INDEX
LIST OF EXPERIMENTS

EX NO 1: WRITE A HTML PROGRAM FOR CREATION OF WEBSITE WITH FORMS,

DATE: FRAMES, LINKS, TABLES ETC.

AIM

To create a HTML page with frames, links, tables and other tags.

THEORY

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language. Using this web page creation, styling etc can be done with notepad

SYNTAX

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

PROCEDURE

1. Create the HTML Page named index.html and create <iframe> tags in that page.
2. Create a new HTML Page with form elements named input, button for submit.
3. Create a link to the page with the newly created page
4. Create another HTML page with table using table properties
5. Create <iframe> tag to display the other 3 pages inside the index.html page.

PROGRAM

1. index.html

```
<div id='header'>

<span>HelloWeb</span>

</div>

<div class='container'>

<iframe src="tables.html"></iframe>

<iframe src="forms.html"></iframe>
```

```
<iframe src="links.html"></iframe>
</div>
<style>
:root{
--theme-color:dodgerblue;
}
#header
{
background:var(--theme-color);
color:white;
top:0;
left:0;
height:50px;
width:100%;
position:fixed;
z-index:2;
text-align:center;
}
#header span{
top:10px;
font-size:22px;
position:relative;
}
.container
{
margin-top:100px;
}
iframe{
width:30%;
height:80%;
}
</style>
```

2. forms.html

```
<form>
```

```
<h1>Join our Community</h1>
<input type='email' placeholder='Enter Email...' />
<input type='text' placeholder='Enter Name...' />
<input type='password' placeholder='Enter Password...' />
<input type='password' placeholder='Confirm Password...' /><br>
<select>
<option>Male</option>
<option>Female</option>
<option>Other</option>
</select><br>
<input type='checkbox' /> Remember me
<br>
<input type='submit' value='submit' />
</form>
<style>
input,select{
border:1px solid grey;
font-size:20px;
padding:10px;
margin:10px;
}
</style>
```

3. links.html

```
<img src='bg.jpg' id='img' />
<a href='https://www.facebook.com'><img src='facebook.png' id='logo' /> Facebook</a><br>
<a href='https://www.instagram.com'><img src='instagram.png' id='logo' />
Instagram</a><br>
<a href='https://www.vkontake.com'><img src='vk.png' id='logo' /> VKontakte</a><br>
<style>
a{
font-size:30px;
text-decoration:none;
}
#img{
```

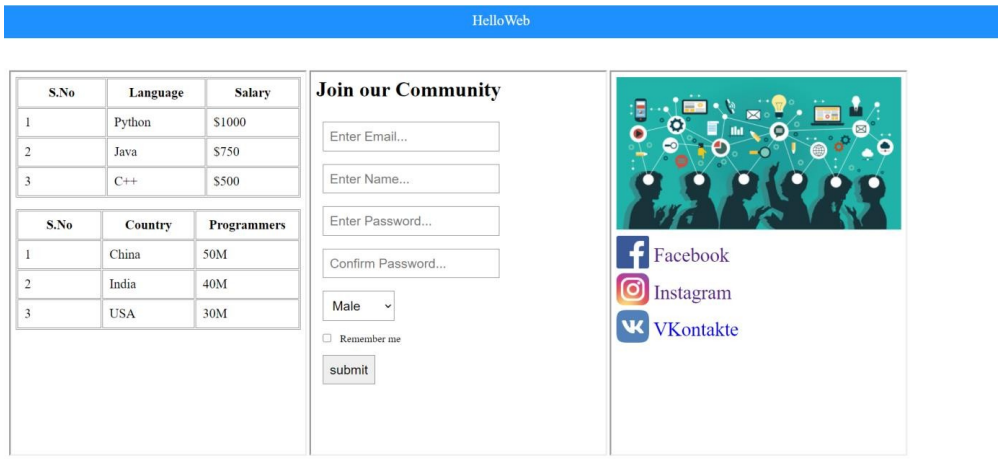
```
max-width:100%;
}
#logo{
width:50px;
position:relative;
top:10px;
}
</style>
```

4. tables.html

```
<table>
<tr>
<th>S.No</th>
<th>Language</th>
<th>Salary</th>
</tr>
<tr>
<td>1</td>
<td>Python</td>
<td>$1000</td>
</tr>
<tr>
<td>2</td>
<td>Java</td>
<td>$750</td>
</tr>
<tr>
<td>3</td>
<td>C++</td>
<td>$500</td>
</tr>
</table>
<br>
<table>
<tr>
```

```
<th>S.No</th>
<th>Country</th>
<th>Programmers</th>
</tr>
<tr>
<td>1</td>
<td>China</td>
<td>50M</td>
</tr>
<tr>
<td>2</td>
<td>India</td>
<td>40M</td>
</tr>
<tr>
<td>3</td>
<td>USA</td>
<td>30M</td>
</tr>
</table>
<style>
table,td,th{
border:1px solid grey;
}
td,th{
width:150px;
padding:10px;
font-size:20px;
}
</style>
```

OUTPUT



RESULT

Thus the HTML page with frames, links, tables and other tags are created successfully.

EXNO 2: **DESIGN A WEBPAGE USING CSS3**

DATE :

AIM

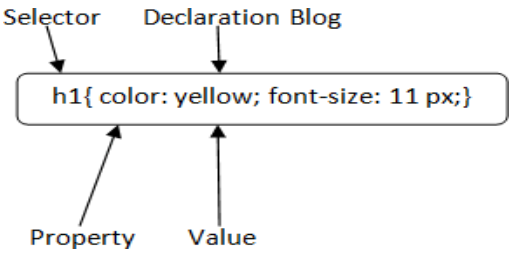
To design a web page using CSS3.

THEORY

CSS3 is used to control the style of a web document in a simple and easy way. CSS3 is the acronym for "Cascading Style Sheet". CSS can create stunning web sites with different styles, color for the font .it helps in styling the html document. It is a widely used language on the web.

SYNTAX

A CSS rule set contains a selector and a declaration block.




```
<link rel='stylesheet' href='style.css' />
<div class='header'>
  <h1 id='title'><b>Survey Form</b> - Programmers Salary</h1>
</div>
<div class='container'>
```

PROCEDURE

1. Create a html page.
2. Design the different type of CSS selectors to style the created html page
3. Merge a CSS stylesheet together.
4. Create the design to style the page
6. Customize the website with the options available in HTML and CSS
7. Finally add the content and images of the website and display it

PROGRAM

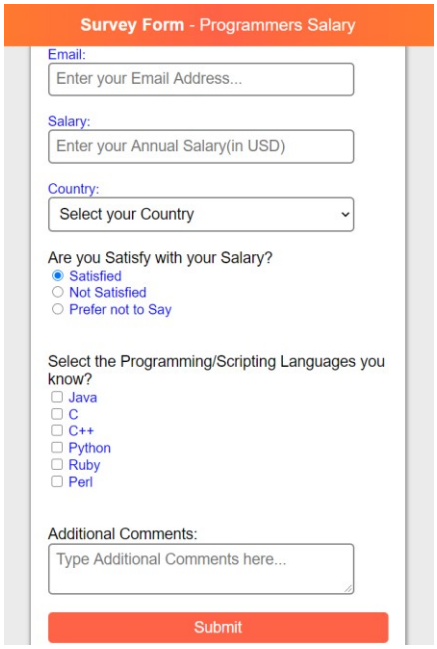
```
<link rel='stylesheet' href='style.css' />
<div class='header'>
  <h1 id='title'><b>Survey Form</b> - Programmers Salary</h1>
```

```
</div>
<div class='container'>
<p id='description'>This is a Simple Survey form for Programmers Salary.</p>
<center>
<form id='survey-form'>
<label for='name' id='name-label'>Name:</label>
<input type='text' id='name' placeholder='Enter your Name...' required /><br><br>
<label for='email' id='email-label'>Email:</label>
<input type='email' id='email' placeholder='Enter your Email Address...' required /><br><br>
<label for='number' id='number-label'>Salary:</label>
<input type='number' id='number' min='1000' max='10000' placeholder='Enter your Annual
Salary(in USD)'required/><br><br>
<label for='dropdown' id='dropdown-label'>Country:<br>
<select id='dropdown' required >
<option value=''>Select your Country</option>
<option value='us'>United States</option>
<option value='uk'>United kingdom</option>
<option value='in'>India</option>
</select>
</label>
<br><br>
<div id='radio-group'>Are you Satisfy with your Salary?<br>
<label><input type='radio' name='satisfaction' checked/> Satisfied</label><br>
<label><input type='radio' name='satisfaction' /> Not Satisfied</label><br>
<label><input type='radio' name='satisfaction' /> Prefer not to Say</label><br>
</div>
<br><br>
<div id='checkbox-group'>Select the Programming/Scripting Languages you know?<br>
<label><input type='checkbox' name='lang' value='java'/> Java</label><br>
<label><input type='checkbox' name='lang' value='c' /> C</label><br>
<label><input type='checkbox' name='lang' value='cpp'/> C++</label><br>
<label><input type='checkbox' name='lang' value='python' /> Python</label><br>
<label><input type='checkbox' name='lang' value='ruby' /> Ruby</label><br>
<label><input type='checkbox' name='lang' value='perl'/> Perl</label><br>
```

```
</div>
<br><br>
Additional Comments:<br>
<textarea placeholder='Type Additional Comments here...'></textarea>
<br><br>
<button id='submit' type='submit' style='color:white;background-
color:tomato;width:100%;border:none;'>Submit</button>
<br><br>
</form>
</center>
</div>
<style>
#description{
color:tomato;
}
#survey-form{
max-width:400px;
padding:20px;
border-radius:20px;
border:1px solid grey;
box-shadow:1px 2px 5px grey;
background-color:white;
text-align:left;
font-size:18px;
}
input,button,textarea,select{
font-size:18px;
border-radius:5px;
padding:7.5px;
border:2px solid grey;
outline:none;
font-family:Arial;
}
#name,#email,#number,textarea,select{
```

```
width:90%;
}
#name:focus,#email:focus,#number:focus,textarea:focus{
border-color:tomato;
}
label{
color:blue;
font-size:16px;
}
</style>
```

OUTPUT



RESULT

Thus, web page using CSS3 was designed and executed successfully.

EX NO 3: A. CREATE A SCRIPT THAT ASKS THE USER FOR THE NAME, THEN
DATE: GREETS THE USER WITH 'HELLO' AND THE USER NAME ON THE PAGE

AIM

To create a script that asks the user for the name, then greets the user with 'Hello' and the user name on the page.

THEORY

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. JavaScript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience. JavaScript usage has now extended to mobile app development, desktop app development, and game development.

SYNTAX

```
<script ...>
var name=document.getElementById('text').value;
alert("Hello, "+name);
document.getElementById('greet_paragraph').innerHTML="Hello, "+name;
</script>
```

PROCEDURE

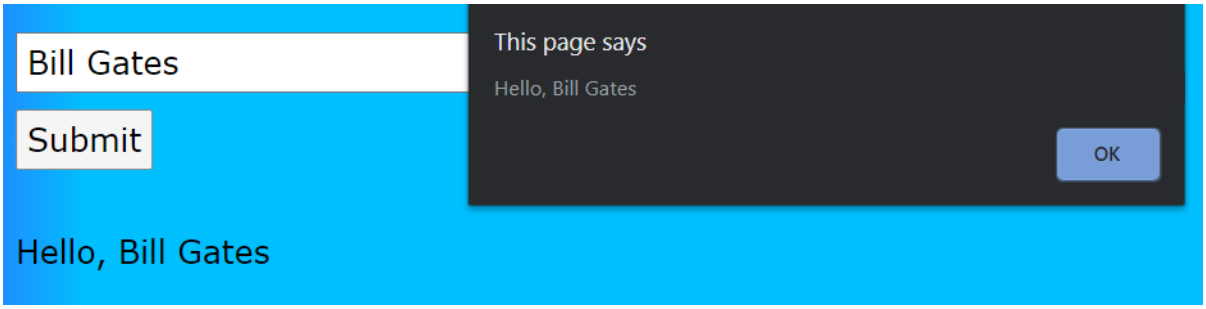
1. Create an html page giving the options as input, paragraph and button on the page.
2. Add the CSS for the input and button tags.
3. Create the JavaScript function inside the html code.
4. When button is clicked, it gets the input value and print the greet message in the paragraph tag.

PROGRAM

```
<script>
function greet(){
        var name=document.getElementById('text').value;
        alert("Hello, "+name);
        document.getElementById('greet_paragraph').innerHTML="Hello, "+name;
}
</script>
<input type='text' id='text' placeholder='Enter the Name...' /><br>
<button onclick='greet()' id='submit'>Submit</button><br><br>
<p id='greet_paragraph'></p>
<style>
html{
```

```
background:linear-gradient(to right,dodgerblue 0%,deepskyblue 5%);
}
p{
font-family:Verdana;
font-size:20px;
}
input,button{
margin-top:10px;
padding:5px;
font-family:Verdana;
font-size:20px;
}
</style>
```

OUTPUT



EXNO 3: B. CREATE A SCRIPT THAT COLLECTS NUMBERS FROM A PAGE AND THEN ADDS THEM UP AND PRINTS THEM TO A BLANK FIELD ON THE PAGE

AIM

To create a script that collects numbers from a page and then adds them up and prints them to a blank field on the page.

PROCEDURE

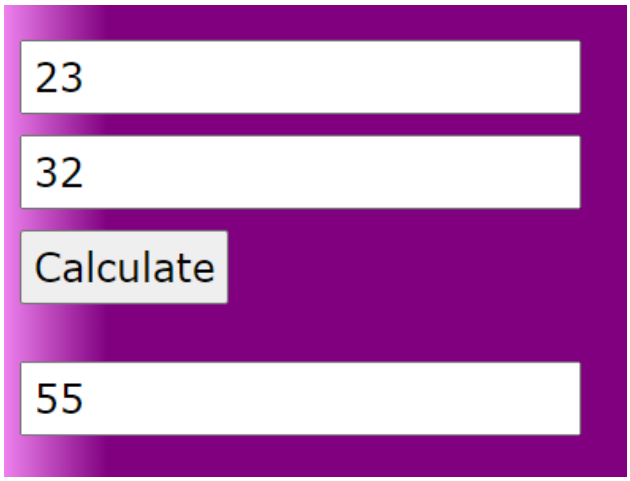
1. Create a HTML page giving three inputs and a button on the page.
2. Add CSS for the input and button tags.
3. Create the JavaScript function inside the html page.
4. When the button is clicked, the input value from the first two input boxes will be added.

5. Finally, it prints the result value in the third input box.

PROGRAM

```
<script>
function add()
{
let num1=document.getElementById('num1').value;
let num2=document.getElementById('num2').value;
num1=parseInt(num1);
num2=parseInt(num2);
const sum=num1+num2;
document.getElementById('answer').value=sum;
}
</script>
<input type='text' id='num1' placeholder='Enter the first number...' /><br>
<input type='text' id='num2' placeholder='Enter the second number...' /><br>
<button onclick='add()' id='submit'>Calculate</button><br><br>
<input type='text' id='answer' placeholder='Answer...' readonly /><br>
<style>
html{
background:linear-gradient(to right,dodgerblue 0%,deepskyblue 5%);
}
P
{
font-family:Verdana;
font-size:20px;
}
input,button{
margin-top:10px;
padding:5px;
font-family:Verdana;
font-size:20px;
}
</style>
```

OUTPUT



RESULT

Thus we have created a script that asks the user for the name, then greets the user with 'Hello' and the user name on the page and a script that collects numbers from a page and then adds them up and prints them to a blank field on the page successfully.

EX NO 4: A. CREATE A SCRIPT THAT PROMPTS THE USER FOR A NUMBER THEN

DATE: COUNTS FROM 1 TO THAT NUMBER DISPLAYING ONLY THE ODD NUMBERS

AIM

To create a script that prompts the user for a number and then counts from 1 to that number displaying only the odd numbers.

PROCEDURE

1. Create a HTML page with input box and button
2. Create a JavaScript function that prompts user for a number and call the function in the JavaScript inside the html page
3. If the number is not equal to zero, then print the number is odd number else
4. Number is not a odd number

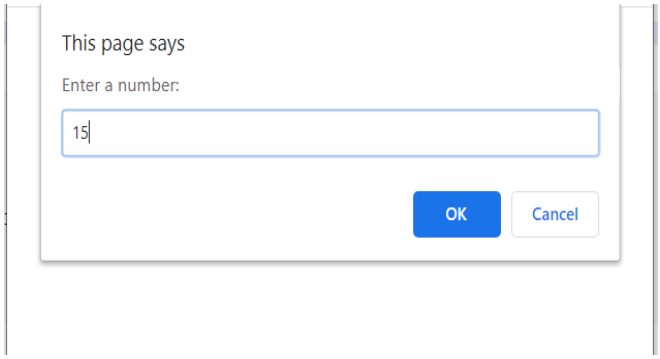
PROGRAM

```
<html>
<body>
<p id='answer'></p>
</body>
<head>
<script>
function displayOdd()
{
const number= prompt("Enter a number:");
for(let i=0;i<=number;i++)
{
if(i%2!=0)
{
document.getElementById('answer').innerHTML+="<br>"+i;
}
}
}
displayOdd();
</script>
</head>
</html>
```

<style>

```
html{
background:linear-gradient(to right,orange 0%,tomato 50%,orange 100%);
}
p{
font-family:Verdana;
font-size:20px;
}
input,button{
margin-top:10px;
padding:5px;
font-family:Verdana;
font-size:20px;
}
</style>
```

OUTPUT



EXNO 4: B. WRITE A JAVASCRIPT PROGRAM FOR PALINDROME

AIM

To write a JavaScript program for palindrome.

PROCEDURE

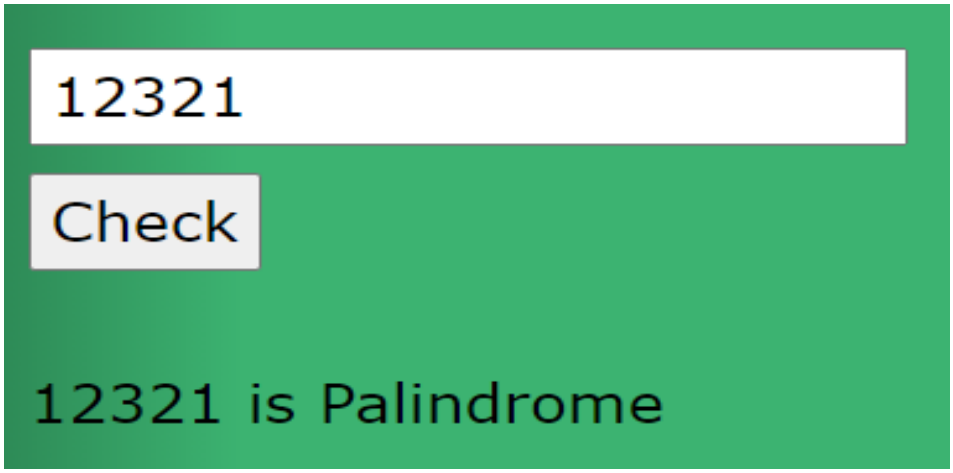
1. Create a HTML page with the input box and a button on the page.
2. Add CSS for the input and button tags.
3. Create the JavaScript function to create a palindrome
4. When button is clicked, input value is called in JavaScript function
5. If both numbers are matched, then prints number is palindrome.
6. Else prints that the number is not a palindrome.

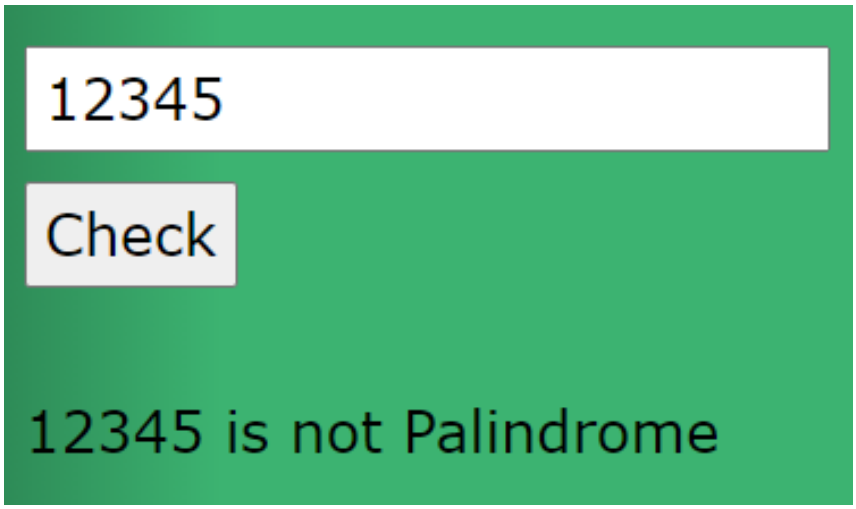
PROGRAM

```
<html>
<head>
<script>
function Palindrome()
{
var remainder, temp, value = 0;
var number = Number(document.getElementById("number").value);
temp = number;
while(number>0)
{
remainder = number%10;
number = parseInt(number/10);
value = value*10+remainder;
}
if(value==temp)
{
document.getElementById('answer').innerHTML=temp+" is Palindrome";
}
else
{
document.getElementById('answer').innerHTML=temp+" is not Palindrome";
}
}
</script>
```

```
</head>
<body>
<input type="text" id = "number" placeholder='Enter the Number...'/><br>
<button onClick="Palindrome()">Check</button><br><br>
<p id='answer'></p>
</body>
</html>
<style>
html{
background:linear-gradient(to right,seagreen 0%,mediumseagreen 5%);
}
p{
font-family:Verdana;
font-size:20px;
}
input,button{
margin-top:10px;
padding:5px;
font-family:Verdana;
font-size:20px;
}
</style>
```

OUTPUT





RESULT

Thus we have created a script that prompts the user for a number and then counts from 1 to that number displaying only the odd numbers and a javascript program for palindrome successfully.

EX NO 5: DEVELOP AND DEMONSTRATE, A XHTML DOCUMENT USING JAVASCRIPT

DATE:

SCENARIO

That collects the roll no (the valid format is: two digit from 0 and 9 followed by two upper-case characters followed by two digits; no embedded spaces are allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.

AIM

To develop and demonstrate a XHTML document using JavaScript

THEORY

XHTML stands for Extensible Hypertext Markup Language.XHTML is a stricter, more XML-based version of HTML .XHTML is HTML defined as an XML application.it is supported by all major browsers. XHTML was developed to make HTML more extensible and flexible to work with other data formats (such as XML)

SYNTAX

```
function validate()
{
var pattern=/[0-9][0-9][A-Z][A-Z][0-9][0-9]$/;
rollno = document.getElementById('rollno');
if( !rollno.value.match(pattern))
alert("Error : Invalid Roll No");
else
alert("Success...");
}
```

PROCEDURE

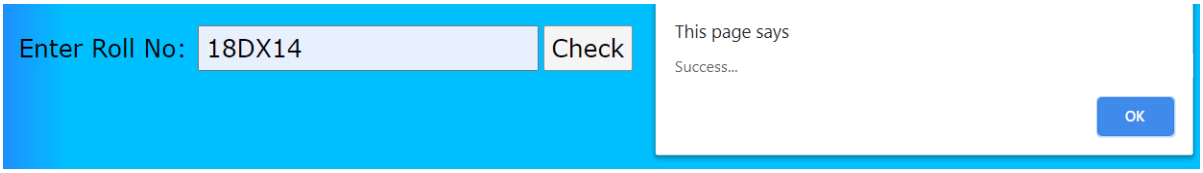
1. Create a XHTML page with input box and submit button.
2. Create a JavaScript function to validate the XHTML form.
3. The user can enter the roll to check it is valid
4. The entered text is matched with the roll no, then it gives alert message as success
5. Else it gives the alerts message as invalid roll no

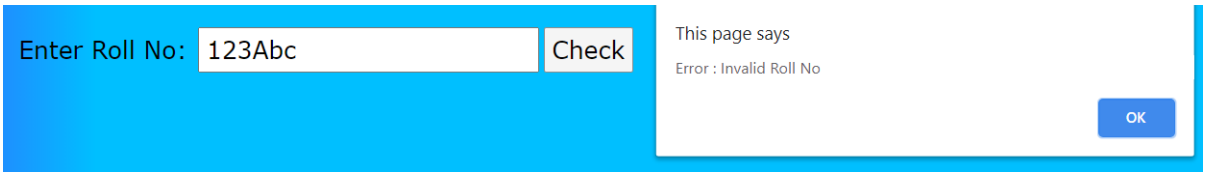
PROGRAM

```
<html>
<head>
<title>5 A. Web Technology Lab</title>
</head>
<script type="text/javascript">
function validate()
{
```

```
var pattern=/[0-9][0-9][A-Z][A-Z][0-9][0-9]$/;
rollno = document.getElementById('rollno');
if( !rollno.value.match(pattern))
alert("Error : Invalid Roll No");
else
alert("Success...");
}
</script>
<body>
<form>
<span>Enter Roll No:</span> <input type="text" id="rollno" />
<input type="submit" value="Check" onclick="validate()"/>
</form>
</body>
</html>
<style>
html{
background:linear-gradient(to right, dodgerblue 0%,deepskyblue 100%);
}
input,button,span{
margin-top:10px;
padding:5px;
font-family:Verdana;
font-size:20px;
}
</style>
```

OUTPUT





RESULT

Thus we have developed and demonstrate, using JavaScript script, a HTML document that collects the roll no (the valid format is: two digit from 0 and 9 followed by two upper-case characters followed by two digits; no embedded spaces are allowed) of the user. event handler must be included for the form element that collects this information to validate the input. messages in the alert windows must be produced when errors are detected successfully.

EXNO 6: A. DESIGN AN XML DOCUMENT TO STORE INFORMATION ABOUT ASTUDENT
DATE : IN A POLYTECHNIC COLLEGE.

SCENARIO

The information must include 100 roll no, name, name of the college, branch, year of joining, and email id. make up sample data for 3 students. create a CSS style sheet and use it to display the document.

AIM

To design an xml document to store information about a student in a polytechnic college..

THEORY

XML stands for extensible Markup Language.it was designed to store and transport data.it was also designed to be both human- and machine-readable. It has sender information. It has receiver information, heading and a message body. XML is a mark up language much like HTML

SYNTAX

```
<root>
<child>
<subchild>     </subchild>
 </child>
 </root>
```

PROCEDURE

1. Create an XML Document which contains 3 students information such as roll no, name, name of the college, branch, year of joining and email ID.
2. Create a CSS stylesheet to apply styles for the XML Document.
3. Save the file as 6a.XML extension in any folder
4. Run the XML Document in the browser.

PROGRAM

6a.xml

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

<?xml-stylesheet type = "text/css" href = "6a.css"?>

<students>

 <student>

 <rollno> 18DX01 </rollno>

 <name> Bill </name>

 <college> PSGPTC </college>

 <branch> DCE</branch>

 <YOJ> 2018 </YOJ>
```

```
<email> bill@gmail.com </email>
</student>
<student>
<rollno> 18DX02</rollno>
<name> John </name>
<college> CIT </college>
<branch> BE (CSE) </branch>
<YOJ> 2016 </YOJ>
<email> john@yahoo.com </email>
</student>
<student>
<rollno> 18DX14 </rollno>
<name> Jack </name>
<college> KCT </college>
<branch> B.TECH (IT)</branch>
<YOJ> 2017 </YOJ>
<email> jack@hotmail.com </email>
</student>
</students>
```

6a.css

```
student{
font-family: verdana;
margin-left: 10px;
background-color:white;
color:tomato;
width:100%;
font-size: 20px;
}
rollno {
color:red;
}
name {
color:blue;
```

```
font-weight: bold;
}
college {
color:pink;
font-weight: bold;
}
branch {
color:red;
}
YOJ {
padding: 0.5px;
}
email {
color:green;
font-weight: bold;
}
```

OUTPUT

18DX01 Bill PSGPTC DCE 2018 bill@gmail.com
18DX02 John CIT BE (CSE) 2016 john@yahoo.com
18DX14 Jack KCT B.TECH (IT) 2017 jack@hotmail.com

EX NO 6: B. CREATE AN XSLT STYLE SHEET FOR ONE STUDENT ELEMENT OF THE ABOVE DOCUMENT AND USE IT TO CREATE A DISPLAY OF THE ELEMENT

AIM

To create an xslt style sheet for one student element of the above document and use it to create a display of the element.

SYNTAX

```
<?xml version="1.0"?>
<xsl:stylesheet version = "1.0" xmlns:xsl = "http://www.w3.org/1999/XSL/Transform" xmlns = "http://www.w3.org/1999/xhtml">
```

PROCEDURE

1. Create a XSLT style sheet to apply styles and values for one student in the XML Document.
2. Apply the xsl styles in the xml document
3. Save the file in sample.xml
4. Run the file sample.xml in a browser.

PROGRAM

6b.xml

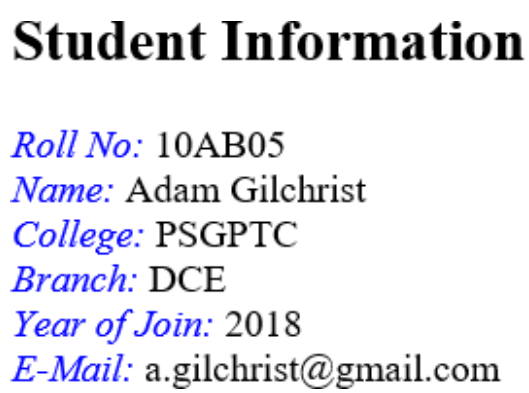
```
<?xml version = "1.0" encoding='UTF-8'?>
<?xml-stylesheet type = "text/xsl" href = "6b.xsl" ?>
<student>
 <rollno> 10AB05 </rollno>
 <name> Adam Gilchrist </name>
 <college> PSGPTC </college>
 <branch> DCE </branch>
 <YOJ> 2018 </YOJ>
 <email> a.gilchrist@gmail.com </email>
</student>
```

6b.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version = "1.0" xmlns:xsl = "http://www.w3.org/1999/XSL/Transform" xmlns =
"http://www.w3.org/1999/xhtml">
<xsl:template match = "student">
<html><head><title> Student Info </title>
 </head><body>
 <h2> Student Information </h2>
 <span style = "font-style: italic; color: blue;"> Roll No:
 </span>
 <xsl:value-of select = "rollno" /> <br />
 <span style = "font-style: italic; color: blue;"> Name:
 </span>
 <xsl:value-of select = "name" /> <br />
 <span style = "font-style: italic; color: blue;"> College:
 </span>
 <xsl:value-of select = "college" /> <br />
 <span style = "font-style: italic; color: blue;"> Branch:
```

```
</span>
<xsl:value-of select = "branch" /> <br />
<span style = "font-style: italic; color: blue;"> Year of Join:
</span>
<xsl:value-of select = "YOJ" /> <br />
<span style = "font-style: italic; color: blue;"> E-Mail:
</span>
<xsl:value-of select = "email" /> <br />
</body></html>
</xsl:template>
</xsl:stylesheet>
```

OUTPUT



RESULT

To design an xml document to store information about a student in a polytechnic college. the information must include 100 roll no, name, name of the college, branch, year of joining, and email id. make up sample data for 3 students. create a CSS style sheet and use it to display the document and to create an xslt style sheet for one student element of the above document and use it to create a display of the element.

EX NO 7: INSTALL AND CONFIGURE TOMCAT AND APACHE WEB SERVER.
DATE:

AIM

To install and configure tom cat and apache web server.

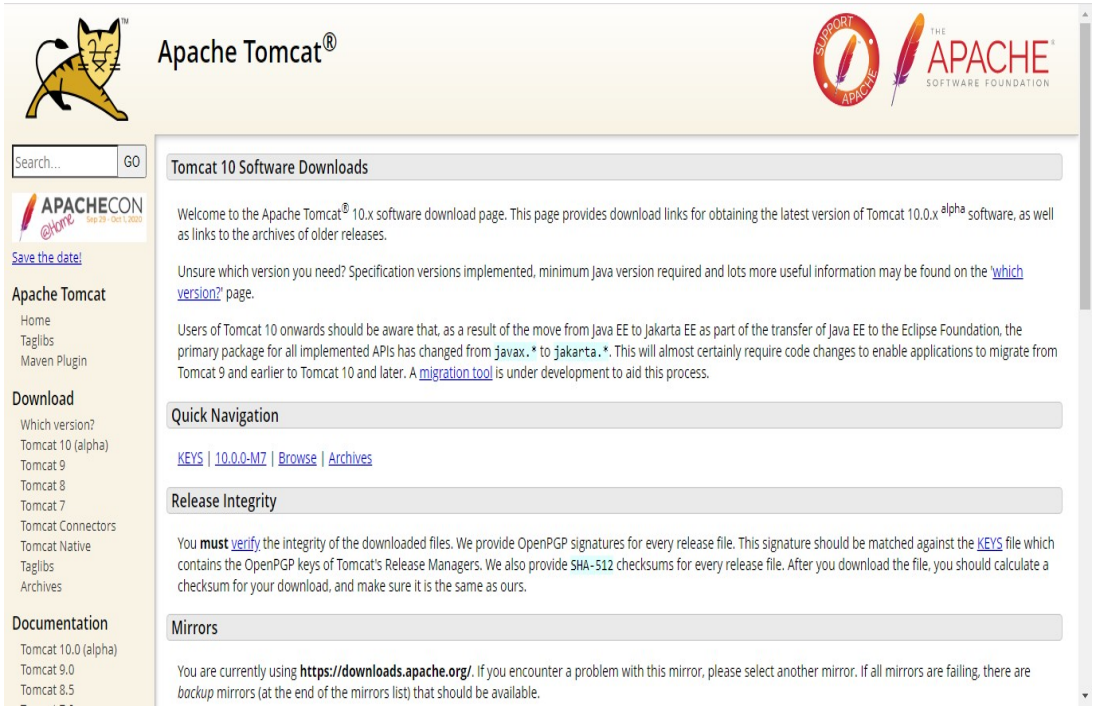
THEORY

Apache is a remarkable piece of application software. It is the most widely used Web Server application in the world with more than 50% share in the commercial web server market. Apache is the most widely used Web Server application in Unix-like operating systems but can be used on almost all platforms such as Windows, OS X, OS/2.

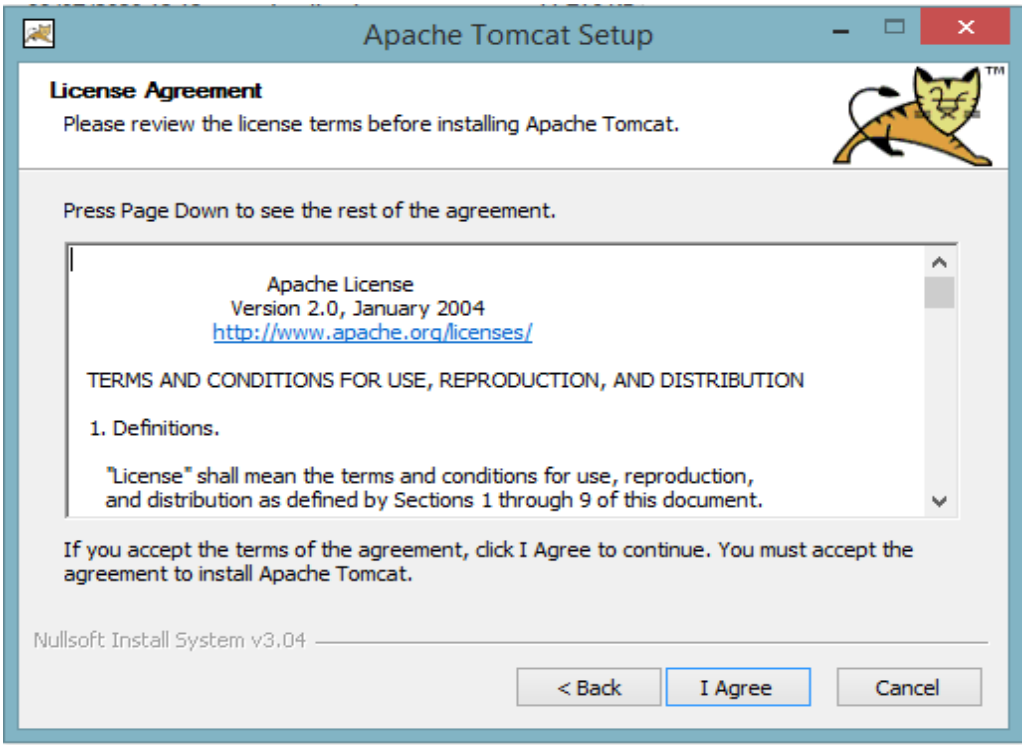
PROCEDURE

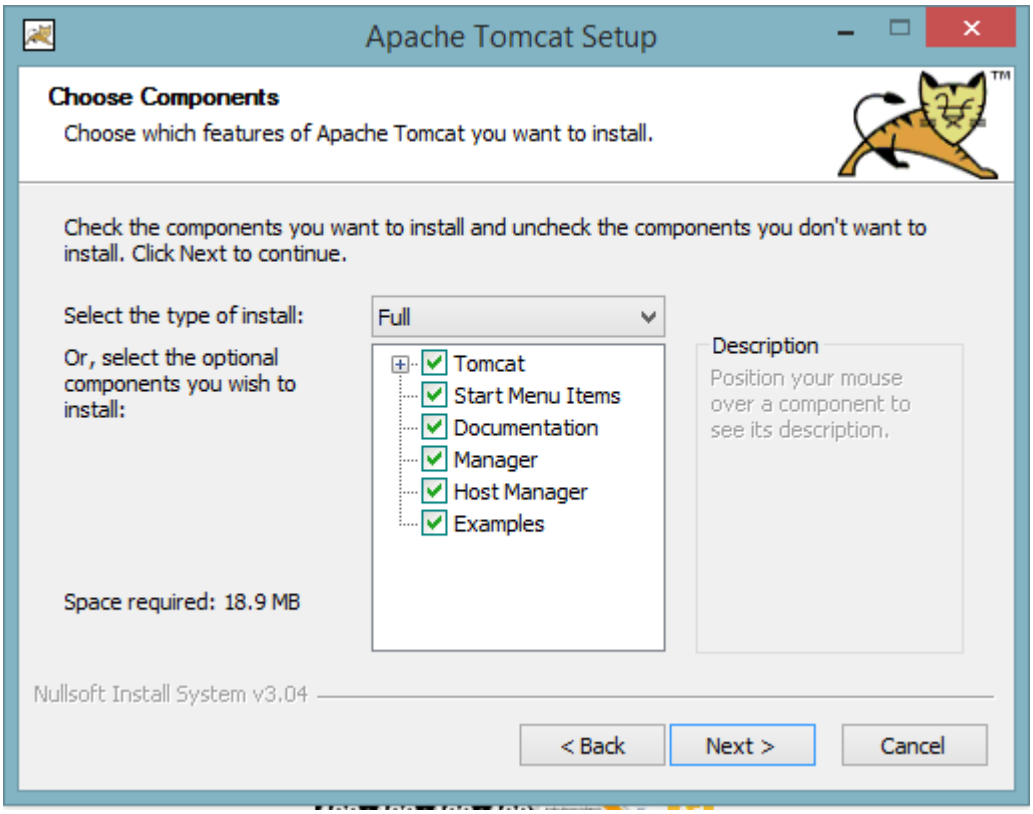
1. Install the apache tomcat
2. Write a jsp program with the required formats and save it in webapps
3. Run it in browser
4. Stop the program

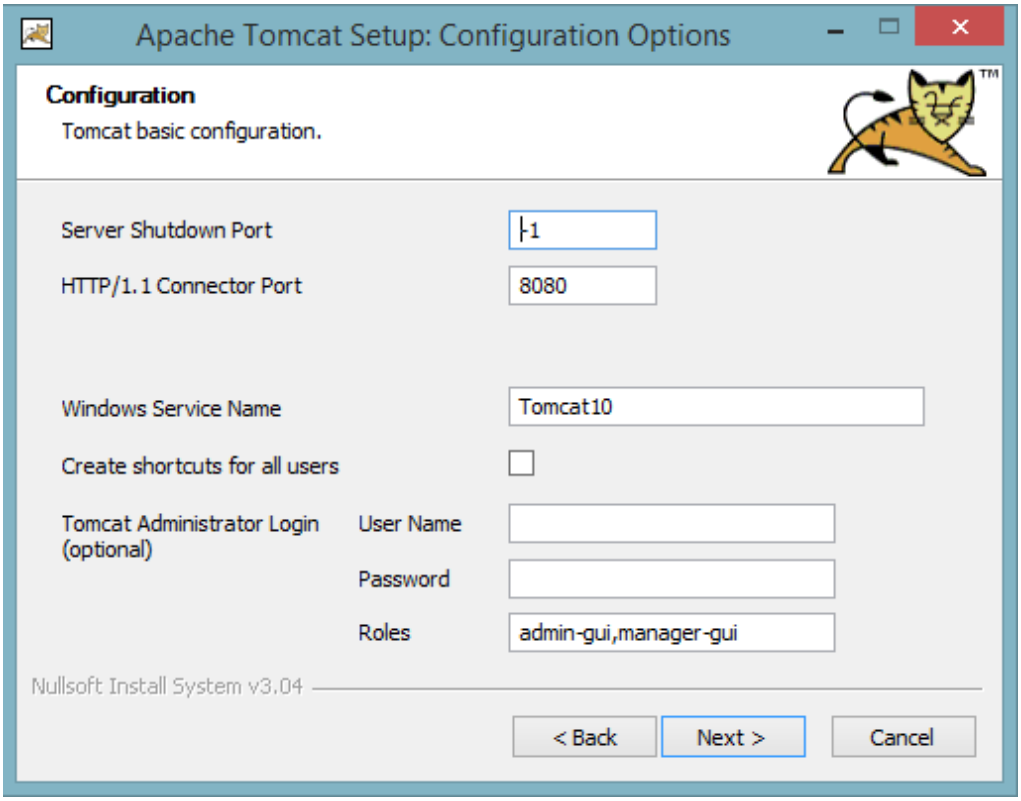
PROCEDURE

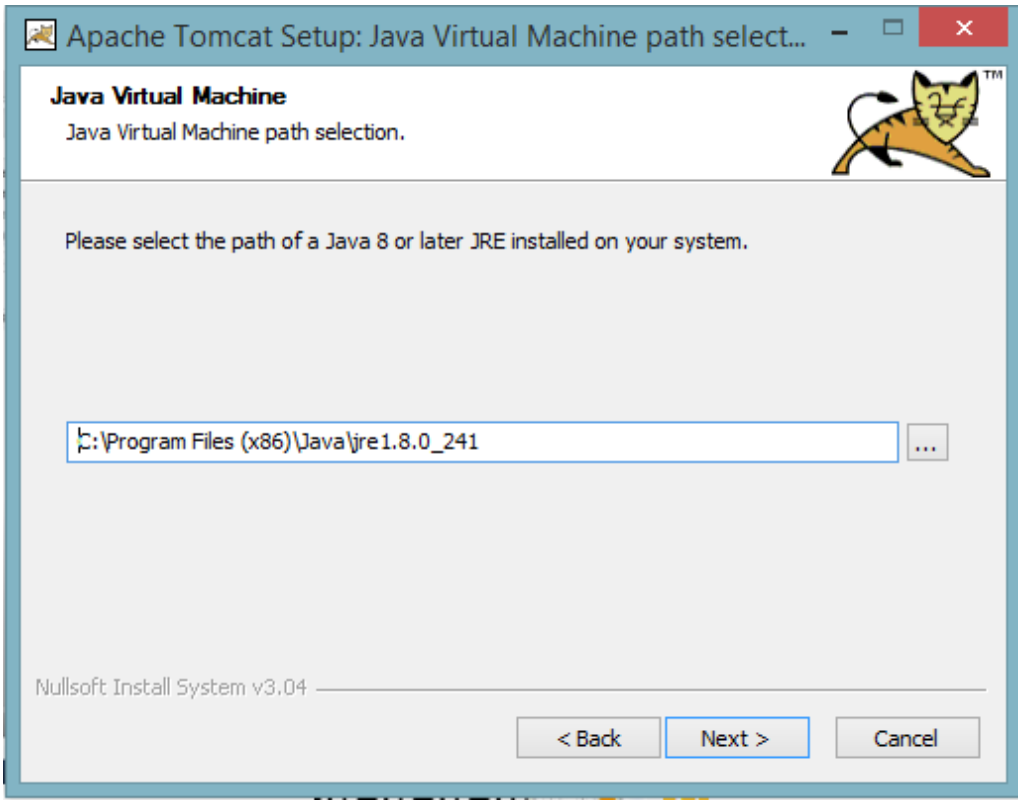


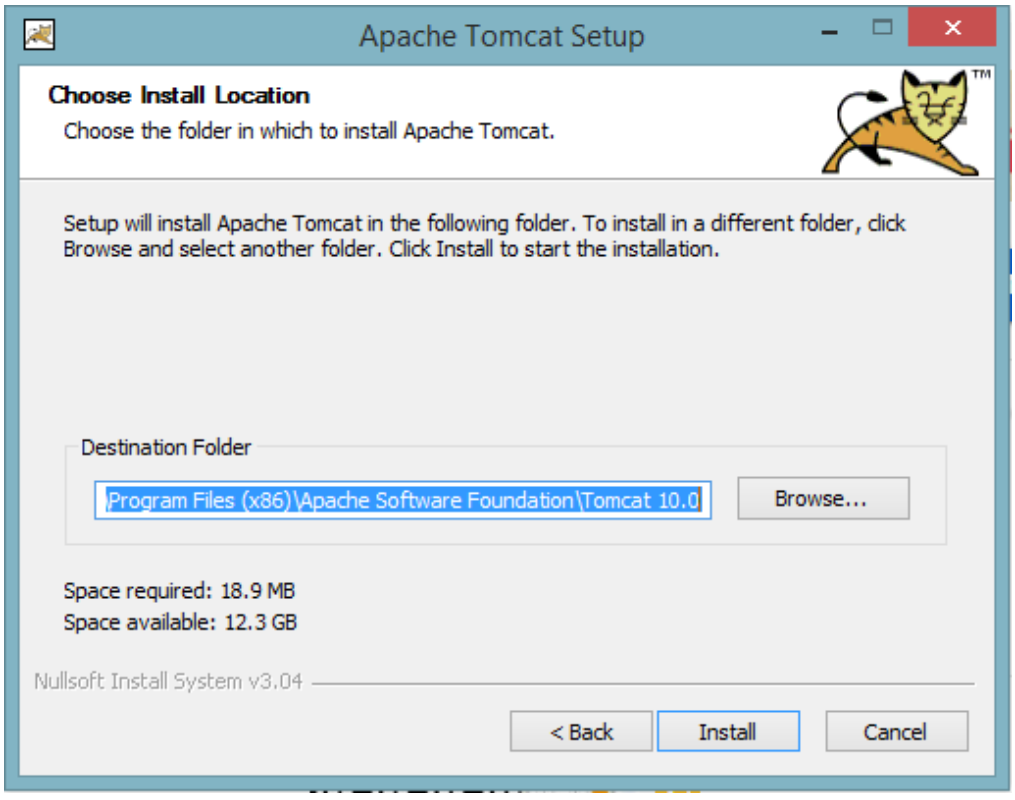

Apache Tomcat Setup
http://tomcat.apache.org
Apache Tomcat 9
Welcome to Apache Tomcat Setup
Setup will guide you through the installation of Apache Tomcat.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

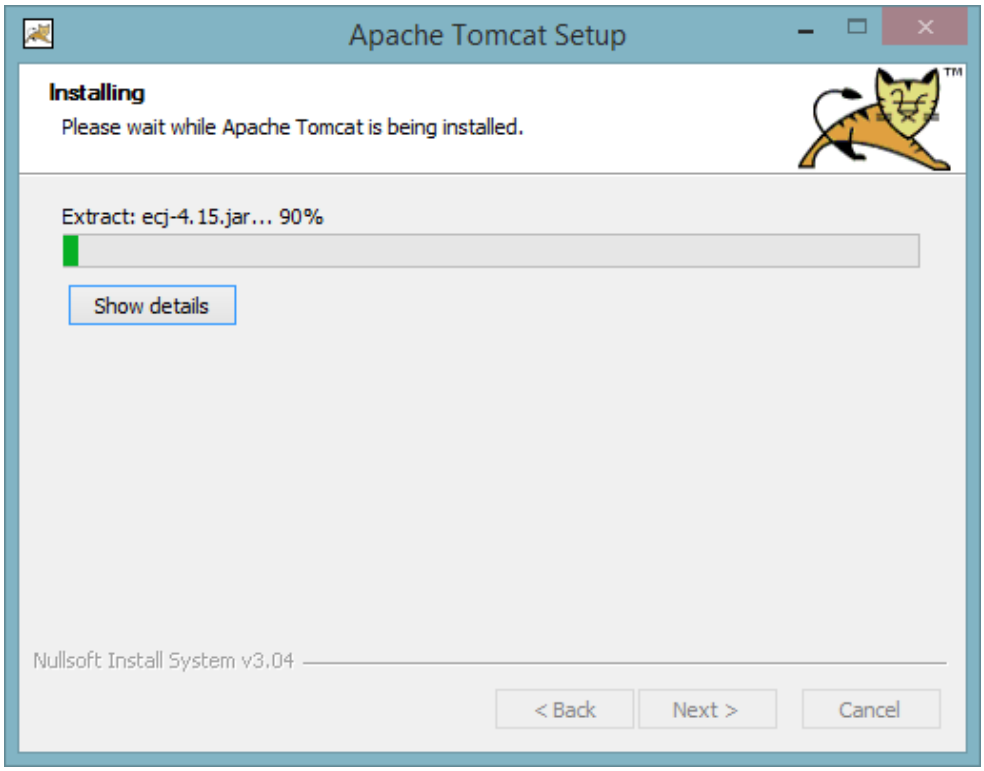

Apache Tomcat Setup
License Agreement
Please review the license terms before installing Apache Tomcat.
Press Page Down to see the rest of the agreement.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"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
< Back
I Agree
Cancel

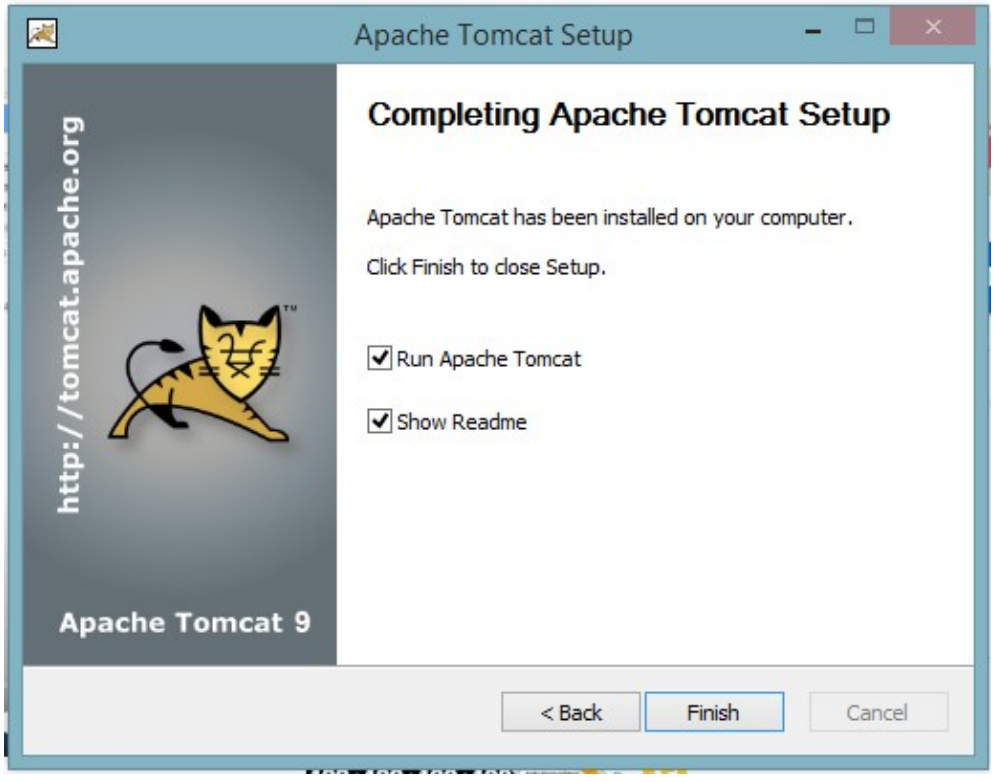

Apache Tomcat Setup
Choose Components
Choose which features of Apache Tomcat you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select the type of install:
Full
Or, select the optional components you wish to install:
Tomcat
Start Menu Items
Documentation
Manager
Host Manager
Examples
Description
Position your mouse over a component to see its description.
Space required: 18.9 MB
Nullsoft Install System v3.04
< Back
Next >
Cancel

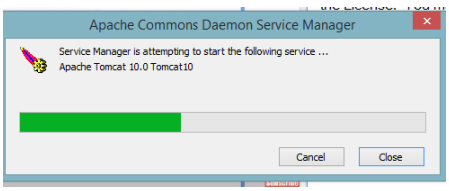

Apache Tomcat Setup: Configuration Options
Configuration
Tomcat basic configuration.
Server Shutdown Port
-1
HTTP/1.1 Connector Port
8080
Windows Service Name
Tomcat10
Create shortcuts for all users
Tomcat Administrator Login (optional)
User Name
Password
Roles
admin-gui,manager-gui
Nullsoft Install System v3.04
< Back
Next >
Cancel

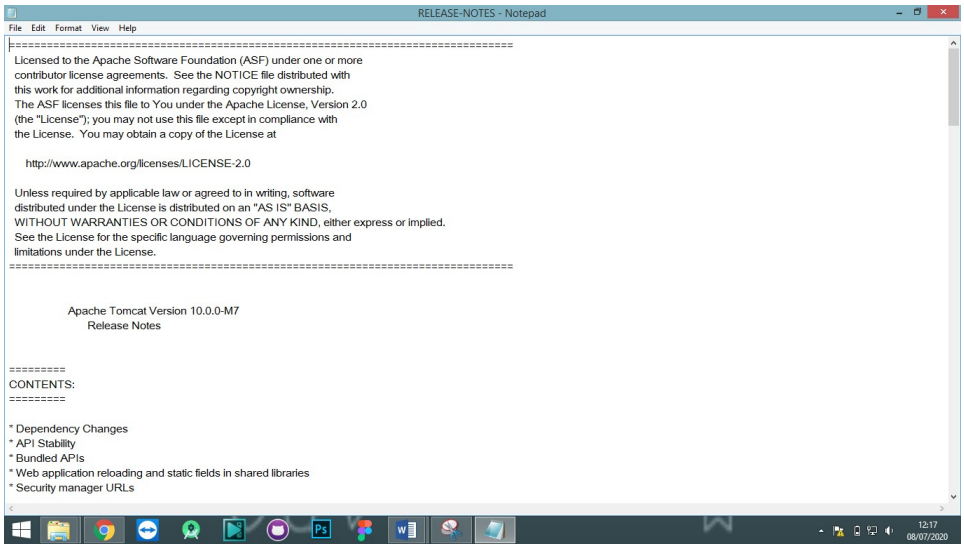

Apache Tomcat Setup: Java Virtual Machine path select...
Java Virtual Machine
Java Virtual Machine path selection.
Please select the path of a Java 8 or later JRE installed on your system.
C:\Program Files (x86)\Java\jre1.8.0_241
...
Nullsoft Install System v3.04
< Back
Next >
Cancel

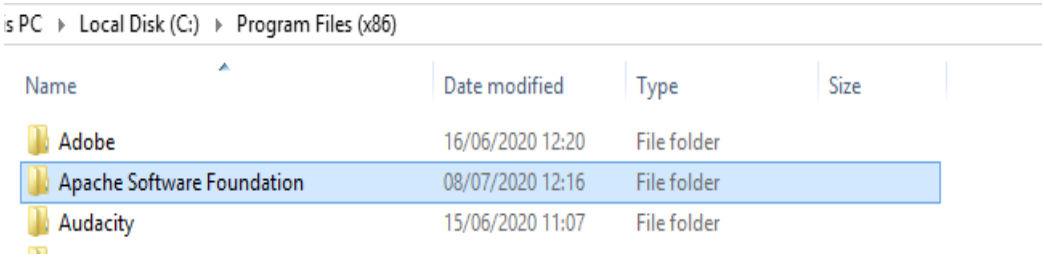

Apache Tomcat Setup
Choose Install Location
Choose the folder in which to install Apache Tomcat.
Setup will install Apache Tomcat in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
Program Files (x86)\Apache Software Foundation\Tomcat 10.0
Browse...
Space required: 18.9 MB
Space available: 12.3 GB
Nullsoft Install System v3.04
< Back
Install
Cancel

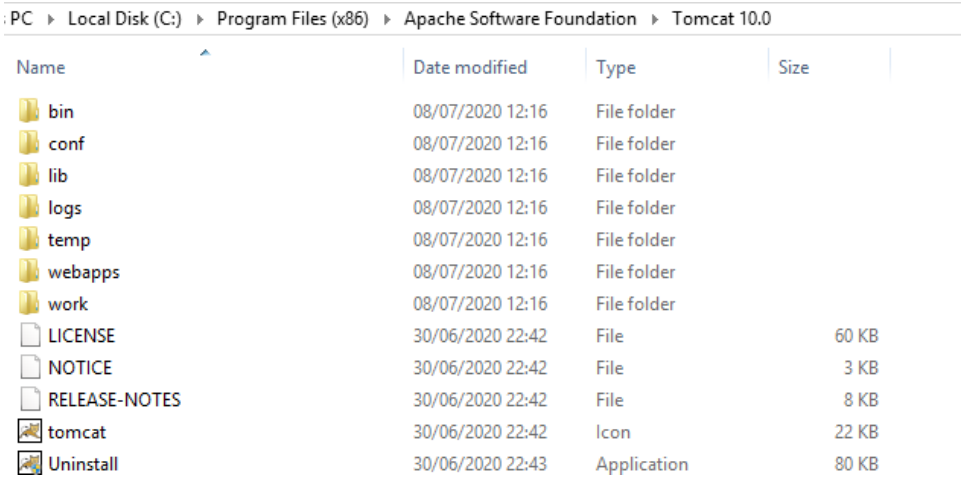

Apache Tomcat Setup
Installing
Please wait while Apache Tomcat is being installed.
Extract: ecj-4.15.jar... 90%
Show details
Nullsoft Install System v3.04
< Back
Next >
Cancel

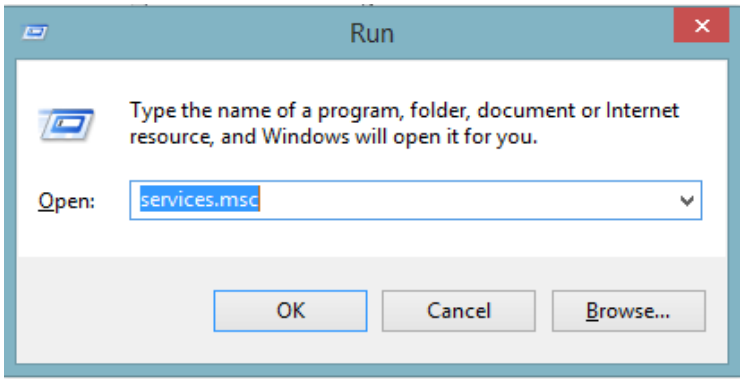

Apache Tomcat Setup
http://tomcat.apache.org
Apache Tomcat 9
Completing Apache Tomcat Setup
Apache Tomcat has been installed on your computer.
Click Finish to close Setup.
Run Apache Tomcat
Show Readme
< Back
Finish
Cancel

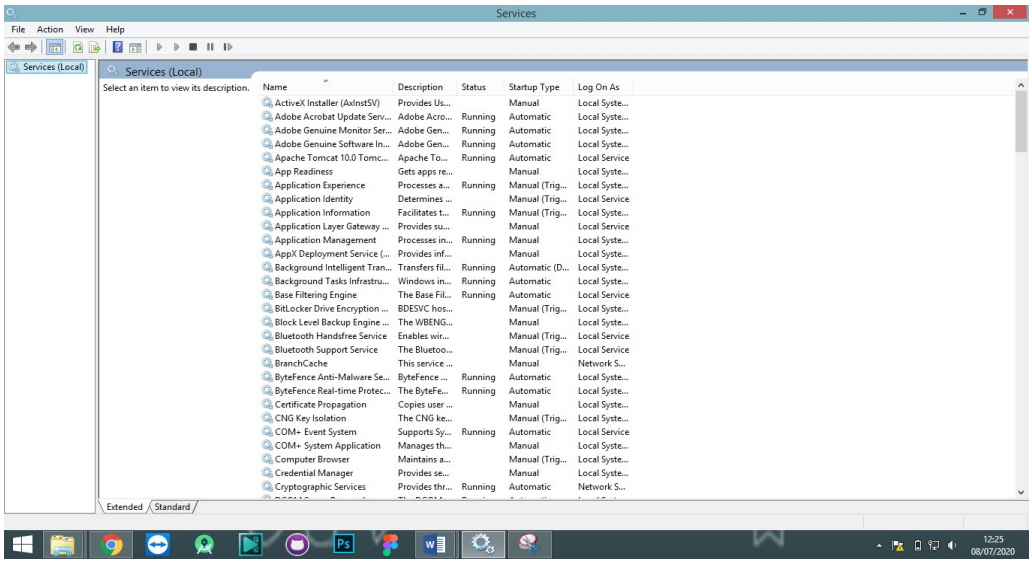

Apache Commons Daemon Service Manager
Service Manager is attempting to start the following service ...
Apache Tomcat 10.0 Tomcat10
Cancel
Close

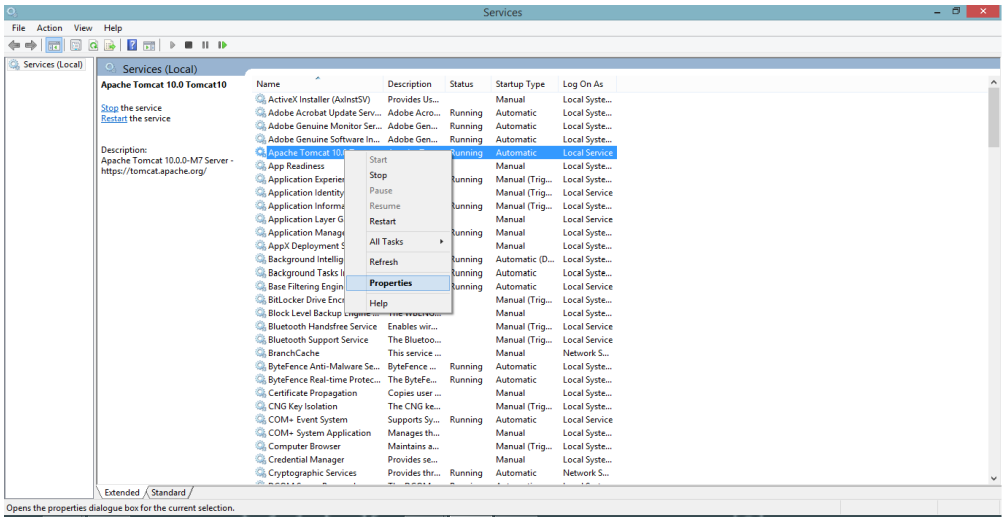

RELEASE-NOTES - Notepad
File Edit Format View Help
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Apache Tomcat Version 10.0.0-M7
Release Notes
CONTENTS:
* Dependency Changes
* API Stability
* Bundled APIs
* Web application reloading and static fields in shared libraries
* Security manager URLs

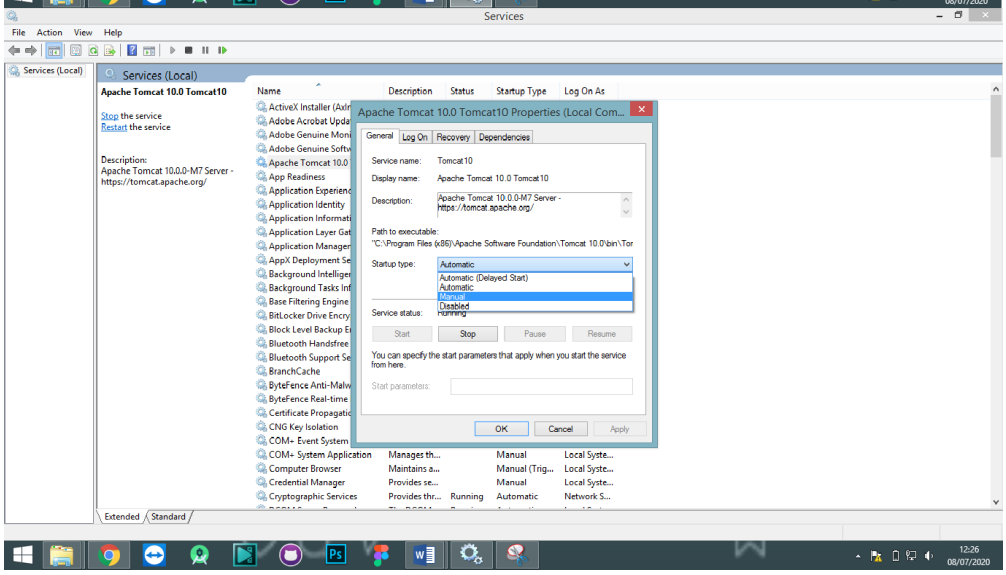

s PC ▸ Local Disk (C:) ▸ Program Files (x86)
Name
Date modified
Type
Size
Adobe
16/06/2020 12:20
File folder
Apache Software Foundation
08/07/2020 12:16
File folder
Audacity
15/06/2020 11:07
File folder

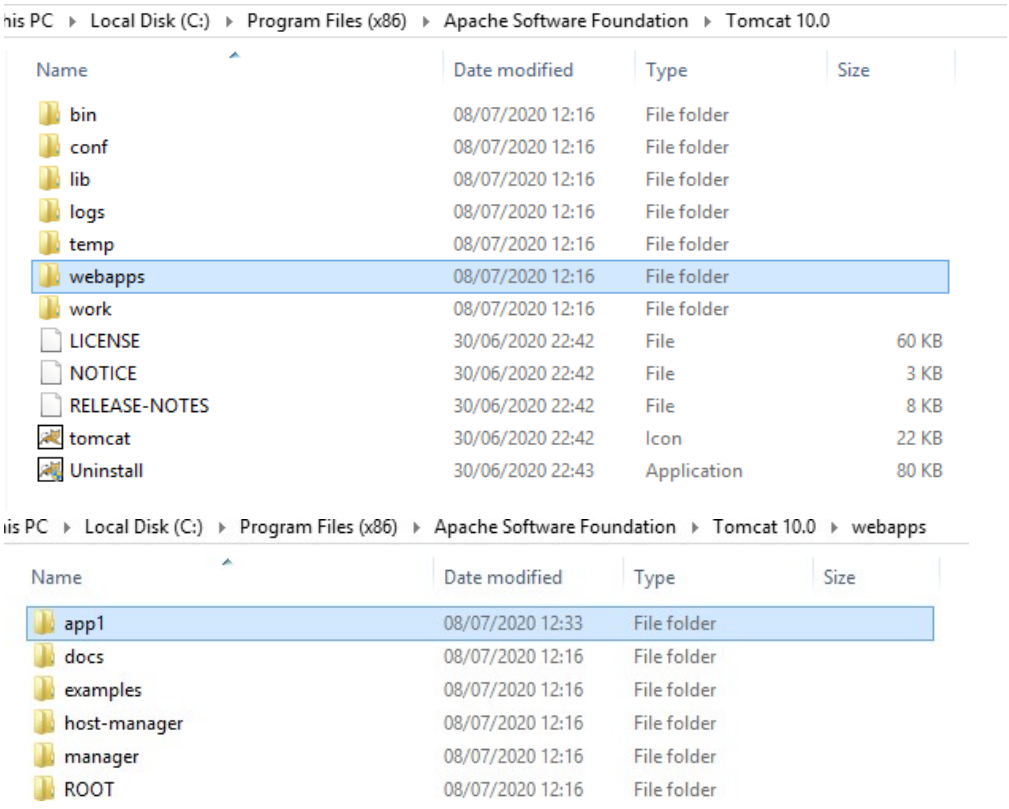

PC ▸ Local Disk (C:) ▸ Program Files (x86) ▸ Apache Software Foundation ▸ Tomcat 10.0
Name
Date modified
Type
Size
bin
08/07/2020 12:16
File folder
conf
08/07/2020 12:16
File folder
lib
08/07/2020 12:16
File folder
logs
08/07/2020 12:16
File folder
temp
08/07/2020 12:16
File folder
webapps
08/07/2020 12:16
File folder
work
08/07/2020 12:16
File folder
LICENSE
30/06/2020 22:42
File
60 KB
NOTICE
30/06/2020 22:42
File
3 KB
RELEASE-NOTES
30/06/2020 22:42
File
8 KB
tomcat
30/06/2020 22:42
Icon
22 KB
Uninstall
30/06/2020 22:43
Application
80 KB

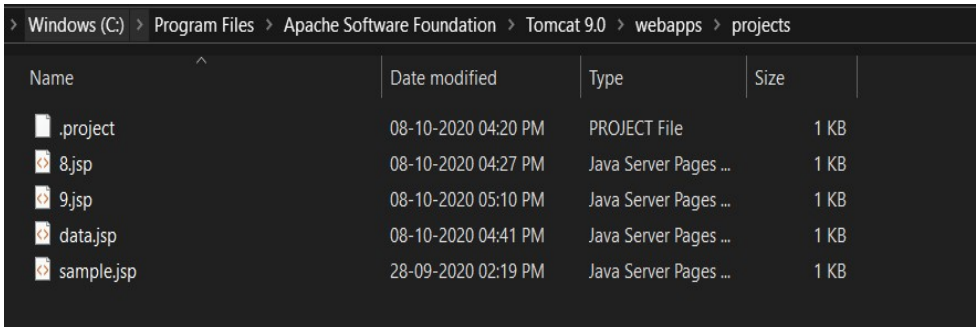

Run
Type the name of a program, folder, document or Internet resource, and Windows will open it for you.
Open:
services.msc
OK
Cancel
Browse...


Services
File Action View Help
Services (Local)
Select an item to view its description.
Name Description Status Startup Type Log On As
Extended Standard


Services
Apache Tomcat 10.0 Tomcat10
Stop the service
Restart the service
Description:
Apache Tomcat 10.0.0-M7 Server - https://tomcat.apache.org/
Start
Stop
Pause
Resume
Restart
All Tasks
Refresh
Properties
Help
Opens the properties dialogue box for the current selection.


Services
Apache Tomcat 10.0 Tomcat10 Properties (Local Com...
General Log On Recovery Dependencies
Service name: Tomcat10
Display name: Apache Tomcat 10.0 Tomcat10
Description: Apache Tomcat 10.0.0-M7 Server - https://tomcat.apache.org/
Path to executable:
"C:\Program Files (x86)\Apache Software Foundation\Tomcat 10.0\bin\Tor
Startup type: Automatic
Automatic (Delayed Start)
Automatic
Manual
Disabled
Service status: Running
Start Stop Pause Resume
You can specify the start parameters that apply when you start the service from here.
Start parameters:
OK Cancel Apply

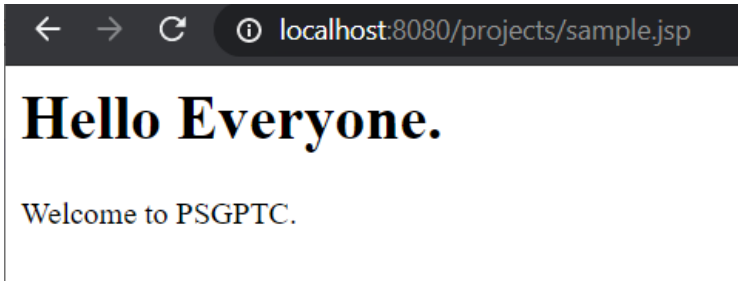




PROGRAM

```
<html>
<body>
<%
String str="Hello Everyone.";
%>
<h1><%=str %></h1>
<% out.println("Welcome to PSGPTC."); %>
</body>
</html>
```

OUTPUT



RESULT

Thus tomcat and apache web server was installed and configured successfully.

EX NO 8: INSTALL DATABASE MYSQL.WRITE A JSP PROGRAM TO THE DATABASE
DATE: AND EXTRACT DATA FROM THE TABLE AND DISPLAY IT

AIM

To install database MySQL.

Write a JSP program to connect to the database and extract data from the table and display it.

THEORY

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. SP is an integral part of Java EE, a complete platform for enterprise class applications. JSP can play a part in the simplest applications to the most complex and demanding.

A scriptlet can contain any number of JAVA language statements, variable or method declarations, or expressions that are valid in the page scripting language.

SYNTAX

<% code fragment %>

The XML equivalent of the above syntax as follows

```
<jsp:scriptlet>
  code fragment
</jsp:scriptlet>
```

PROCEDURE

1. Start a program
2. Create a database
3. Create a table and insert the values
4. Connect the database with the jsp file using mysql connector
5. Run the jsp file in browser
6. Stop the program

PROGRAM

```
<%@page import="java.sql.*"%>
<% Class.forName("com.mysql.jdbc.Driver");
Connection connection = null;
Statement statement = null;
ResultSet resultSet = null;
%>
<h2 align="center"><font><strong>Retrieve data from database in jsp</strong></font></h2>
<table align="center" cellpadding="5" cellspacing="5" border="1">
<tr bgcolor="#A52A2A">
<td><b>Name</b></td>
<td><b>Email</b></td></tr>
```

```
<% connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/webtechnology", "root", "");
statement=connection.createStatement();
String sql ="SELECT * FROM users";
resultSet = statement.executeQuery(sql);
while(resultSet.next()){
%>
<tr bgcolor="#DEB887">
<td><%=resultSet.getString("name") %></td>
<td><%=resultSet.getString("email") %></td>
</tr><% } %>
</table>
```

OUTPUT

DATABASE NAME

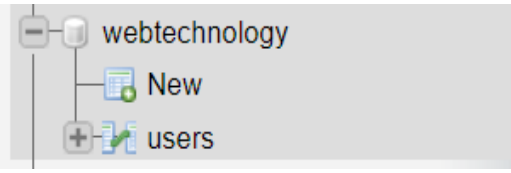
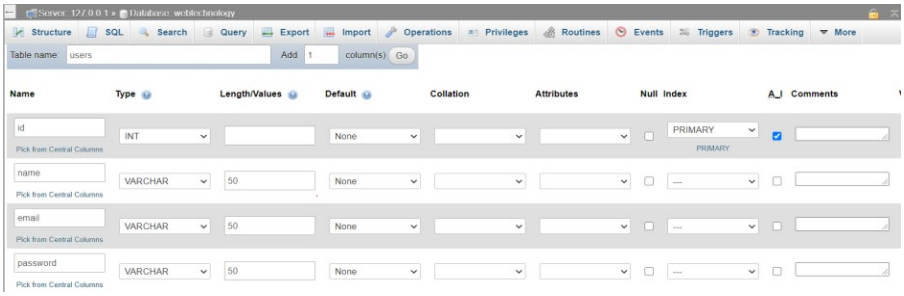


TABLE NAME



INSERTING THE VALUES

Retrieve data from database in jsp

Name	Email
Bill Gates	bill@microsoft.com
Jeff Bezos	jeff@amazon.com

RESULT

Thus to install database MySQL and to write a JSP program to connect to the database and extract data from the table and display it was done successfully.

EX NO : 9 **WRITE A JSP SERVLET FOR AUTHENTICATING USER BY HIS**

DATE: **PASSWORD.**

AIM

To Write a JSP servlet for authenticating user by his password.

THEORY

Java Server Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.A Java Server Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application.

Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands. JSP can collect input from users through Webpage forms, present records from a database or another source, and create Webpages dynamically. JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.

PROCEDURE

1. Start the program
2. Create the user name and password
3. Check if the username and password present in the database
4. If present, return login successfully else login failed.
5. Stop the program

PROGRAM

9.jsp

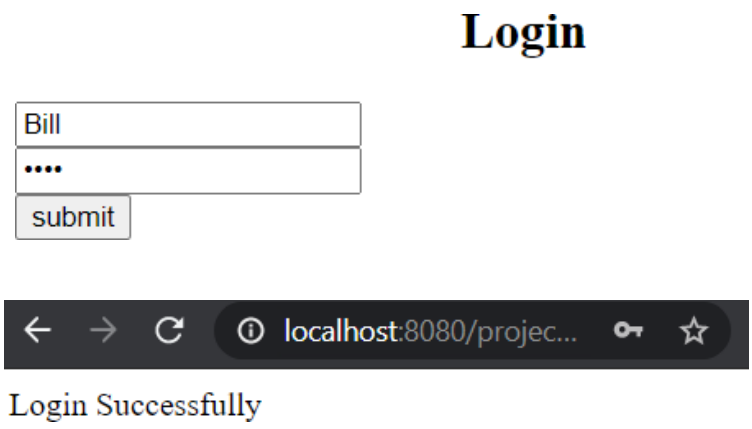
```
<h2>Login</ h2>
<form action='data.jsp' method='get' >
<input type='text' placeholder='Name...' name='name' /><br>
<input type='password' placeholder='Password...' name='password' /><br>
<input type='submit' value='submit' /><br>
</form>
```

Data.jsp

```
<%@page import="java.sql.*"%>
<%
Class.forName("com.mysql.jdbc.Driver");
Connection connection = null;
```

```
Statement statement = null;
ResultSet resultSet = null;
String name=request.getParameter("name");
String pass=request.getParameter("password");
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/webtechnology","roo
t", “”);
statement=connection.createStatement();
String sql ="SELECT * FROM users WHERE name='"+name+"' AND password='"+pass+"'";
resultSet = statement.executeQuery(sql);
int rowCount=0;
while(resultSet.next()){
        rowCount++; }
if(rowCount>0){ %>
Login Successfully
<% }
else{ %>
Login Failed
<% }  %>
```

OUTPUT



RESULT

Thus to Write a JSP servlet for authenticating user by his password was done successfully.

EX NO 10: DEVELOP A SINGLE PAGE WEBSITE USING THE ANGULAR JS.

DATE:

AIM

To develop a single page website using the angular JS.

THEORY

AngularJS is a very powerful JavaScript Framework. It is used in Single Page Application (SPA) projects. It extends HTML DOM with additional attributes and makes it more responsive to user actions. AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache license version 2.1. AngularJS is a efficient framework that can create Rich Internet Applications (RIA). AngularJS provides developers an option to write client-side applications using JavaScript in a clean Model View Controller (MVC) way. Applications written in AngularJS are cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.

SYNTAX

```
angular.module("myapp", [])
.controller("appController", function($scope) {
// definition of controller
} );
```

PROCEDURE

1. Start a program
2. Include the source link for the AngularJS script.
3. Create the ng-app element.
4. Include HTML code and CSS styling in your page.
5. Save the file as .html and run the file in your browser.

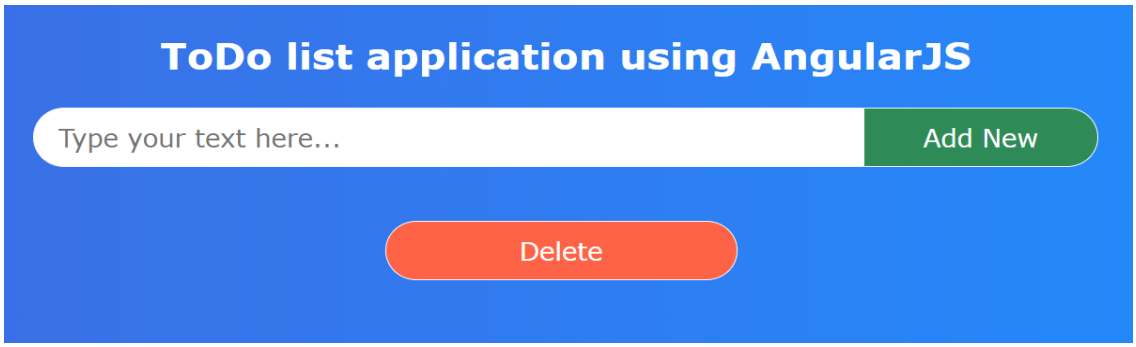
PROGRAM

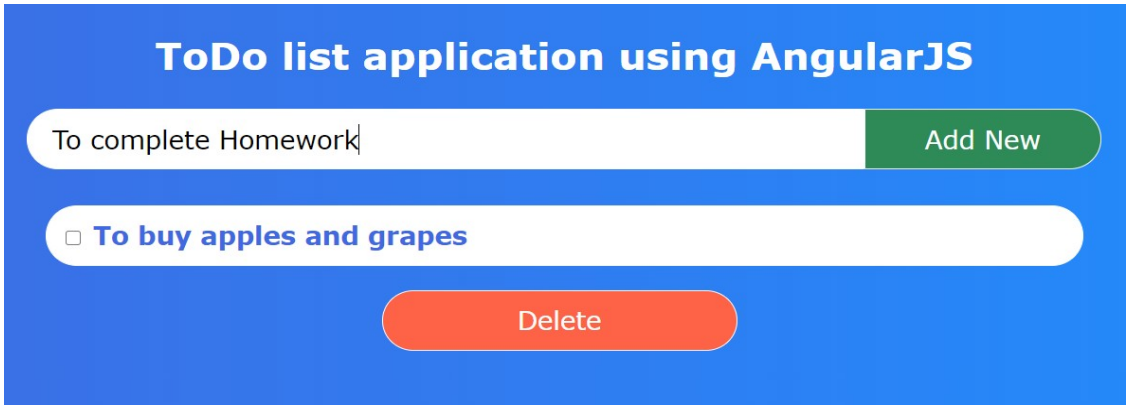
```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body ng-app="myApp" ng-controller="todoCtrl">

<h2>ToDo application using AngularJS</h2>

<form ng-submit="todoAdd()">
   <input id='input' type="text" ng-model="todoInput" size="50" placeholder="Type your text here..." required>
   <input id='btn' type="submit" value="Add New">
</form>
<br>
```

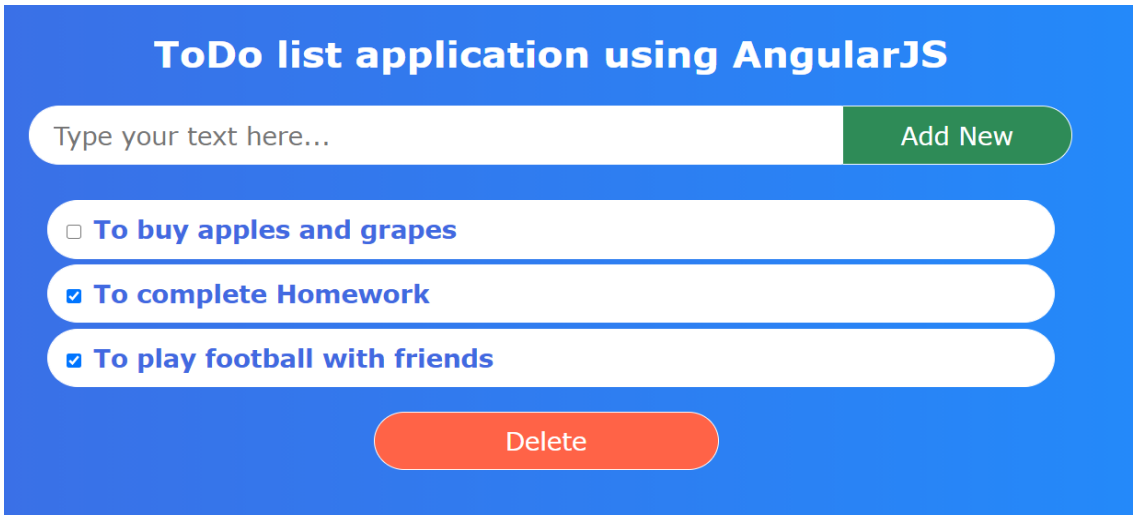
```
<div id='todos' ng-repeat="x in todoList">
<input type="checkbox" ng-model="x.done">
<span ng-bind="x.todoText"></span>
</div>
</center>
<p><button id='btn-remove' ng-click="remove()">Delete</button></p>
<script>
var app = angular.module('myApp', []);
app.controller('todoCtrl', function($scope) {
$scope.todoList = [];
 $scope.todoAdd = function() {
 $scope.todoList.push({todoText:$scope.todoInput, done:false});
  $scope.todoInput = "";
    };

 $scope.remove = function() {
  var oldList = $scope.todoList;
  $scope.todoList = [];
  angular.forEach(oldList, function(x) {
  if (!x.done) $scope.todoList.push(x);
   });
   };
   });
</script>
</body>
</html>
<style>
body{
background-image: linear-gradient(to right,royalblue 0%,dodgerblue 100%);
font-family:Verdana;
color:white;
font-size:22px;
text-align:center;
}
input,button{
font-family:Verdana;
}
#input{
font-size:22px;
padding:12px;
padding-left:20px;
border:1px solid white;
outline:none;
border-radius:30px 0 0 30px;
```

```
margin-right:0;
}
#btn{
color:white;
background:seagreen;
border-radius:0 30px 30px 0;
border:1px solid white;
outline:none;
padding:12px;
font-size:22px;
width:200px;
margin-left:-8px;
}
#btn:active{
background:mediumseagreen;
}
#btn-remove{
color:white;
background:tomato;
border-radius:30px;
border:1px solid white;
outline:none;
padding:12px;
font-size:22px;
width:300px;
margin-left:-8px;
}
#btn-remove:active{
background:red;
}
#todos{
margin-top:5px;
padding:12px;
border-radius:30px;
border:1px solid white;
width:850px;
background:white;
color:royalblue;
text-align:left;
font-weight:bold;
}
</style>
```

OUTPUT







RESULT

Thus single page website using the angular JS was developed successfully.