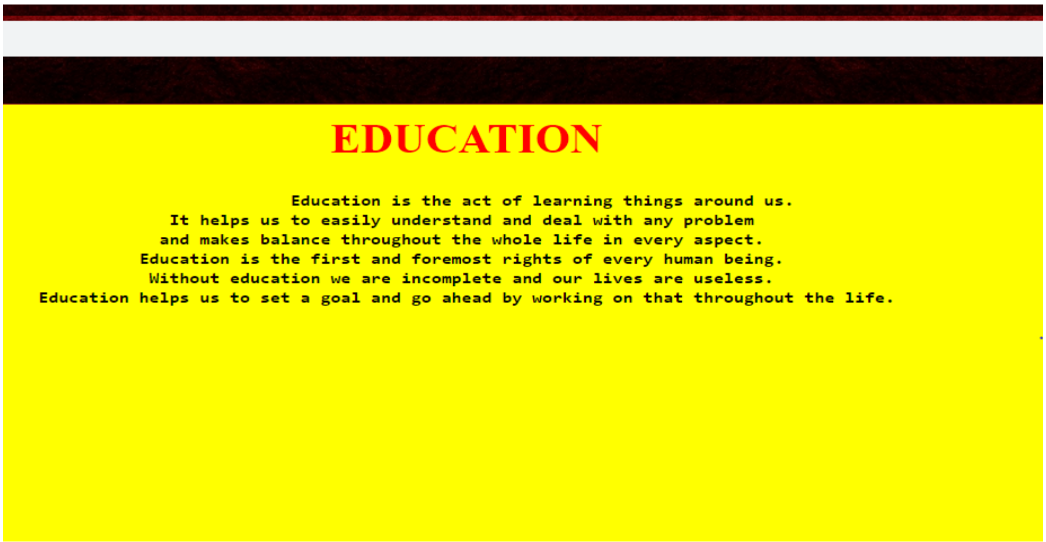
Source code:

```
<html>
<head>
<link rel="stylesheet" href ="external.css">
<title>external.css</title>
</head>
<body>
<center><h1>EDUCATION</h1></center>
<center><pre>
                    Education is the act of learning things around us. It helps us to easily understand
and deal with any problem and makes balance throughout the whole life in every aspect.
Education is the first and foremost rights of every human being. Without education we are
incomplete and our lives are useless. Education helps us to set a goal and go ahead by working
on that throughout the life.
</pre></center>
</body>
</html
```

EXTERNAL.CSS:

```
body
{
        background-color:yellow;
}
h1
{
        color:red;
}
pre
{
        font-weight:bold;
}
```

OUTPUT:



EDUCATION

```
                        Education is the act of learning things around us.
            It helps us to easily understand and deal with any problem
           and makes balance throughout the whole life in every aspect.
         Education is the first and foremost rights of every human being.
          Without education we are incomplete and our lives are useless.
   Education helps us to set a goal and go ahead by working on that throughout the life.
```

RESULT:

Thus the webpages are created and the output was produced successfully.

2B.CREATE A WEB PAGE WITH INTERNAL CSS

Aim:

To create a webpage with the following css method.

- Internal

Procedure:

Step 1: Start the process.

Step 2: write a html code to perform internal css method.

Step 2.1: Add the styles to the webpage by using the <style> tag.

Step 2.2: Inside the style tag mention the areas that needs styling.

Step 2.3: The syntax are given within the {}.

Step 2.4:close the style with </style> tag.

Step 3: Display the output in any of the browser.

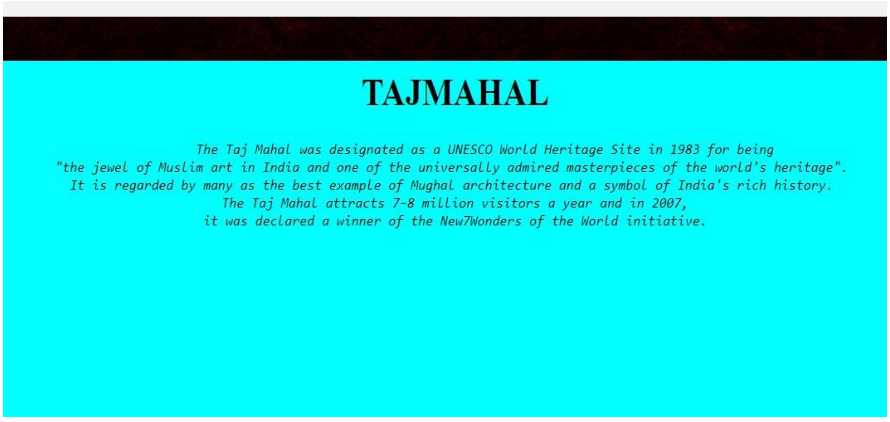
Step 4: Stop the process.

Source code:

```
<html>
<head>
<title>internal.css</title>
<style>
body
{
background-color:cyan;
}
h1{
color:grassgreen;
}
pre
{
font-style:italic;
}</style>
</head>
```

```
<body>
<center><h1>TAJMAHAL</h1></center>
<center><pre>        The Taj Mahal was designated as a UNESCO World Heritage Site in 1983 for being
"the jewel of Muslim art in India and one of the universally admired masterpieces of the world's heritage".
It is regarded by many as the best example of Mughal architecture and a symbol of India's rich history.
The Taj Mahal attracts 7–8 million visitors a year and in 2007,
it was declared a winner of the New7Wonders of the World initiative.
</pre></center>
</body>
</html>
```

OUTPUT:



TAJMAHAL

```
        The Taj Mahal was designated as a UNESCO World Heritage Site in 1983 for being
"the jewel of Muslim art in India and one of the universally admired masterpieces of the world's heritage".
It is regarded by many as the best example of Mughal architecture and a symbol of India's rich history.
The Taj Mahal attracts 7–8 million visitors a year and in 2007,
it was declared a winner of the New7Wonders of the World initiative.
```

RESULT:

Thus the webpages are created and the output was produced successfully.

2C. CREATE A WEB PAGE WITH INLINE CSS

Aim:

To create a webpage with the following css method.

- Inline

Procedure:

Step 1: Start the process.

Step 2: write a html code to perform inline css method.

Step 2.1: The styles are included using the keyword “style”.

Step 2.2: The styles are specified along with the html tag.

Step 2.3: Only the specified tag elements will change with corresponding css styles.

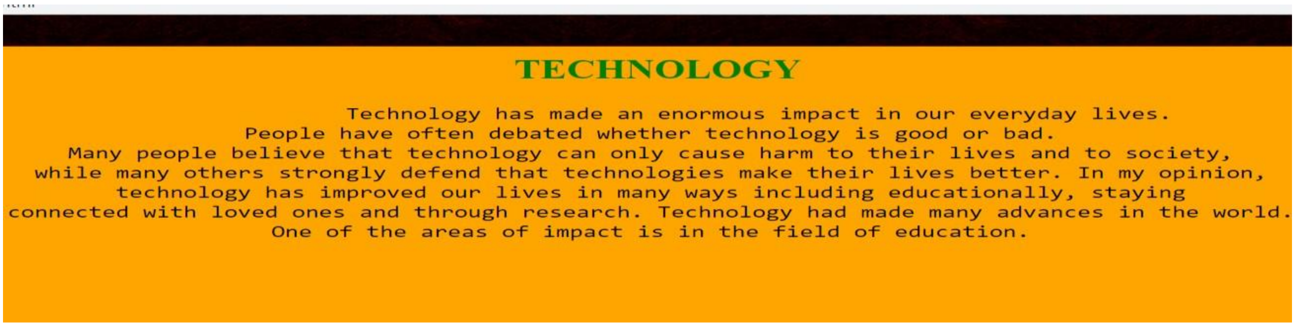
Step 3: Display the output in any of the browser.

Step 4: Stop the process.

<u>Source code:</u>

```
<html>

<head>

<title>inline.css</title>

</head>

<body style="background-color:orange">

<center>

<h1 style="color:green">TECHNOLOGY</h1>

</center>

<center><pre style="font-size:20px">

                    Technology has made an enormous impact in our everyday lives. People have
often debated whether technology is good or bad. Many people believe that technology can
only cause harm to their lives and to society, while many others strongly defend that
technologies make their lives better. In my opinion, technology has improved our lives in many
ways including educationally, staying connected with loved ones and through research.
Technology had made many advances in the world. One of the areas of impact is in the field
of education.

</pre></center>

</body>

</html>
```

OUTPUT:



TECHNOLOGY

Technology has made an enormous impact in our everyday lives.
People have often debated whether technology is good or bad.
Many people believe that technology can only cause harm to their lives and to society,
while many others strongly defend that technologies make their lives better. In my opinion,
technology has improved our lives in many ways including educationally, staying
connected with loved ones and through research. Technology had made many advances in the world.
One of the areas of impact is in the field of education.

RESULT:

Thus the webpages are created and the output was produced successfully.

EX NO:3 CLIENT SIDE SCRIPTING FOR VALIDATING WEB FORM CONTROLS

Aim:

To write the HTML codes using JavaScript and css to create client side scripts for validating Web Form Controls.

Procedure:

Step 1: Start the code.

Step 2: Create the HTML header and give the title as “WebForm”.

Step 3: Specify the style type as “text/css”.

Step 4: Define the properties for the Layout and Layer.

Step 5: Specify the script type as “text/javascript”.

Step 6: Create a function as “validate ()”.

Step 7: Specify the conditions for the form to be created using javascript.

Step 8: Open the body of the HTML.

Step 9: Create the division as “Layer”.

Step 10: Create a form and name it. Create a table for better alignment of the contents of the form.

Step11: Specify the requirements of the form like Name, Qualification, Date Of Birth, etc.

Step 12: Define the input type, name and id of all the required parameters.

Step 13: Create a “Submit” button and make a link to the function using onclick=“validate ()”.

Step 14: Save and run the HTML file to get the output.

<u>Source code:</u>

<u>Index.html:</u>

```
<!DOCTYPE html>
<html>
   <head>
      <title>Metazod Community Registration</title>
      <link rel="stylesheet" href="style.css">
      <script src="script.js"></script>
   </head>
   <body>
```

```
<img src="logo.jpg" alt="logo" height="200px">
    <form method="post" name="form" action="register.php" onsubmit="return formvalidation()">
      <table align="center">
        <tr>
          <th colspan="2">Registration Form</th>
        </tr>
        <tr>
          <td align="right">Name:</td>
          <td><input type="text" size="65" name="Name"></td>
        </tr>
        <tr>
          <td align="right">Date of Birth:</td>
          <td><input type="date" size="65" name="DOB"></td>
        </tr>
        <tr>
          <td align="right">Gender:</td>
          <td><input type="radio" name="gender" value="m" checked />Male<input
            type="radio" name="gender" value="f" />Female</td>
        </tr>
        <tr>
          <td align="right">Department</td>
          <td><select name="Department">
              <option value="-1" selected>[select]</option>
              <option value="1">IT</option>
              <option value="2">CSE A</option>
              <option value="3">CSE B</option>
              <option value="4">ECE</option>
              <option value="5">AI & DS</option>
              <option value="6">CS & BS</option>
              <option value="7">MECH</option>
          </select></td>
        </tr>
        <tr>
          <td align="right">Year of Study</td>
          <td><Select name="Year">
            <option value="-1" selected>[select]</option>
            <option value="1">1st Year</option>
            <option value="2">2nd Year</option>
            <option value="3">3rd Year</option>
            <option value="4">4th year</option>
          </Select></td>
        </tr>
        <tr>
          <td align="right">Mobile No:</td>
          <td><input type="text" size="65" name="Mobile"></td>
          </tr>
          <tr>
          <td align="right">E-mail Address:</td>
          <td><input type="text" size="65" name="Email"></td>
        </tr>
        <tr>
          <td><input type="submit" value="Submit" name="submit"></td>
        </tr>

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

Style.CSS

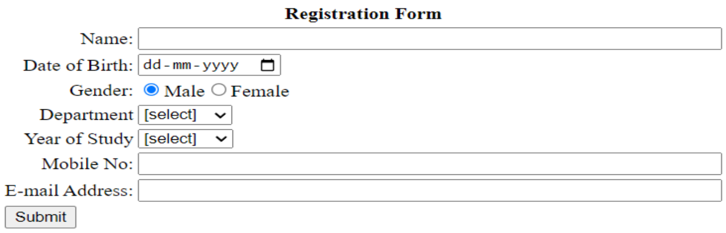
```
body {
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 100% 100%;
 }
```

Script.js

```
function formvalidation(){
  var name = document.form.Name;
  var dob = document.form.DOB;
  var gender = document.form.gender;
  var department = document.form.Department;
  var year = document.form.Year;
  var no = document.form.Mobile;
  var email = document.form.Email;

  if (name.value == "") {
    window.alert("Please enter your name.");
    name.focus();
    return false;
  }
```

```
    if (dob.value == "") {
        window.alert("Please enter your date of birth.");
        dob.focus();
        return false;
    }
    if (gender.value == "") {
        window.alert("Select your gender.");
        gender.focus();
        return false;
    }
    if (department.value == "") {
        window.alert("Select your department.");
        department.focus();
        return false;
    }
    if (year.value == "") {
        window.alert("Select your of Study.");
        year.focus();
        return false;
    }
    if (no.value == "") {
        window.alert("Please enter your Ph No.");
        no.focus();
        return false;
    }
    if (email.value == "") {
        window.alert("Please enter your email.");
        email.focus();
        return false;
    }



}
```

OUTPUT:



Registration Form

Name:

Date of Birth: dd-mm-yyyy

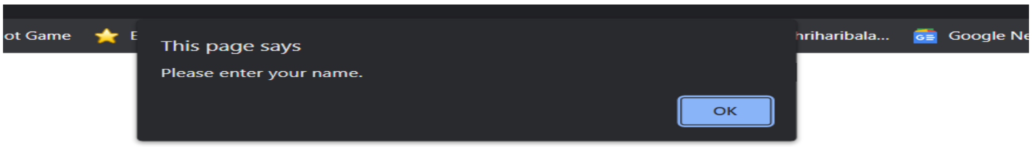
Gender: ● Male ○ Female

Department [select]

Year of Study [select]

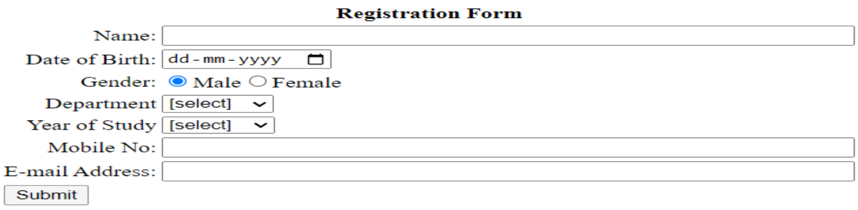
Mobile No:

E-mail Address:



This page says

Please enter your name.



Registration Form

Name:

Date of Birth: dd-mm-yyyy

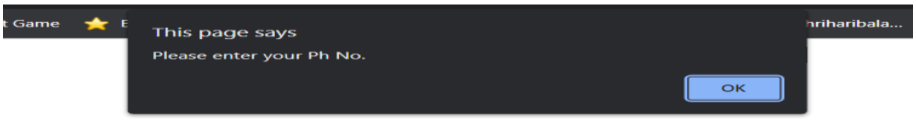
Gender: ● Male ○ Female

Department [select]

Year of Study [select]

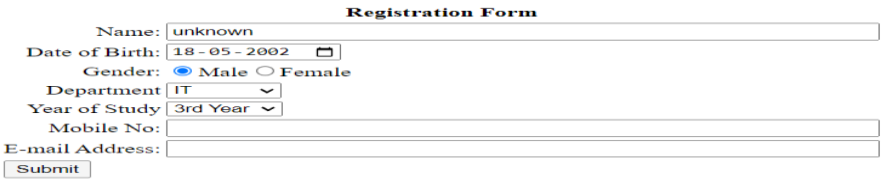
Mobile No:

E-mail Address:



This page says

Please enter your Ph No.



Registration Form

Name: unknown

Date of Birth: 18-05-2002

Gender: ● Male ○ Female

Department IT

Year of Study 3rd Year

Mobile No:

E-mail Address:

RESULT:

Thus the validation using javascript and html output was produced successfully.

EX NO:4 INSTALLATION OF APACHE TOMCAT WEB SERVER

Aim:

To install the Apache Tomcat web server in the system

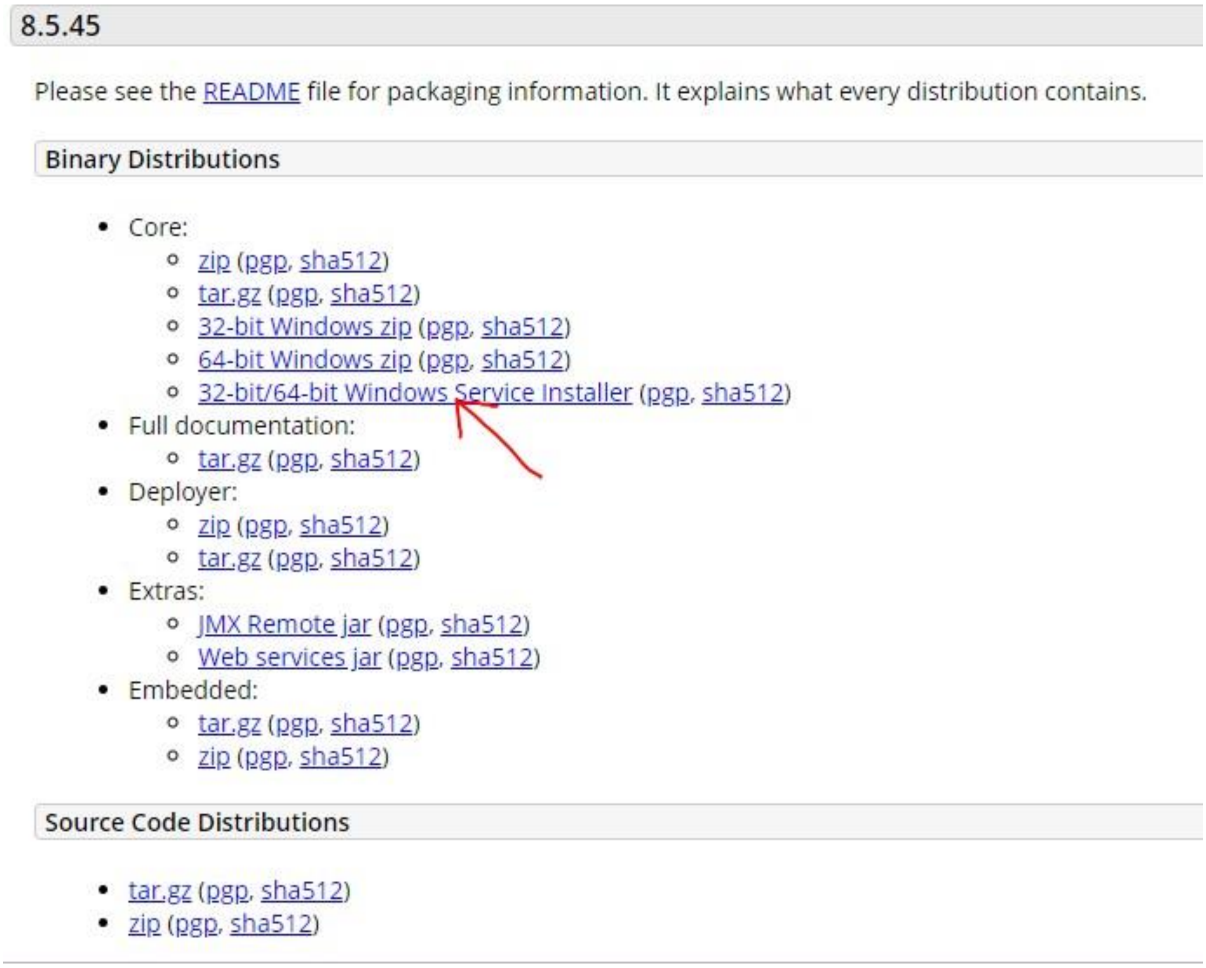
Procedure:

Step 1 :

Download the Apache Tomcat web server from the site.
https://tomcat.apache.org/download-80.cgi

Step 2:

Click on the link mentioned in the following screenshot:



8.5.45

Please see the README file for packaging information. It explains what every distribution contains.

Binary Distributions

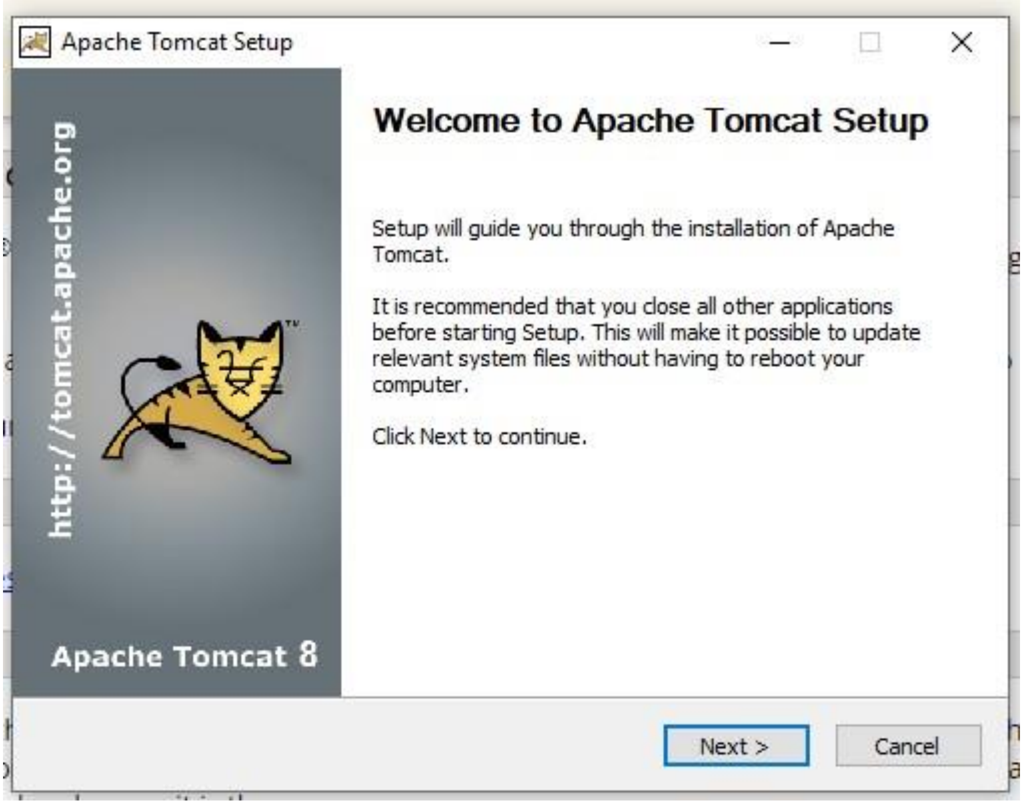
- Core:
  - zip (pgp, sha512)
  - tar.gz (pgp, sha512)
  - 32-bit Windows zip (pgp, sha512)
  - 64-bit Windows zip (pgp, sha512)
  - 32-bit/64-bit Windows Service Installer (pgp, sha512)
- Full documentation:
  - tar.gz (pgp, sha512)
- Deployer:
  - zip (pgp, sha512)
  - tar.gz (pgp, sha512)
- Extras:
  - JMX Remote jar (pgp, sha512)
  - Web services jar (pgp, sha512)
- Embedded:
  - tar.gz (pgp, sha512)
  - zip (pgp, sha512)

Source Code Distributions

- tar.gz (pgp, sha512)
- zip (pgp, sha512)

Step 3:

Install the downloaded file an proceed by clicking the next button until you reach configuration option. Here we can set the name the service, shutdown port and running port of Tomcat. By default Tomcat runs on port 8080.

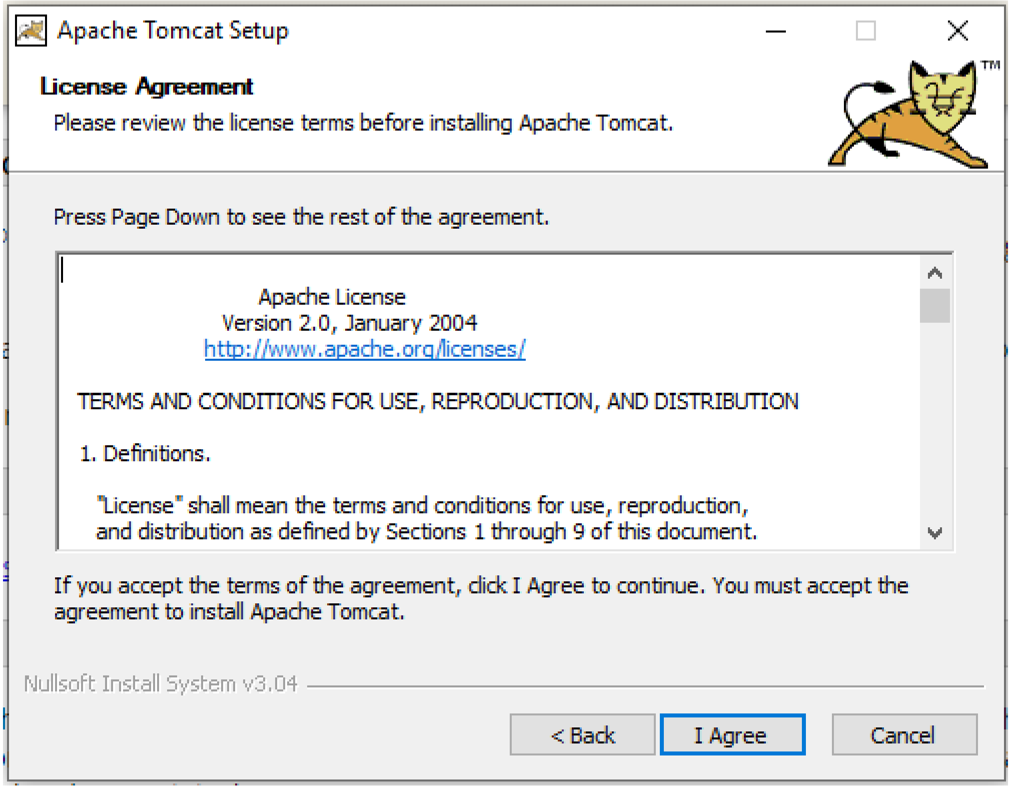


Welcome to Apache Tomcat Setup

Setup will guide you through the installation of Apache Tomcat.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



License Agreement

Please review the license terms before installing Apache Tomcat.

Press Page Down to see the rest of the agreement.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

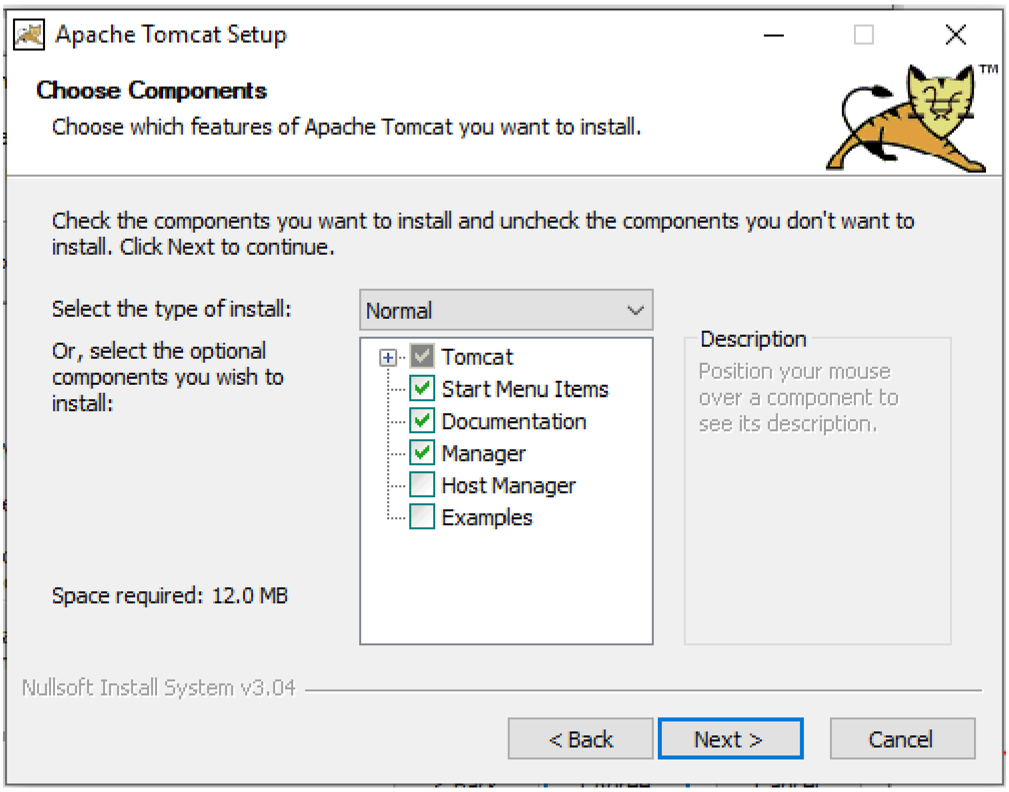
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Apache Tomcat.

Nullsoft Install System v3.04



Apache Tomcat Setup

Choose Components

Choose which features of Apache Tomcat you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Normal

Or, select the optional components you wish to install:

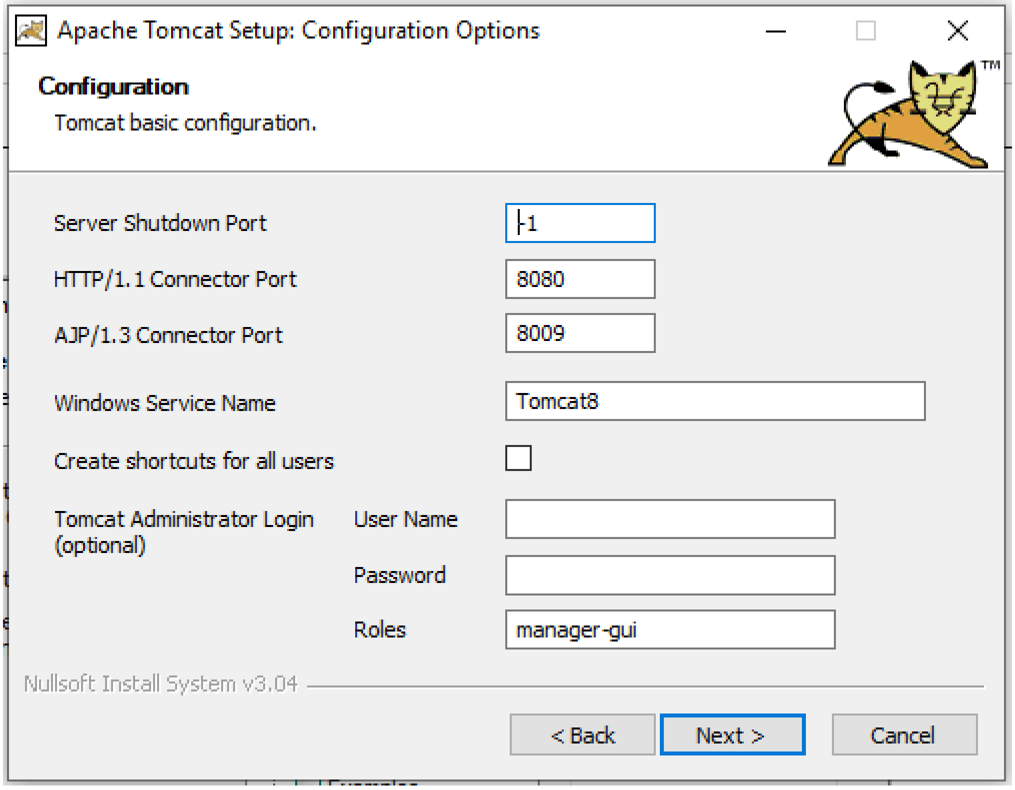
- ☑ Tomcat
  - ☑ Start Menu Items
  - ☑ Documentation
  - ☑ Manager
  - ☐ Host Manager
  - ☐ Examples

Description

Position your mouse over a component to see its description.

Space required: 12.0 MB

Nullsoft Install System v3.04



Apache Tomcat Setup: Configuration Options

Configuration

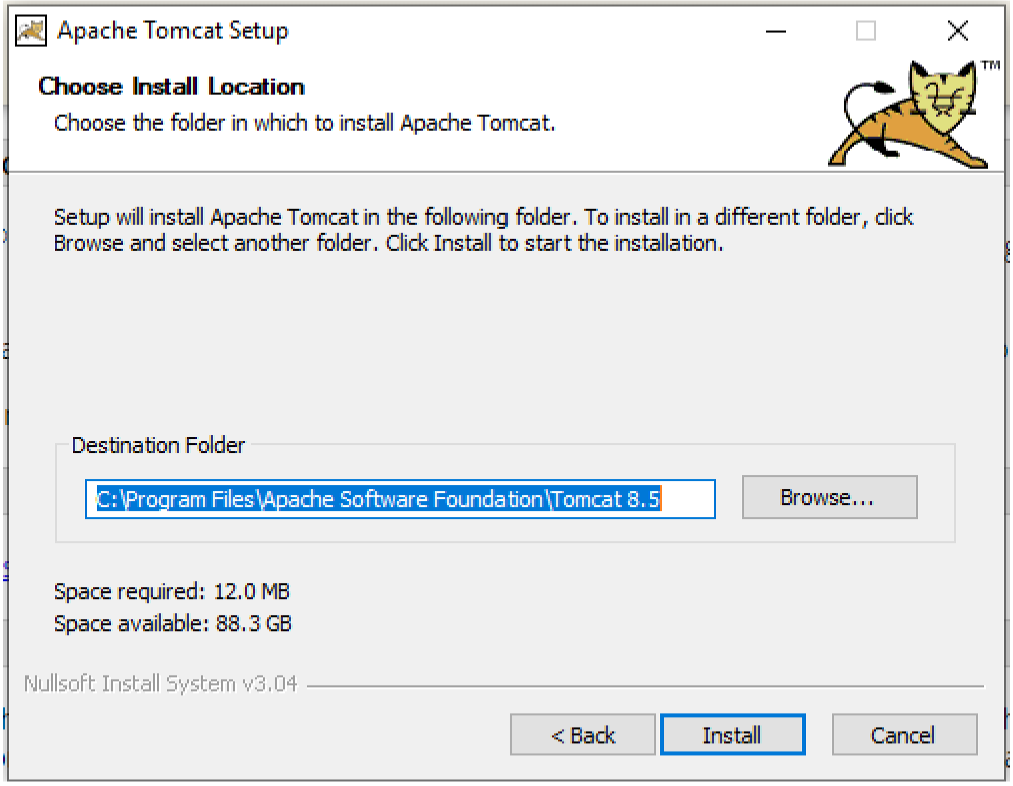
Tomcat basic configuration.

Server Shutdown Port		-1
HTTP/1.1 Connector Port		8080
AJP/1.3 Connector Port		8009
Windows Service Name		Tomcat8
Create shortcuts for all users		☐
Tomcat Administrator Login (optional)	User Name	
	Password	
	Roles	manager-gui

Nullsoft Install System v3.04

Step 4:

After completing the installation, Tomcat folder can present at C:\Program Files\Apache Software Foundation\Tomcat 8.5



Choose Install Location

Choose the folder in which to install Apache Tomcat.

Setup will install Apache Tomcat in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files\Apache Software Foundation\Tomcat 8.5

Space required: 12.0 MB

Space available: 88.3 GB

We can start up Tomcat using startup.bat file and stop the tomcat using shutdown.bat file.

RESULT:

Thus the install of Apache Tomcat web server in the system is done successfully

EX NO:5 SESSION TRACKING IN JAVA USING SERVLETS

Aim:

To Create a Java Program Session Tracking Using Servlet.

Procedure:

Step 1: Start the process.

Step 2: Write the Java Servlet Program For Session Tracking.

Step 3: Write the Servlet Program and Write the Header file to Implement the Session Tracking implement

the Session Tracking.

Step4: Include the HTTP Servlet Request and HTTP Servlet Response as a Header File.

Step 5: Use the Exception as IOException and the Servlet Exception to implement the code.

Step 6: Create the String title,visitcount,visitcountkey,userid and the user.

Step 7: Using If Statement to Count the Visting the No.of times by user.

Step 8: Now Visit the Page Once user Will Refresh the Page and Last time as been visit the web page by the user.

Step 9: Print the Entire Visit Count by the user and use HTML to Align the Visit Count.

Step 10: Display the Output.

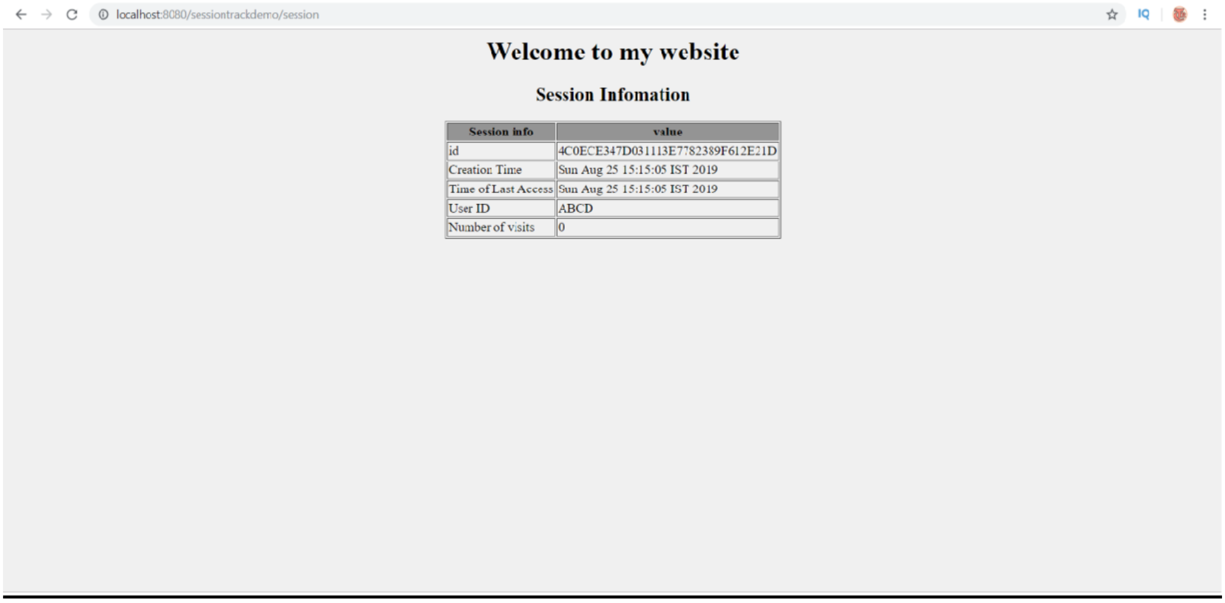
Step 11: Stop the Process.

Source Code:

```
// Import required java libraries

import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

import java.util.*;

 // Extend HttpServlet class

public class SessionTrack extends HttpServlet {

   public void doGet(HttpServletRequest request,
                 HttpServletResponse response)
             throws ServletException, IOException
    {
      // Create a session object if it is already not  created.
      HttpSession session = request.getSession(true);
      // Get session creation time.
      Date createTime = new Date(session.getCreationTime());
      // Get last access time of this web page.
      Date lastAccessTime =
                   new Date(session.getLastAccessedTime());

      String title = "Welcome Back to my website";
      Integer visitCount = new Integer(0);
      String visitCountKey = new String("visitCount");
      String userIDKey = new String("userID");
      String userID = new String("ABCD");
      // Check if this is new comer on your web page.
      if (session.isNew()){
        title = "Welcome to my website";
        session.setAttribute(userIDKey, userID);
      } else {
        visitCount = (Integer)session.getAttribute(visitCountKey);
        visitCount = visitCount + 1;
        userID = (String)session.getAttribute(userIDKey);
      }
      session.setAttribute(visitCountKey,  visitCount);
      // Set response content type
```

```
response.setContentType("text/html");
PrintWriter out = response.getWriter();

String docType =
"<!doctype html public \"-//w3c//dtd html 4.0 " +
"transitional//en\">\n";
out.println(docType +
        "<html>\n" +
        "<head><title>" + title + "</title></head>\n" +
        "<body bgcolor=\"#f0f0f0\">\n" +
        "<h1 align=\"center\">" + title + "</h1>\n" +
         "<h2 align=\"center\">Session Infomation</h2>\n" +
        "<table border=\"1\" align=\"center\">\n" +
        "<tr bgcolor=\"#949494\">\n" +
        " <th>Session info</th><th>value</th></tr>\n" +
        "<tr>\n" +
        "  <td>id</td>\n" +
        " <td>" + session.getId() + "</td></tr>\n" +
        "<tr>\n" +
        "  <td>Creation Time</td>\n" +
        "  <td>" + createTime +
        " </td></tr>\n" +
        "<tr>\n" +
        "  <td>Time of Last Access</td>\n" +
        "  <td>" + lastAccessTime +
        " </td></tr>\n" +
        "<tr>\n" +
            " <td>User ID</td>\n" +
            "  <td>" + userID +
            " </td></tr>\n" +
            "<tr>\n" +
            "  <td>Number of visits</td>\n" +
            " <td>" + visitCount + "</td></tr>\n" +
            "</table>\n" +
            "</body></html>");
  }
}
```

Output:



Welcome to my website

Session Infomation

Session info	value
id	4C0ECE347D031113E7782389F612E21D
Creation Time	Sun Aug 25 15:15:05 IST 2019
Time of Last Access	Sun Aug 25 15:15:05 IST 2019
User ID	ABCD
Number of visits	0

Result:

Thus the above program was executed successfully.

EX NO:6a THREE TIER APPLICATION - ONLINE EXAMINATION

Aim:

To create three tier application for online examination using jsp and database.

Procedure:

Step 1: Create a form to get the input from the user.
Step 2: Use radio buttons to make various options for the questions.
Step 3: Set the URL of the server (ExamServer.jsp) as the value of the action attribute.
Step 4: Use submit button to invoke the server and send the form data to the server.
Step 5: Create the JSP file with the following
Step 6: Read the input from the client.
Step 7: Retrieve the answers from the database.
Step 8: Match the answers from the user with the correct answers from the database table.
Step 9: For each correct answer increment the mark by 5.
Step 10: Server displays the mark and result to the client as a response.

Source code:

index.html

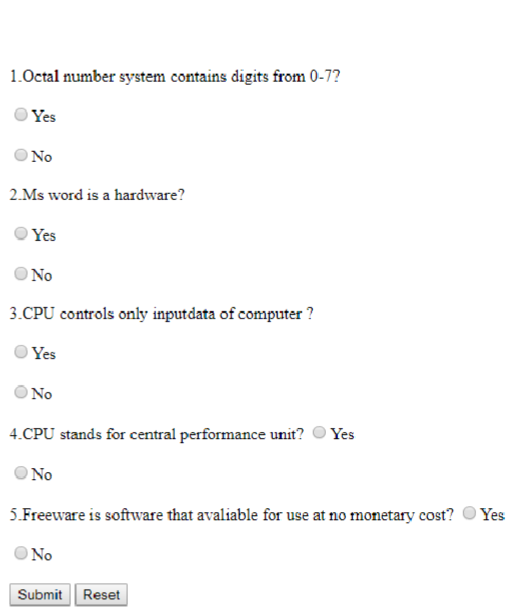
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title><h1>QUIZ</h1></title>
        <script src="validate.jsp"></script>
    </h:head>
    <h:body>
        <form  method="post" action="validate.jsp" name="Form1" >
        <center>  <h1>QUIZ</h1></center>
        1.Octal number system contains digits from 0-7?<BR></BR>
        <input type="radio" name="que1" value="Yes">Yes</input> <br></br>
         <input type="radio" name="que1" value="No">No</input> <br></br>

        2.Ms word is a hardware?<br></br>
        <input type="radio" name="que2" value="Yes">Yes</input> <br></br>
         <input type="radio" name="que2" value="No">No</input> <br></br>

        3.CPU controls only inputdata of computer ?<br></br>
        <input type="radio" name="que3" value="Yes">Yes</input> <br></br>
         <input type="radio" name="que3" value="No">No</input> <br></br>
          4.CPU stands for central performance unit?
          <input type="radio" name="que4" value="Yes">Yes</input> <br></br>
         <input type="radio" name="que4" value="No">No</input> <br></br>
          5.Freeware is software that avaliable for use at no monetary cost?
          <input type="radio" name="que5" value="Yes">Yes</input> <br></br>
         <input type="radio" name="que5" value="No">No</input> <br></br>
        <input type="submit" value="Submit" name="submit" ></input>
        <input type="reset" value="Reset" name="reset"></input>
        </form>
          </h:body>
</html>
```

Validate.jsp

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.lang.*"%>
<%@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>

        <%
            try{
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/Answer","root","");
                Statement stmt=conn.createStatement();
                ResultSet rs = stmt.executeQuery("select * from Question");
                String Answer1 = request.getParameter("que1");
                String Answer2 = request.getParameter("que2");
                String Answer3 = request.getParameter("que3");
                String Answer4 = request.getParameter("que4");
                String Answer5 = request.getParameter("que5");
                String Answer;
                int   Mark   =   0;
                while(rs.next()){
                    Answer = rs.getString("Answer");
                    if(Answer.equals(Answer1))
                    {
                    Mark += 2;
                    }
                    else
                    {
                    Mark += 0;
                    }

                if(Answer.equals(Answer2))
                    {
                    Mark += 2;
```

```
          }
          else
          {
          Mark += 0;
          }
         if(Answer.equals(Answer3))
          {
          Mark += 2;
          }
          else
          {
          Mark += 0;
          }

        if(Answer.equals(Answer4))
          {
          Mark += 2;
          }
          else
          {
          Mark += 0;
          }
        if(Answer.equals(Answer5))
          {
          Mark += 2;
          }
          else
          {
          Mark += 0;
          }
        }


        %>MARK<% out.print(Mark);
      }
      catch(Exception e){
        out.println("Error" + e);
      }
      %>
  </body>
</html>
```

OUTPUT:
index.html



QUIZ

1.Octal number system contains digits from 0-7?

○ Yes

○ No

2.Ms word is a hardware?

○ Yes

○ No

3.CPU controls only inputdata of computer ?

○ Yes

○ No

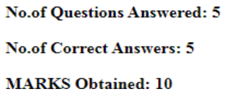
4.CPU stands for central performance unit? ○ Yes

○ No

5.Freeware is software that avaliable for use at no monetary cost? ○ Yes

○ No

validate.jsp



No.of Questions Answered: 5

No.of Correct Answers: 5

MARKS Obtained: 10

RESULT:

Thus the program using JSP and database connectivity was successfully executed and verified.

EX NO:6b THREE TIER APPLICATION - STUDENT MARKLIST

Aim:

To create three tier application for student mark list using JSP and database.

Procedure:

Step 1: Create a form to get the input (Register Number) from the user.
Step 2: Set the URL of the server (marklist.jsp) as the value of the action attribute.
Step 3: Use submit button to invoke the server and send the form data to the server.
Step 4: Create the JSP file with the following
Step 5: Read the parameter value (Register Number) from the form by using the method getParameter().
Step 6: Server retrieves the details from the database table with respect to the form input.
Step 7: Server displays the mark list to the client as the response.

Source code:

index.html

```
<!DOCTYPE html>

<!--

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

-->

<html>

<head>

<title>Form Validation Example</title>

<script>

function ValidateForm()

{

   var name = document.Form1.Name;

   var Fname = document.Form1.FName;
```

```
    if (name.value == "")
    {
        window.alert("Please enter your Username.");
        name.focus();
        return false;
    }
            if (Fname.value == "")
    {
        window.alert("Please enter your Password");
        Fname.focus();
        return false;
    }
    else{
        window.alert("Redirected");
    }
    return true;
}
</script>
<style>
    input[type="text"]
{
    background: transparent;
    border: none;
}
textarea{
    background: transparent;
    border: none; }
```

```
/*@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,300,700);*/

body{

  background: -webkit-linear-gradient(left, #25c481, #25b7c4);

  background: linear-gradient(to right, #25c481, #25b7c4);

  font-family: 'Roboto', sans-serif;

}

</style>

</head>

<body style="background-color:F9FFFF;">

    <br><br><br>

<form method="post" action="CheckLogin.jsp" name="Form1" onsubmit="return
ValidateForm();">

<center>

<table cellspacing="2" cellpadding="2" border="1">

                    <tr>

                        <th colspan="2">MarkSheet Information</th>

                    </tr>

<tr>

<td align="right">Name:</td>

<td> <input type="text" size="65" name="Name"></td>

</tr>

<tr>

<td align="right">Password:</td>

<td> <input type="text" size="65" name="FName"></td>

</tr>

<tr>

<td><input type="submit" value="Send" name="submit"></td>
```

```
  <td><input type="reset" value="Reset" name="reset"></td>
</tr>
            </table> </center>
</form>
</body>
</html>
```

Retrieve.jsp

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@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>
    <style>
        /*
            Side Navigation Menu V2, RWD
            ===================
            License:
            https://goo.gl/EaUPrt
            ===================
            Author: @PableraShow
 */
@charset "UTF-8";
@import url(https://fonts.googleapis.com/css?family=Open+Sans:300,400,700);
```

```
body {
  font-family: 'Open Sans', sans-serif;
  font-weight: 300;
  line-height: 1.42em;
  color:#Black;
  background-color:white;
}
h1 {
  font-size:3em;
  font-weight: 300;
  line-height:1em;
  text-align: center;
  color: Black;
}
h2 {
  font-size:1em;
  font-weight: 300;
  text-align: center;
  display: block;
  line-height:1em;
  padding-bottom: 2em;
  color: Black;
}
h2 a {
  font-weight: 700;
  text-transform: uppercase;
  color: #FB667A;
  text-decoration: none;
}
```

```
.blue { color:black; }
.yellow { color: Black; }
.container th h1 {
            font-weight: bold;
            font-size: 1em;
  text-align: left;
  color: black;
}
.container td {
            font-weight: normal;
            font-size: 1em;

}
.container {
            text-align: left;
            overflow: hidden;
            width: 50%;
            margin: 0 auto;
  display: table;
  padding: 0 0 8em 0;
}
.container td, .container th {
            padding-bottom: 2%;
            padding-top: 2%;
  padding-left:2%;
}
/* Background-color of the odd rows */
.container tr:nth-child(odd) {
            background-color: white;
```

```
}
/* Background-color of the even rows */
.container tr:nth-child(even) {
              background-color: #white;
}
.container th {
    background-color: #ffffff;
}
.container td:first-child { color: #FB667A; }
@media (max-width: 600px) {
.container td:nth-child(4),
.container th:nth-child(4) { display: none; }
}
    </style>
    <body>
        <div>
            <a href="index.html"><img src="whitehome.png" width="50" height="50"></a>
            <h1>MarkSheet</h1>
        </div>
        <%
           try{
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/Marksheet","root","");
                Statement stmt=conn.createStatement();
                ResultSet rs = stmt.executeQuery("select * from Marksheet");
                out.println("<br>");
                out.println("<br>");%>
                <table style="width:100%" border="5" class="container">
                    <tr>
    <th>SUBNAME</th>

    <th>SUBCODE</th>

    <th>GRADE</th>

    <th>STATUS</th>

  </tr>

                <%while(rs.next()){%>

    <tr>

    <td><%out.print(" " + rs.getString("SUBNAME"));%></td>

    <td><%out.print(" " + rs.getString("SUBCODE"));%></td>

    <td><%out.print(" " + rs.getString("GRADE"));%></td>

    <td><%out.print(" " + rs.getString("STATUS"));%></td>

  </tr>

                <%}%></table><%

            }catch(Exception e){

                out.println("Error" + e);

            }

        %>

    </body>

</html>
```

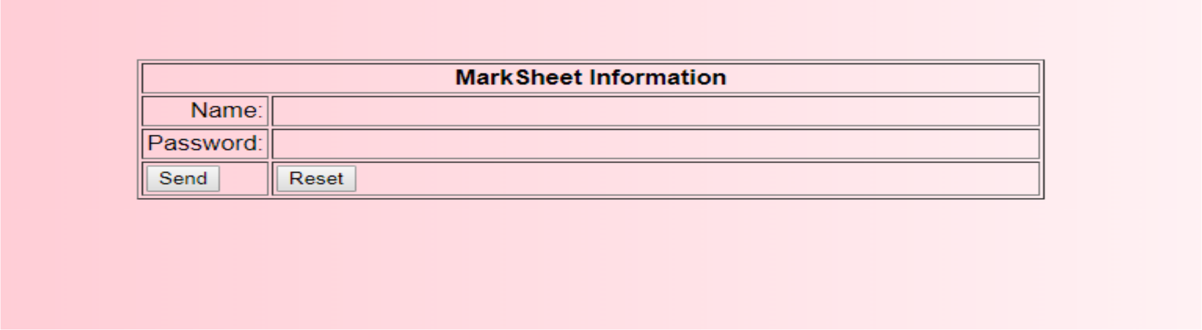
CheckLogin.jsp

```
<%--
    Document   : dataStore
    Created on : 21 Aug, 2019, 11:50:31 AM
    Author     : jahaziel_sam
--%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>
<%@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>
   <style>

   input[type="text"]
{
   background: transparent;
   border: none;
}

textarea{
   background: transparent;
   border: none;
}


/*@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,300,700);*/
body{
  background: -webkit-linear-gradient(left, #25c481, #25b7c4);
  background: linear-gradient(to right, #25c481, #25b7c4);
  font-family: 'Roboto', sans-serif;
}

</style>
   <body>
      <%
         try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/userlogin","root","");
            Statement stmt=conn.createStatement();
            ResultSet rs = stmt.executeQuery("select * from userlogin");
            out.println("<br>");
            String UserName = request.getParameter("Name");
            String Password = request.getParameter("FName");
            while(rs.next()){
            String us = rs.getString("UserName");
            String pwd = rs.getString("Password");
            if((UserName.equals(us)) && (Password.equals(pwd))){
              out.print("LoginSuccessfull");
              %>
            <a href = 'retrieveRecords.jsp'><button type=\"button\">Click to view the
MarkSheet!</button></a>
              <%
              break;
            }
            else{
              out.print("LoginFailed");
              break;
            }
            }
```

```
        %>
        <br><br>
        <center>

          <br>
          <br>

        </center>
        <%
      }catch(Exception e){
        out.println(e);
      }
    %>
  </body>
</html>
```

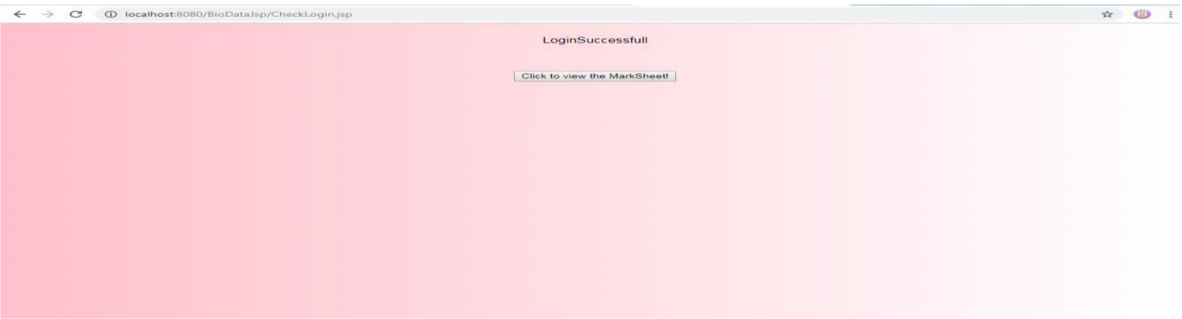
OUTPUT



MarkSheet Information

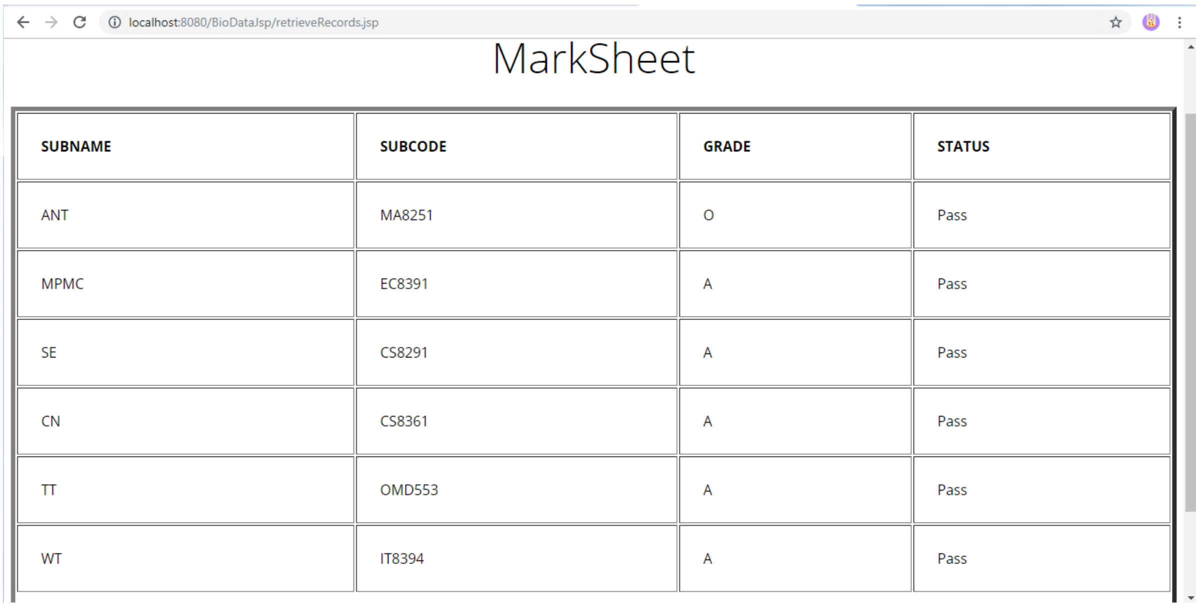
Name:

Password:



localhost:8080/BioDataJsp/CheckLogin.jsp

LoginSuccessfull



localhost:8080/BioDataJsp/retrieveRecords.jsp

MarkSheet

SUBNAME	SUBCODE	GRADE	STATUS
ANT	MA8251	O	Pass
MPMC	EC8391	A	Pass
SE	CS8291	A	Pass
CN	CS8361	A	Pass
TT	OMD553	A	Pass
WT	IT8394	A	Pass

RESULT:

Thus the program using JSP and Database connectivity was successfully executed and verified.

EX NO:7 PROGRAMS USING XML – SCHEMA – XSLT/XSL

Aim:

To write a program using XML schema with XSLT/XSL.

Procedure:

Step 1: Start the Program .
Step 2: Create a root process for food.
Step 3: Create a style for XSLT with focus on each item.
Step 4: Output the items .
Step 5: Stop STRUCTURE.XML.

Source code:

index.xml

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

<?xml-stylesheet type="text/xsl" href="employee.xsl"?>

<employee id="KGiSL Institute of Technology">

<emp>

<name>Mohan</name>

<job>Professor</job>

<department>IT</department>

<salary>100000</salary>

</emp>

<emp>

<name>Siva</name>

<job>Associate Professor</job>

<department>ECE</department>

<salary>55000</salary>

</emp>

<emp>

<name>Elan</name>

<job>Associate Professor</job>
```

```
<department>EEE</department>
<salary>50000</salary>
</emp>
<emp>
<name>Baskar</name>
<job>Associate Professor</job>
<department>FT</department>
<salary>45000</salary>
</emp>
<emp>
<name>Kumar</name>
<job>Professor</job>
<department>CSE</department>
<salary>95000</salary>
</emp>
</employee>
```

Employee.xsl

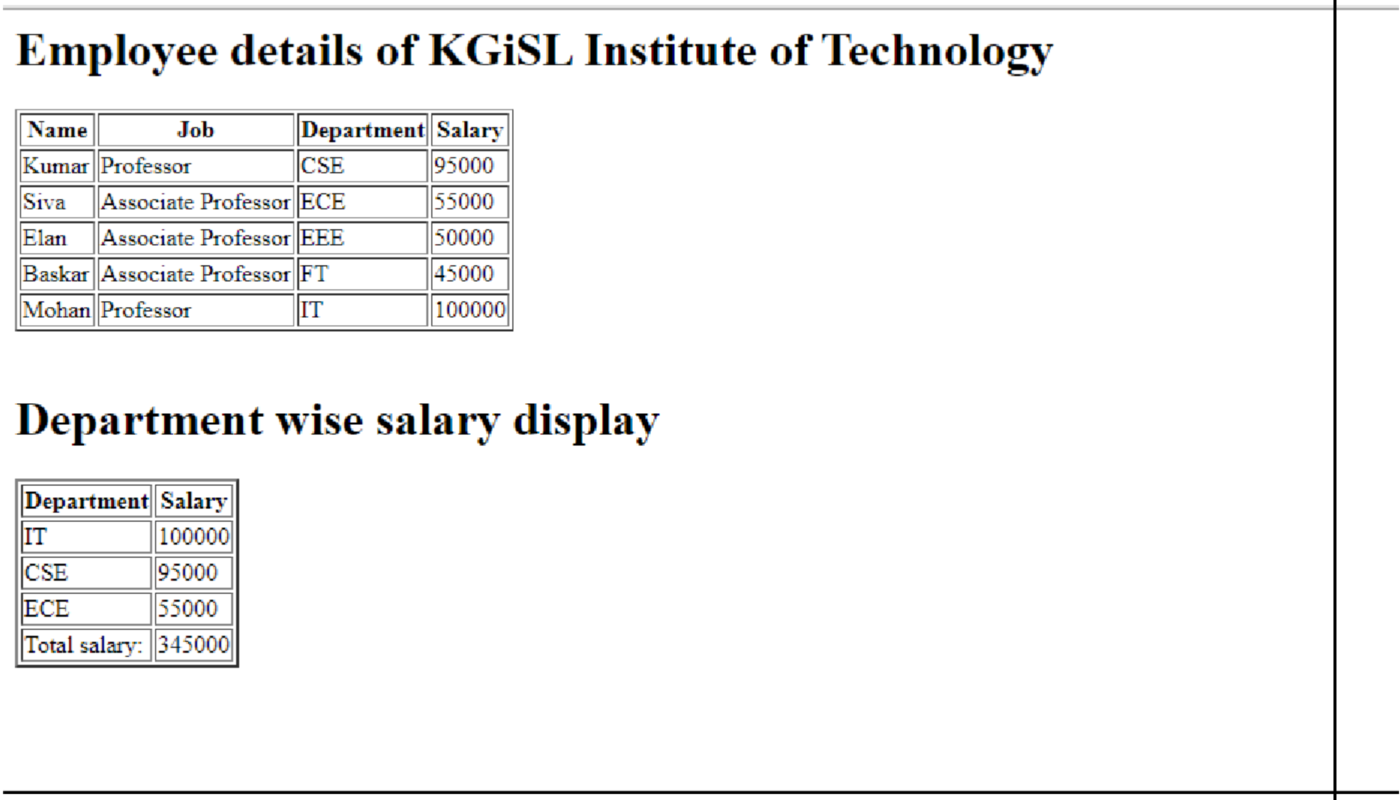
```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0">
<xsl:output method="html" omit-xml-declaration="yes"/>
<xsl:template match="/">
<html>
<head>
<title>Employee Details
<xsl:value-of select="employee/@id"/>
</title>
</head>
<body>
<h1>Employee details of <xsl:value-of select="employee/@id"/></h1>
```

```
<table border="1">
<thead>
<tr>
<th>Name</th>
<th>Job</th>
<th>Department</th>
<th>Salary</th>
</tr>
</thead>
<xsl:for-each select="employee/emp">
<xsl:sort select="department" data-type="text" order="ascending"/>
<tr>
<td><xsl:value-of select="name"/></td>
<td><xsl:value-of select="job"/></td>
<td><xsl:value-of select="department"/></td>
<td><xsl:value-of select="salary"/></td>
</tr>
</xsl:for-each>
</table>
<br/>
<h1>Department wise salary display</h1>
<table border="2">
<tr>
<th>Department</th>
<th>Salary</th>
</tr>
<tr>
<td>IT</td>
<td>
```

```
<xsl:variable name="salarycount1"
select="sum(employee/emp[department='IT']/salary)"/>
<xsl:value-of select="$salarycount1"/>
</td>
</tr>
<tr>
<td>CSE</td>
<td>
<xsl:variable name="salarycount2"
select="sum(employee/emp[department='CSE']/salary)"/>
<xsl:value-of select="$salarycount2"/>
</td>
</tr>
<tr>
<td>ECE</td>
<td>
<xsl:variable name="salarycount3"
select="sum(employee/emp[department='ECE']/salary)"/>
<xsl:value-of select="$salarycount3"/>
</td>
</tr>
<tr>
<td>Total salary:</td>
<td>
<xsl:variable name="salarycount"
select="sum(employee/emp/salary)"/>
<xsl:value-of select="$salarycount"/>
</td>
</tr>
```

```
</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>
```

OUTPUT:



Employee details of KGiSL Institute of Technology

Name	Job	Department	Salary
Kumar	Professor	CSE	95000
Siva	Associate Professor	ECE	55000
Elan	Associate Professor	EEE	50000
Baskar	Associate Professor	FT	45000
Mohan	Professor	IT	100000

Department wise salary display

Department	Salary
IT	100000
CSE	95000
ECE	55000
Total salary:	345000

RESULT:

Thus the program using XML schema with XSLT/XSL was successfully executed and verified.

EX NO:8A PROGRAM USING DOM WITH XML

Aim:

To write a program using DOM along with XML file.

Procedure:

Step 1: Start the program.
Step 2: Keep the xml and DOM java file in the same folder.
Step 3: Import the required packages initially. (Download w3c-dom.jar and located in C:\Program Files\Java\jdk1.8.0_31\lib)
Step 4: Run the java file, after creating class file read the contents from the command prompt.
Step 5: Get the name of the xml file from the user to execute.
Step 6: Using XML Reader obtain the parser.
Step 7: If the document is correctly parsed then print document is well formed else print document is not well formed.

Program:

Main.java

```
import java.io.*;

import javax.xml.parsers.*;

import org.w3c.dom.*;

import org.xml.sax.*;



public class Main {

        static public void main(String[] arg) {

                try {

                        System.out.print("Enter the name of XML document ");

                        BufferedReader input = new BufferedReader(new
InputStreamReader(System.in));

                        String file_name = input.readLine();

                        File fp = new File(file_name);

                        if (fp.exists()) {
```

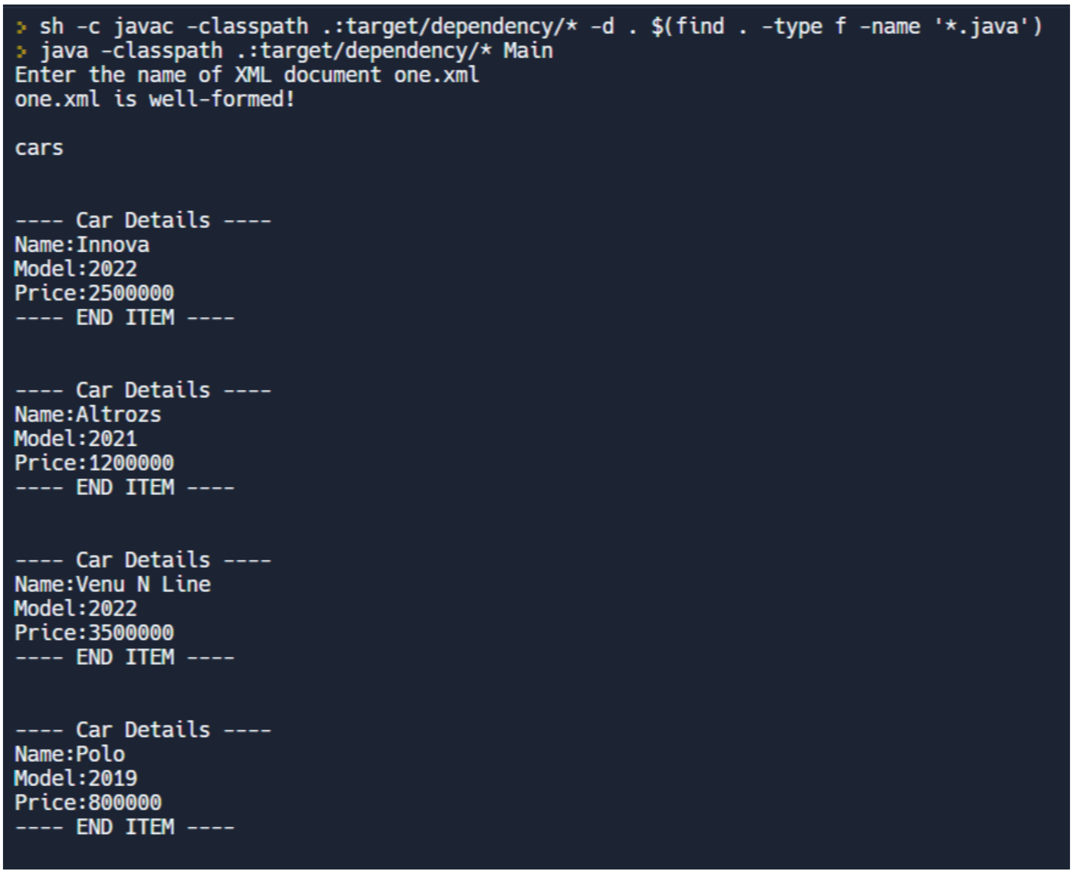
```
                try {
                    DocumentBuilderFactory Factory_obj =
DocumentBuilderFactory.newInstance();
                    DocumentBuilder builder =
Factory_obj.newDocumentBuilder();
                    InputSource ip_src = new InputSource(file_name);
                    Document doc = builder.parse(ip_src);
                    System.out.println(file_name + " is well-formed!");
                    Element root = doc.getDocumentElement();
                    System.out.println();
                    System.out.println(root.getNodeName());
                    System.out.println();
                    NodeList list = doc.getElementsByTagName("item");
                    for (int i = 0; i < list.getLength(); i++) {
                        Node node = list.item(i);
                        if (node.getNodeType() ==
Node.ELEMENT_NODE) {
                            Element element = (Element) node;
                            System.out.println("\n---- START ITEM
------");
                            System.out.println("Name:" +
element.getElementsByTagName("name").item(0).getTextContent());
                            System.out.println("Quantity:" +
element.getElementsByTagName("quantity").item(0).getTextContent());
                            System.out.println("Price:" +
element.getElementsByTagName("price").item(0).getTextContent());
                            System.out.println("---- END ITEM ----
\n");
                        }
                    }
                } catch (Exception e) {
                    System.out.println(file_name + " isn't well-formed!");
                    System.exit(1);
                }
            } else {
                System.out.print("File not found!");
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

One.xml:

```
<?xml version="1.0"?>
<menu>
    <item>
        <name>Innova</name>
        <quantity>2022</quantity>
        <price>2500000</price>
    </item>
    <item>
        <name>Altrozs</name>
        <quantity>2021</quantity>
        <price>1200000</price>
    </item>
    <item>
        <name>Venu N Line</name>
```

```
                <quantity>2022</quantity>
                <price>350000</price>
        </item>
        <item>
                <name>Polo</name>
                <quantity>2019</quantity>
                <price>800000</price>
        </item>
        <item>
                <name>Fortuner</name>
                <quantity>2015</quantity>
                <price>160000</price>
        </item>
</menu>
```

OUTPUT:



```
› sh -c javac -classpath .:target/dependency/* -d . $(find . -type f -name '*.java')
› java -classpath .:target/dependency/* Main
Enter the name of XML document one.xml
one.xml is well-formed!

cars


---- Car Details ----
Name:Innova
Model:2022
Price:2500000
---- END ITEM ----


---- Car Details ----
Name:Altrozs
Model:2021
Price:1200000
---- END ITEM ----


---- Car Details ----
Name:Venu N Line
Model:2022
Price:3500000
---- END ITEM ----


---- Car Details ----
Name:Polo
Model:2019
Price:800000
---- END ITEM ----
```

Result:

Thus the Linked list program to link the DOM and XML was compiled and executed successfully.

EX NO:8B PROGRAM USING SAX WITH XML

Aim:

To write a program using SAX along with XML file.

Procedure:

Step 1: Start the program.
Step 2: Keep the xml and SAX java file in the same folder.
Step 3: Import the required packages initially. (Download w3c-dom.jar and located in C:\Program Files\Java\jdk1.8.0_31\lib)
Step 4: Run the java file, after creating class file read the contents from the command prompt.
Step 5: Get the name of the xml file from the user to execute.
Step 6: Using XML Reader obtain the parser.
Step 7 If the document is correctly parsed then print document is well formed else print document is not well formed.

Program:

Main.java

```
import java.io.*;

import javax.xml.parsers.*;

import org.xml.sax.*;

import org.xml.sax.helpers.*;

public class Main {

        public static void main(String[] args) throws IOException {

                try {

                        System.out.print("Enter the name of XML document ");

                        BufferedReader input = new BufferedReader(new
InputStreamReader(System.in));

                        String file_name = input.readLine();

                        File fp = new File(file_name);

                        if (fp.exists()) {

                                try {
```

```
                    XMLReader reader =
XMLReaderFactory.createXMLReader();
                    reader.parse(file_name);
                    System.out.println(file_name + " is well-formed.");


                    SAXParserFactory saxParserFactory =
SAXParserFactory.newInstance();
   SAXParser saxParser = saxParserFactory.newSAXParser();
                    MenuHandler handler = new MenuHandler();


                    saxParser.parse(fp,handler);


                } catch (Exception e) {
                    System.out.println(file_name + " is not well-formed.");
                    System.exit(1);
                }
            } else {
                System.out.println("File is not present: " + file_name);
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
class MenuHandler extends DefaultHandler{
    boolean isName = false;
    boolean isQuantity = false;
    boolean isPrice = false;
```

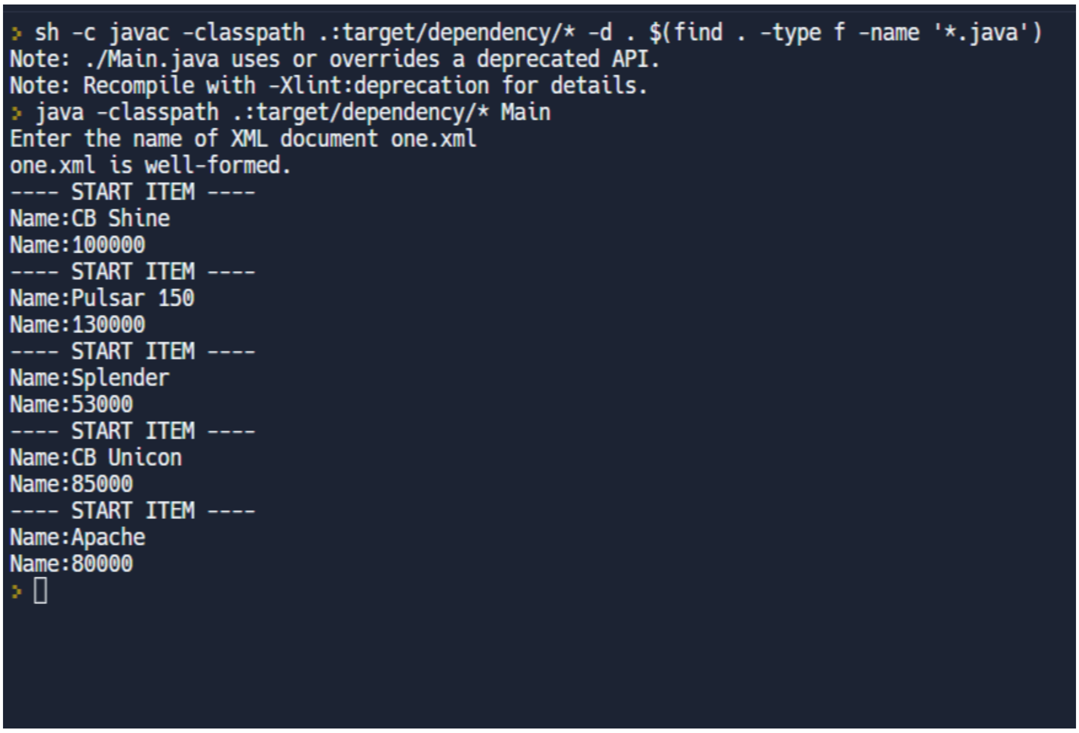
```
        }

        public void startElement(String uri, String localName, String attributeName,
    Attributes attributes) throws SAXException {

            if(attributeName.equals("item"))

                System.out.println("---- START ITEM ---- ");

            else if(attributeName.equals("name"))

                isName = true;

            else if(attributeName.equals("quantity"))

                    isQuantity = true;

            else if(attributeName.equals("price"))

                    isPrice = true;

        }


        public void endElement(String uri, String localName, String attributeName,
    Attributes attributes) throws SAXException {

            if(attributeName.equals("item")){

                System.out.println("---- END ITEM ----");

            }

        }

        public void characters(char character[], int start, int length) throws SAXException {

            if(isName){

                System.out.println("Name:"+new String(character,start,length));

                isName = false;

            }

            else if(isQuantity){

                System.out.println("Quantity:"+new String(character,start,length));

                isQuantity = false;

            }

            else if(isPrice){
```

```
                        System.out.println("Name:"+new String(character,start,length));

                        isPrice = false;

                }

        }



}
```

One.xml

```
    <?xml version="1.0"?>
    <bikes>
            <item>
                    <name>CB Shine</name>
                    <quantity>2022</quantity>
                    <price>100000</price>
            </item>
            <item>
                    <name>Pulsar 150</name>
                    <quantity>2021</quantity>
                    <price>130000</price>
            </item>
            <item>
                    <name>Splender</name>
                    <quantity>2012</quantity>
                    <price>53000</price>
            </item>
            <item>
                    <name>CB Unicon</name>
                    <quantity>2015</quantity>
                    <price>85000</price>
            </item>
            <item>
                    <name>Apache</name>
                    <quantity>2016</quantity>
                    <price>80000</price>
            </item>
    </bikes>
```

Output:



```
› sh -c javac -classpath .:target/dependency/* -d . $(find . -type f -name '*.java')
Note: ./Main.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
› java -classpath .:target/dependency/* Main
Enter the name of XML document one.xml
one.xml is well-formed.
---- START ITEM ----
Name:CB Shine
Name:100000
---- START ITEM ----
Name:Pulsar 150
Name:130000
---- START ITEM ----
Name:Splender
Name:53000
---- START ITEM ----
Name:CB Unicon
Name:85000
---- START ITEM ----
Name:Apache
Name:80000
› ▯
```

Result:

Thus the Linked list program was compiled and executed successfully

EX NO:9 PROGRAM USING AJAX

Aim:

To write programs to search and display chemistry element's definition detail using AJAX and JSP.

Procedure:

Step 1: Start the program.
Step 2: Enter the chemistry element in index.html **.**
Step 3: Read the element data by request.getParameter() on chems.jsp.
Step 4: Check given element in element list.
Step 5: Display the definition of given element.
Step 6: Stop the Program.

Program:

index.html:

```
<html>

<head>

<script type="text/javascript">

function loadXMLDoc(){

var xmlhttp;

var t1 = document.getElementsByName("t1");

var data = (t1[0].value);

console.log(data);

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function()
{
if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
{
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
}
}
xmlhttp.open("GET","chems.jsp?t1="+data,true);
xmlhttp.send(); }
</script>
</head>
<body>
<table>
<tr><td witdh=150> Enter the Chem Name : </td>
<td> <Input type=text name="t1" ></td>
 </tr>
 <tr><td witdh=150> Chem Name : </td><td> <div id="myDiv"></div></td>
 </tr>
 <tr><td> <button type="button" onclick="loadXMLDoc()">Change Content</button></td>
 </tr>
 </table>
 </body>
 </html>
```

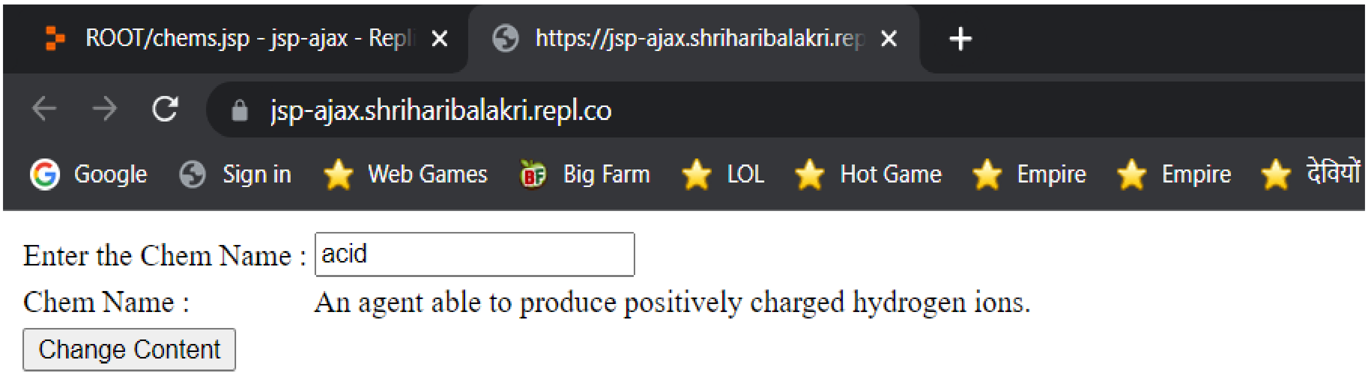
Chems.jsp

```
<%
String d=request.getParameter("t1");
String[] name;
int i=0,n=0;
```

```
name=new String[10];
name[0]="atomic number";
name[1]="catalyst";
name[2]="acid";
name[3]="base";
name[4]="bond energy";
name[5]="chain reactons";
name[6]="covalent bonds";
name[7]="element";
name[8]="enzyme";
name[9]="kinetics";
String[] defn;
defn=new String[10];
defn[0]="It is defined as the number of protons or electrons.";
defn[1]="a catalyst is a substance which fastens a reaction without themselves undergoing any change.";
defn[2]="An agent able to produce positively charged hydrogen ions.";
defn[3]="A base is a substance that can combine with a proton.";
defn[4]="The energy required to break a particular bond by hompolytic process.";
defn[5]="chain reaction:reactions which proceed by means of a set of repeating cyclic steps.";
defn[6]="Linkage of two atoms by the sharing of two electrons.";
defn[7]="a substance which cannot be further subdivided by chemical methods.";
defn[8]="a naturally occuringb substance able to catalyse a chemical reaction.";
defn[9]="The study of rate of reactions.";
for(i=0;i<9;i++)
{
if(d.equals(name[i]))
```

```
 n=i;
}
 out.println(defn[n]);
 %>
```

Output:



ROOT/chems.jsp - jsp-ajax - Repl
https://jsp-ajax.shriharibalakri.rep
jsp-ajax.shriharibalakri.repl.co
Google
Sign in
Web Games
Big Farm
LOL
Hot Game
Empire
Empire
देवियों

Enter the Chem Name : acid

Chem Name : An agent able to produce positively charged hydrogen ions.

Change Content

Result:

Thus to write programs to search and display chemistry element’s definition detail using AJAX and JSP.

EX NO:10(a) CREATION OF WEB SERVICE

Aim:

To create a web service for adding few numbers using NetBeans.

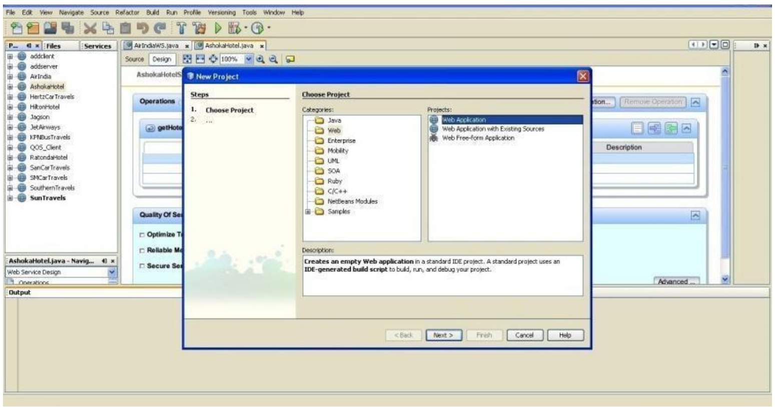
Algorithm:

1. Using the Netbeans API create a project of the type web application.
2. Create a web service in the project.
3. Click on the Design tab and design the prototype of the web service.
4. Click on source tab and modify the application logic of the web service.
5. Save the project.
6. Right click on the project and click on deploy and undeploy.
7. Then test the web service.

STEPS TO CREATE ADDITION WEB SERVICE:

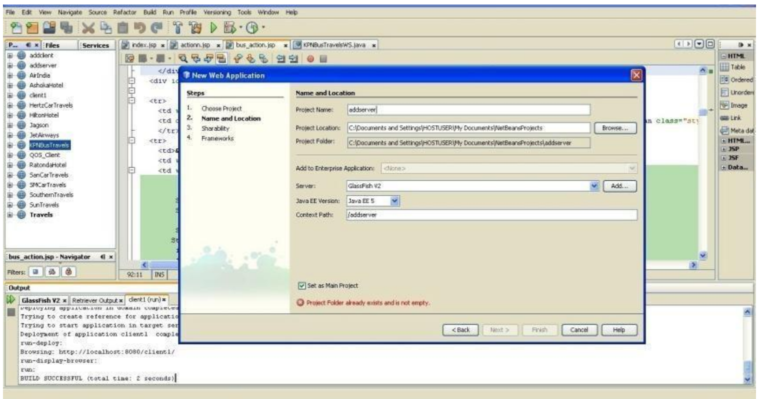
I.STEPS FOR CREATING SERVER SIDE PROJECT:

1. OPEN File->New->NewProject->Web->Web App..click next



Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.

2. Give Project name->addserver…then click finish



Project Name: addserver

Project Location: C:\Documents and Settings\HOSTUSER\My Documents\NetBeansProjects

Project Folder: C:\Documents and Settings\HOSTUSER\My Documents\NetBeansProjects\addserver

Add to Enterprise Application: <None>

Server: GlassFish V2

Java EE Version: Java EE 5

Context Path: /addserver

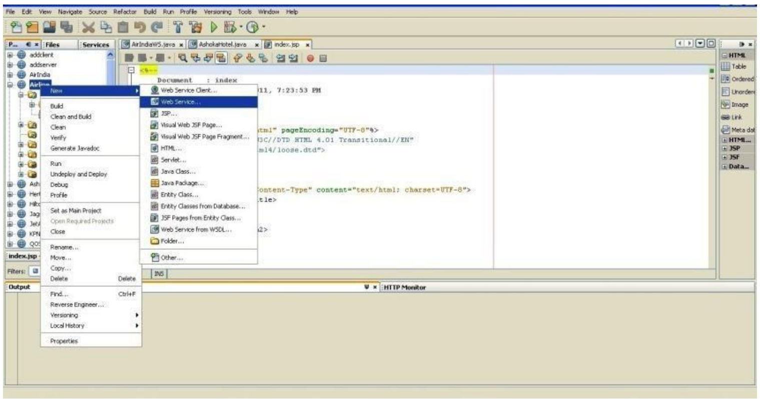
☑ Set as Main Project

Project Folder already exists and is not empty.

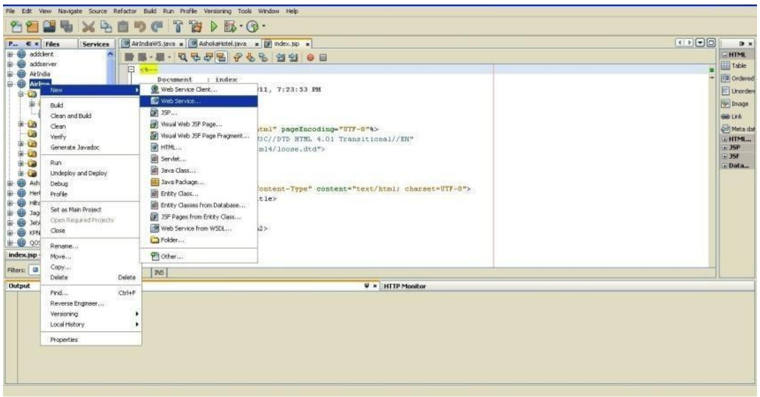
```
Trying to create reference for applicatio
Trying to start application in target ser
Deployment of application client1  comple
run-deploy:
Browsing: http://localhost:8080/client1/
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 2 seconds)
```

3. The addserver project will be created in right side.Right click it and choose the following.

Give the web service name as addweb.

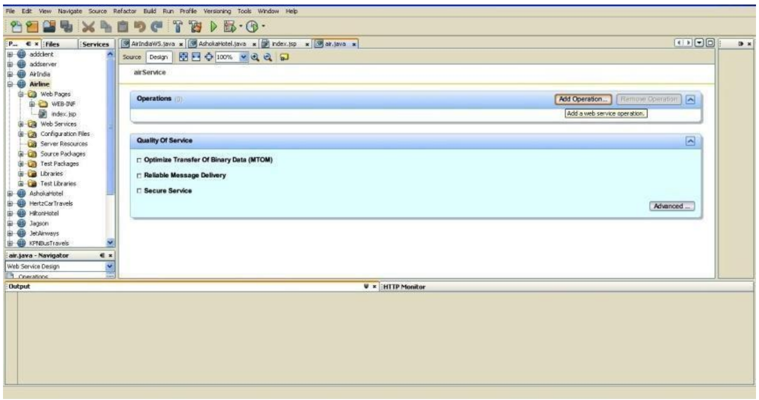


4. After this in left side ,the design window choose the add operation



5. Give the following in the opened window for creating operation

Name->add



□ Optimize Transfer Of Binary Data (MTOM)

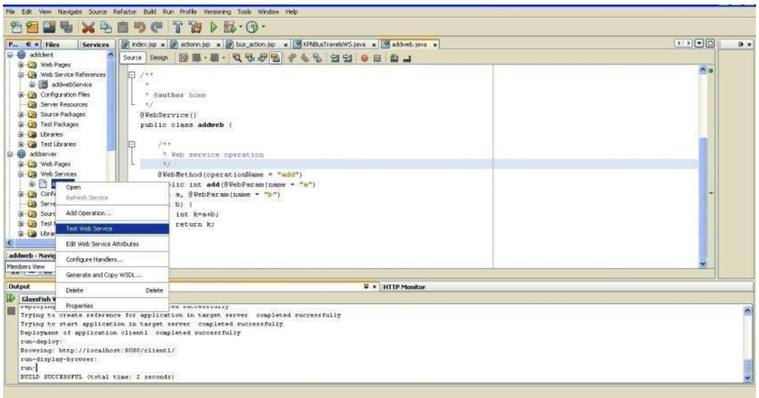
□ Reliable Message Delivery

□ Secure Service

6.Then in the source add the following code and save it.

```
package org;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
@WebService()
public class addweb {
    /**
     * Web service operation
     */
    @WebMethod(operationName = "add")
    public int add(@WebParam(name = "a")
    int a, @WebParam(name = "b")
    int b) {
        int k=a+b;
        return k;
    }
}
```

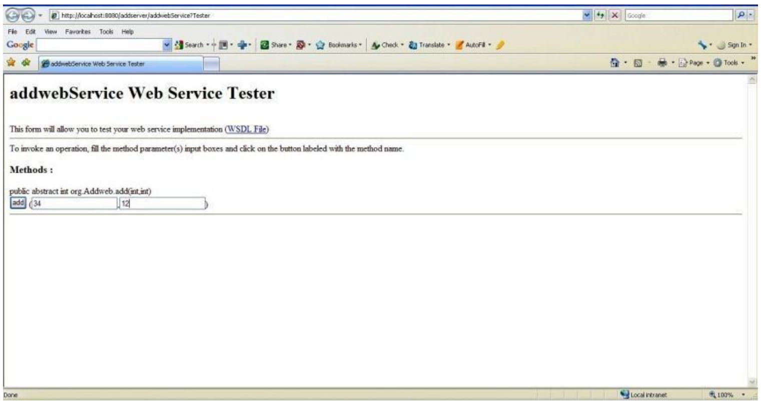
7. Then right click on add addserver and perform undeploy and deploy…after that right click on addweb and do test web service to see the SOAP request and response message.



```
/**
 *
 * @author home
 */
@WebService()
public class addweb {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "add")
    lic int add(@WebParam(name = "a")
     a, @WebParam(name = "b")
     b) {
     int k=a+b;
     return k;
```

```
Trying to create reference for application in target server  completed successfully
Trying to start application in target server  completed successfully
Deployment of application client1  completed successfully
run-deploy:
Browsing: http://localhost:8080/client1/
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 2 seconds)
```

Give some integers and click add.



http://localhost:8080/addserver/addwebService?Tester
File Edit View Favorites Tools Help
addwebService Web Service Tester
Done
Local intranet
100%

addwebService Web Service Tester

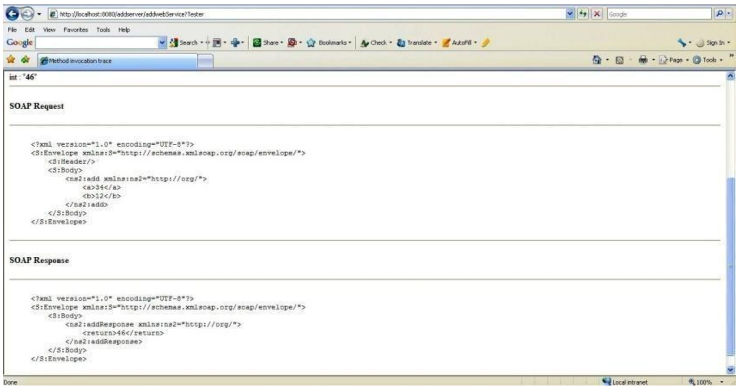
This form will allow you to test your web service implementation (WSDL File)

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

Methods :

public abstract int org.Addweb.add(int,int)

add (34 , 12)



http://localhost:8080/addserver/addwebService?Tester
File Edit View Favorites Tools Help
Method invocation trace
Done
Local intranet
100%

int : "46"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:add xmlns:ns2="http://org/">
            <a>34</a>
            <b>12</b>
        </ns2:add>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:addResponse xmlns:ns2="http://org/">
            <return>46</return>
        </ns2:addResponse>
    </S:Body>
</S:Envelope>
```

EX NO:10(b) CREATION OF WEB SERVICE CLIENT

Aim:

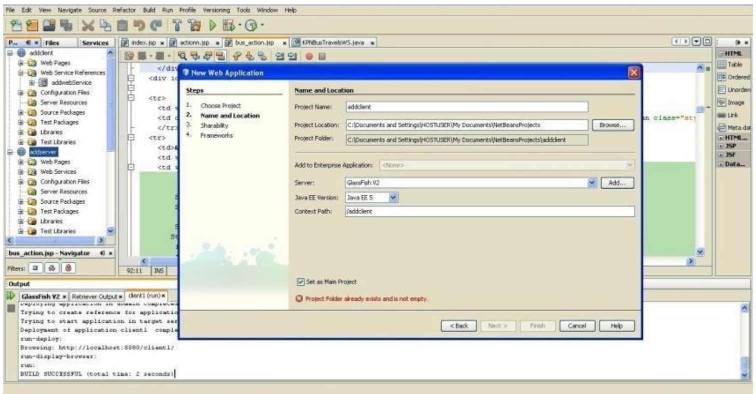
To create a web service for adding few numbers using NetBeans and write client side code to invoke the web service.

Algorithm:

1. Using the Netbeans API create a project of the type web application.
2. Create a web service in the project.
3. Click on the Design tab and design the prototype of the web service.
4. Click on source tab and modify the application logic of the web service.
5. Save the project.
6. Right click on the project and click on deploy and undeploy.
7. Then test the web service.
8. Create another web application project and create a jsp file.
9. Right click on project and click on create web service client.
10. Browse and choose the web service created i.e wsdl url
11. Drag and drop the web service reference to the source code window.
12. Then pass the appropriate parameters to the web service client and invoke the web service.

STEPS TO CREATE CLIENT SIDE PROJECT:

1. Create the new project as above and give the name as addclient.



New Web Application

Steps

1. Choose Project
2. Name and Location
3. Sharability
4. Frameworks

Name and Location

Project Name: addclient

Project Location: C:\Documents and Settings\HOSTUSER\My Documents\NetBeansProjects

Project Folder: C:\Documents and Settings\HOSTUSER\My Documents\NetBeansProjects\addclient

Add to Enterprise Application: <None>

Server: GlassFish V2

Java EE Version: Java EE 5

Context Path: /addclient

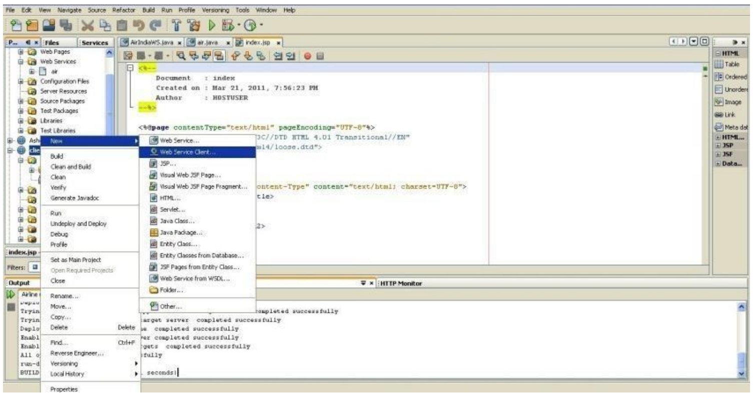
☑ Set as Main Project

Project Folder already exists and is not empty.

< Back | Next > | Finish | Cancel | Help

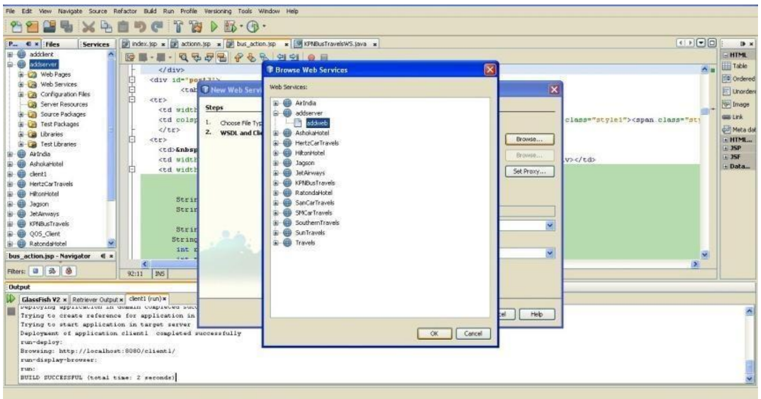
```
Deploying application in domain complete
Trying to create reference for applicatio
Trying to start application in target ser
Deployment of application client1  comple
run-deploy:
Browsing: http://localhost:8080/client1/
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 2 seconds)
```

2. addclient project will be created. right click it and choose the following.



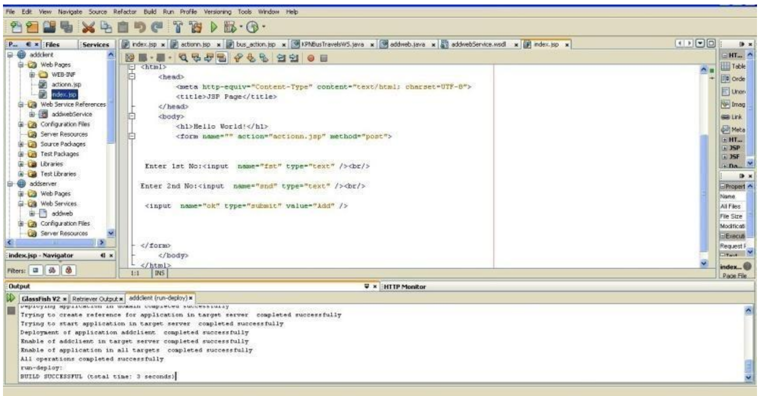
```
<%--
    Document   : index
    Created on : Mar 21, 2011, 7:56:23 PM
    Author     : HOSTUSER
--%>

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

3. Then browse and choose the addweb wsdl file



```
Trying to create reference for application in
Trying to start application in target server
Deployment of application client1  completed successfully
run-deploy:
Browsing: http://localhost:8080/client1/
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 2 seconds)
```

4. Then choose the following and add the source code in index.jsp and save it.



```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
        <form name="" action="actionn.jsp" method="post">

 Enter 1st No:<input  name="fst" type="text" /><br/>

Enter 2nd No:<input  name="snd" type="text" /><br/>

 <input  name="ok" type="submit" value="Add" />



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

```
Trying to create reference for application in target server  completed successfully
Trying to start application in target server  completed successfully
Deployment of application addclient  completed successfully
Enable of addclient in target server completed successfully
Enable of application in all targets  completed successfully
All operations completed successfully
run-deploy:
BUILD SUCCESSFUL (total time: 3 seconds)
```

Index.jsp source code:

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

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

   "http://www.w3.org/TR/html4/loose.dtd">

<html>

    <head>

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

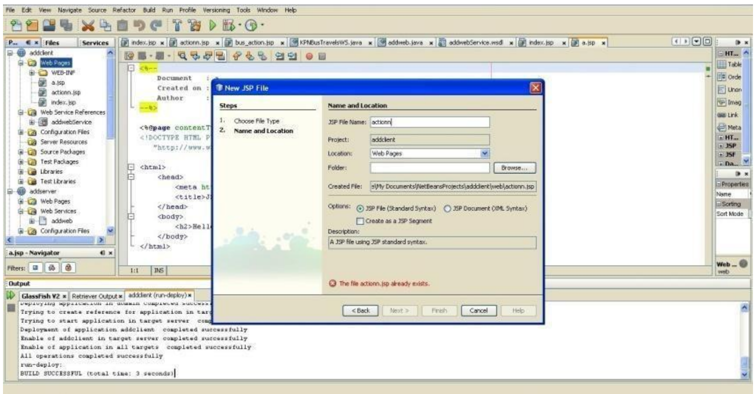
```
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
        <form name="" action="actionn.jsp" method="post">
 Enter 1st No:<input name="fst" type="text" /><br/>
Enter 2nd No:<input  name="snd" type="text" /><br/>
 <input  name="ok" type="submit" value="Add" />
</form>
    </body>
</html>
```

5. Then create an action.jsp as follows.

Right click web page in addclient and choose new->jsp

Name:action

Click finish



New JSP File

Steps
1. Choose File Type
2. Name and Location

Name and Location

JSP File Name: actionn

Project: addclient

Location: Web Pages

Folder:

Created File: e\My Documents\NetBeansProjects\addclient\web\actionn.jsp

Options: ● JSP File (Standard Syntax) ○ JSP Document (XML Syntax)

□ Create as a JSP Segment

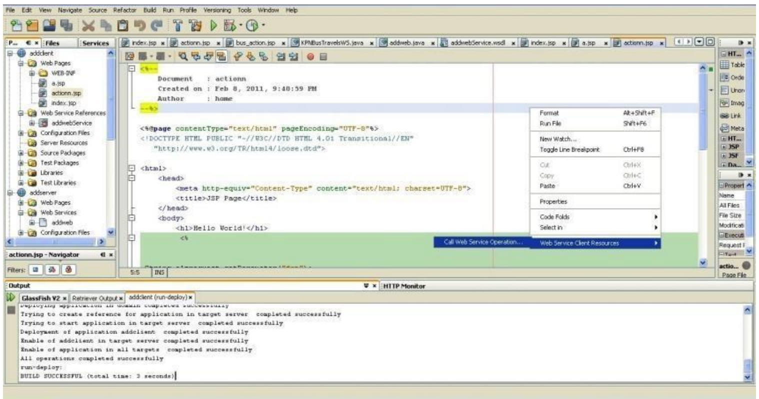
Description:

A JSP file using JSP standard syntax.

The file actionn.jsp already exists.

```
Trying to create reference for application in tar
Trying to start application in target server  com
Deployment of application addclient  completed successfully
Enable of addclient in target server completed successfully
Enable of application in all targets  completed successfully
All operations completed successfully
run-deploy:
BUILD SUCCESSFUL (total time: 3 seconds)
```

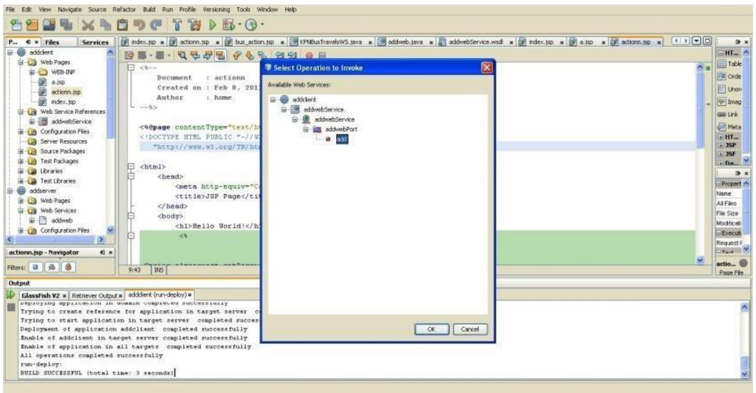
6. click on the action.jsp page..then right click in it and choose web service client reference ->call web service



```
<%--
    Document   : actionn
    Created on : Feb 8, 2011, 9:48:59 PM
    Author     : home
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
         <%
```

```
Trying to create reference for application in target server  completed successfully
Trying to start application in target server  completed successfully
Deployment of application addclient  completed successfully
Enable of addclient in target server completed successfully
Enable of application in all targets  completed successfully
All operations completed successfully
run-deploy:
BUILD SUCCESSFUL (total time: 3 seconds)
```

7. To invoke the add service.



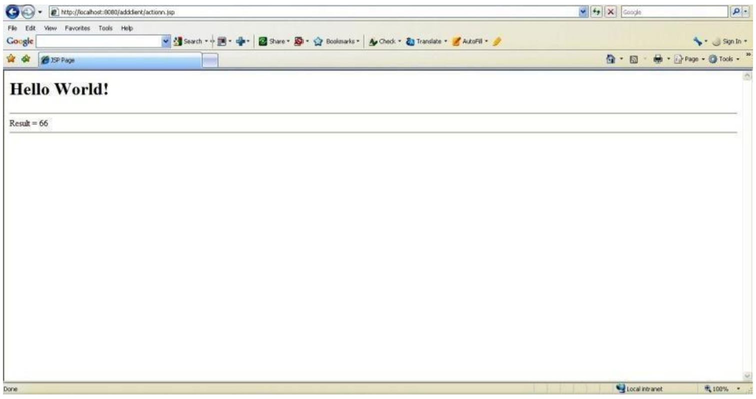
```
Deployment of application addclient  completed successfully
Enable of addclient in target server completed successfully
Enable of application in all targets  completed successfully
All operations completed successfully
run-deploy:
BUILD SUCCESSFUL (total time: 3 seconds)
```

8. add the following code in the action.jsp

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

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

   "http://www.w3.org/TR/html4/loose.dtd">

<html>

    <head>

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

        <title>JSP Page</title>

    </head>

    <body>

        <h1>Hello World!</h1>

         <%
```

```
 String  a1=request.getParameter("fst");
String b1=request.getParameter("snd");
int aa=Integer.parseInt(a1);
int bb=Integer.parseInt(b1);
  %>
    <%-- start web service invocation --%><hr/>
    <%
    try {
            org.AddwebService service = new org.AddwebService();
            org.Addweb port = service.getAddwebPort();
             // TODO initialize WS operation arguments here
            int a = aa;
            int b = bb;
            // TODO process result here
            int result = port.add(a, b);
            out.println("Result = "+result);
    } catch (Exception ex) {
            // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
    </body>
</html>
```

8. Finally undeploy and deploy the addclient and run it.



Hello World!

Result = 66

Result:

Thus the study of web services is done successfully.