EX.NO:1
DATE:

IMAGE MAPPING

AIM:

To create a web page which includes a map and display the related information when a hot spot is clicked on the map.

ALGORITHM:

1. Create a html file with map tag.
2. Set the source attribute of the image tag to the location of the image and also set the use map attribute.
3. Specify an area with name, shape and href set to the appropriate values.
4. Repeat step 3 as many hot spots you want to put in the map.
5. Create html files for each and every hot spot the user want to select.

CODING:

Image.html

```
<html>
<title> Image Mapping</title>
<body>
<body style="backgorund-color:black">
<img src="indiamap.gif" usemap="#India">
<map name="India">
<area href="tamilnadu.html" shape="circle" coords="280,741,40">
<area href="andhra.html" shape="circle" coords="295,633,40">
<area href="karnataka.html" shape="circle" coords="224,644,40">
<area href="kerala.html" shape="circle" coords="232,766,40">
</map>
</body>
</html>
```

Tamilnadu.html

```
<html>
<head>
<title>About Tamilnadu</title>
</head>
<body>
<h1>Tamilnadu</h1>
<UL>
<LI>Area : 1,30,058 Sq. Kms.</LI>
<LI>Capital : Chennai</LI>
<LI>Language : Tamil</LI>
<LI>Population :72,147,030</LI>
</UL>
<a href="image.html">Indiamap</a>
</body>
</html>
```

Andhra.html

```
<html>
<head>
<title>About Andhra Pradesh</title>
</head>
<body>
<h1>Andhra Pradesh</h1>
<UL>
<LI>Area : 160,205 Sq. Kms.</LI>
<LI>Capital : Hyderabad</LI>
<LI>Language : Telugu</LI>
<LI>Population :49,386,799</LI>
</UL>
<a href="image.html">Indiamap</a>
</body>
</html>
```

Kerala.html

```
<html>
<head>
<title>About Kerala</title>
</head>
<body>
<h1>Kerala</h1>
<UL>
<LI>Area : 38,863 Sq. Kms.</LI>
<LI>Capital : Thiruvananthapuram</LI>
<LI>Language : Malayalam</LI>
<LI>Population :33,387,677</LI>
</UL>
<a href="image.html">Indiamap</a>
</body>
</html>
```

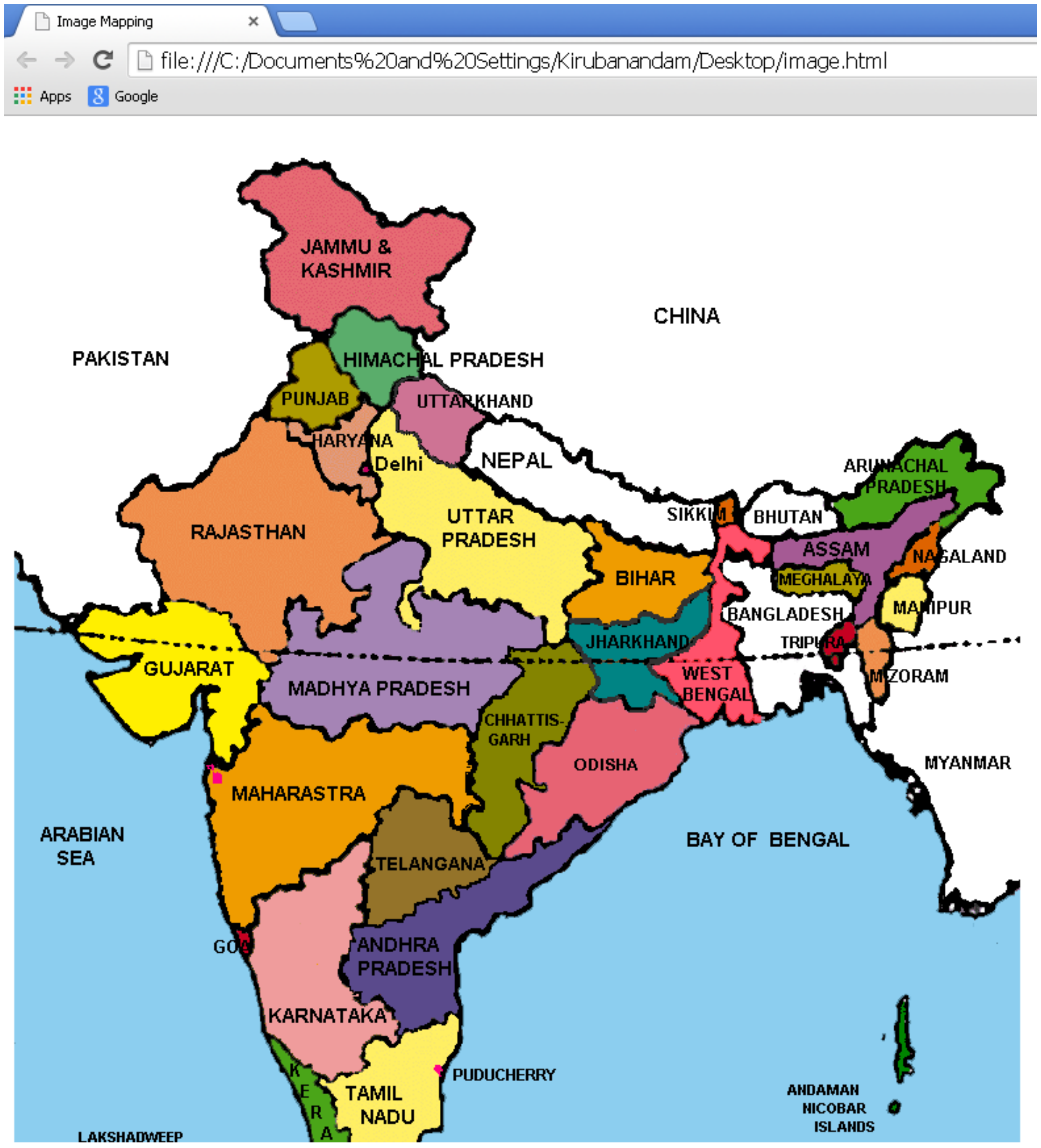
Karnataka.html

```
<html>
<head>
<title>About Karnataka</title>
</head>
<body>
<h1>Karnataka</h1>
<UL>
<LI>Area : 1,91,791 Sq. Kms</LI>
<LI>Capital : Bangalore</LI>
<LI>Language : Kannada</LI>
<LI>Population : 6,1,130,704</LI>
</UL>
```

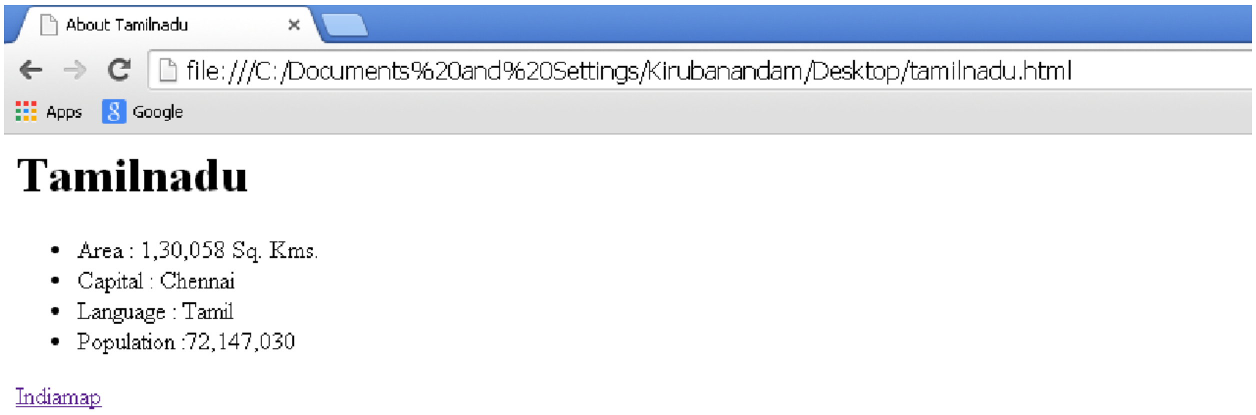
```
<a href='image.html'>India Map</a>
</body>
</html>
```

OUTPUT:

Image.html



Tamilnadu.html

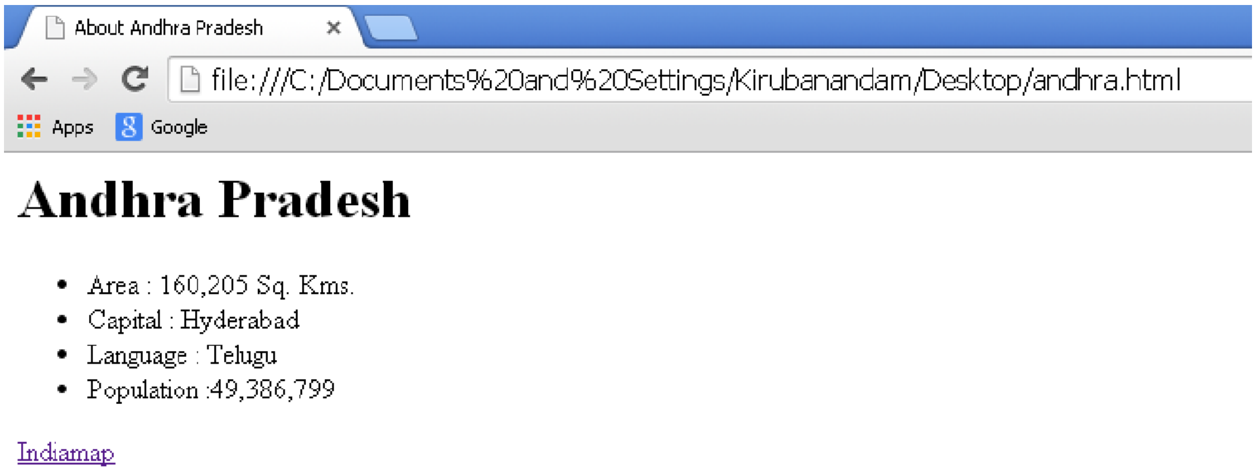

About Tamilnadu
file:///C:/Documents%20and%20Settings/Kirubanandam/Desktop/tamilnadu.html
Apps
Google

Tamilnadu

- Area : 1,30,058 Sq. Kms.
- Capital : Chennai
- Language : Tamil
- Population :72,147,030

Indiamap

Andhra.html

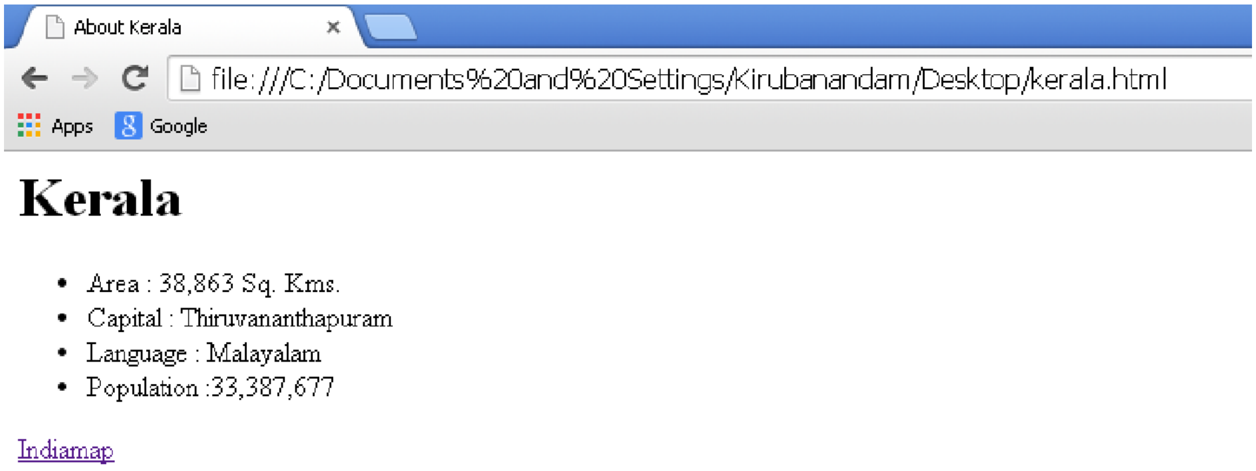

About Andhra Pradesh
file:///C:/Documents%20and%20Settings/Kirubanandam/Desktop/andhra.html
Apps
Google

Andhra Pradesh

- Area : 160,205 Sq. Kms.
- Capital : Hyderabad
- Language : Telugu
- Population :49,386,799

Indiamap

Kerala.html

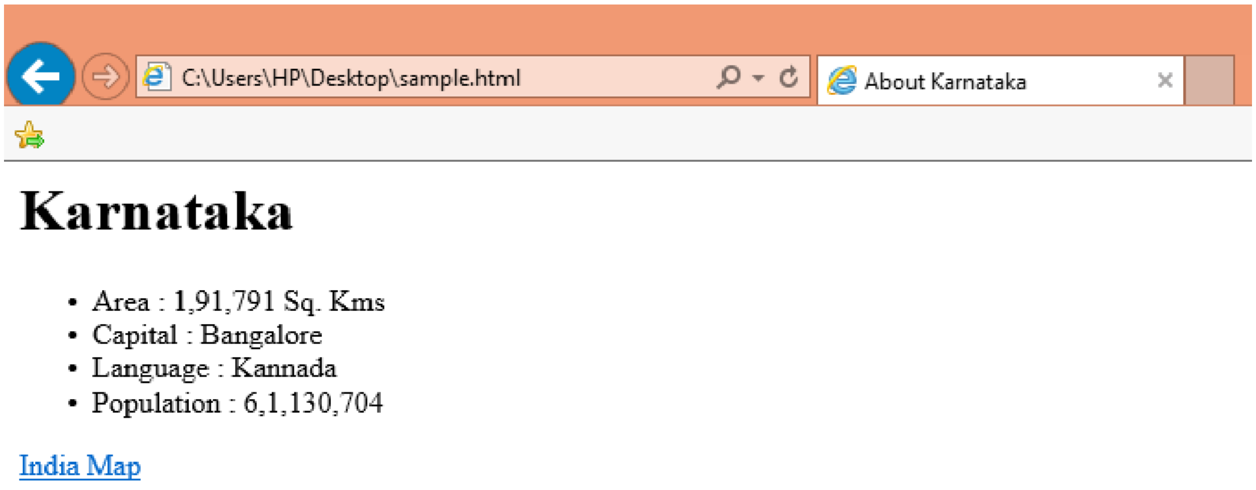

About Kerala
file:///C:/Documents%20and%20Settings/Kirubanandam/Desktop/kerala.html
Apps
Google

Kerala

- Area : 38,863 Sq. Kms.
- Capital : Thiruvananthapuram
- Language : Malayalam
- Population :33,387,677

Indiamap

Karnataka.html



Karnataka

- Area : 1,91,791 Sq. Kms
- Capital : Bangalore
- Language : Kannada
- Population : 6,1,130,704

India Map

RESULT: Thus the creation of a web page which includes a map and display the related information when a hot spot is clicked on the map was executed successfully.

EX.NO:2a
DATE:

CASCADING STYLESHEET
INLINE STYLESHEET

AIM:
To apply inline style sheets to a web page using HTML tags.

ALGORITHM:
1. Start the program with inclusion of <html> tag.
2. The <style> tag is used for combining various styles.
3. Specify the font color, weight and size inside style as its attributes.
4. Close the corresponding tags with a'/' symbol.
5. Stop the program.

CODING:

```
<html>
<head>
<title> Inline stylesheet </title>
</head>
<body>
<h1 style= "font-family: jokerman; color: blue; margin-left:
50px;">Inline style sheet</h1>
<p style= "font-family: Comic Sans MS; color: green; margin-left:
30px;">An inline style loses many of the advantages of a style sheet
by mixing content with presentation.</p>
<p style= "font-family: Calibri; color: red; font-size: 20; font-
weight: bold;"> URI identifies the resource upon which to apply the
request. <br>It is intended to be associated with particular resource.
<br>Cell spacing attribute determines the amount of space between two
adjacent cells or between side of cells.</p>
</body>
</html>
```

OUTPUT:

```
Inline style sheet

An inline style loses many of the advantages of a style sheet by mixing content
with presentation.

URI identifies the resource upon which to apply the request.
It is intended to be associated with particular resource.
Cell spacing attribute determines the amount of space between two adjacent cells or
between side of cells.
```

RESULT:
Thus the program to perform inline styles to a web page has been executed successfully and the output is verified.

EX.NO:2b
DATE:

INTERNAL STYLESHEET

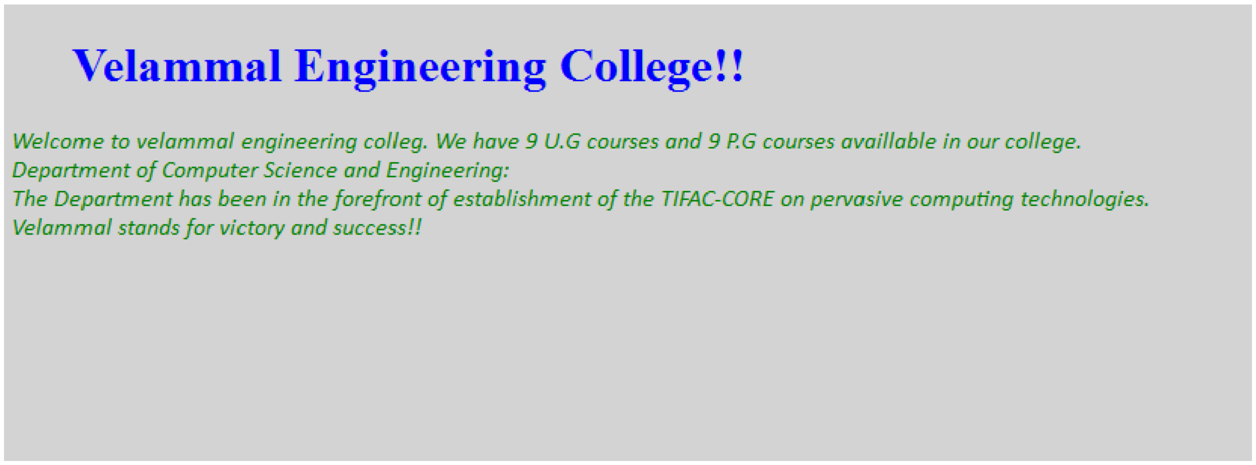
AIM:
To implement internal stylesheet in css HTML.

ALGORITHM:
1. Start.
2. Open the program with <DOCTYPE! html> tag.
3. Insert the style sheet within the <head > tag.
4. Use the internal stylesheet in the <body> tag.
5. Stop the program.

CODING:

```
<!DOCTYPE html>
<html>
<head>
<style>
body  {  background-color: lightgrey }
h1 {  color: blue;  margin-left: 40px; }
 P {  color: green;  font-family: "Calibri";  font-style: italic; }
</style>
</head>
<body>
<h1>Velammal Engineering College!!</h1>
<p>Welcome to velammal engineering colleg. We have 9 U.G courses and 9
P.G courses availlable in our college.<br>Department of Computer
Science and Engineering:<br> The Department has been in the forefront
of establishment of the TIFAC-CORE on pervasive computing
technologies.<br> Velammal stands for victory and success!!</p>
</body>
</html>
```

OUTPUT:



Velammal Engineering College!!

Welcome to velammal engineering colleg. We have 9 U.G courses and 9 P.G courses availlable in our college.
Department of Computer Science and Engineering:
The Department has been in the forefront of establishment of the TIFAC-CORE on pervasive computing technologies.
Velammal stands for victory and success!!

RESULT: Thus the above program to implement external stylesheet has been executed successfully.

EX.NO:2c
DATE:

EXTERNAL STYLESHEET

AIM:
To create a webpage implementing external stylesheet.

ALGORITHM:
1. Start the program with inclusion of <html> tag.
2. The <link> tag is used for linking css files.
3. Specify the font color, weight and size inside the .css files.
4. Close the corresponding tags with a'/' symbol.
5. Stop the program.

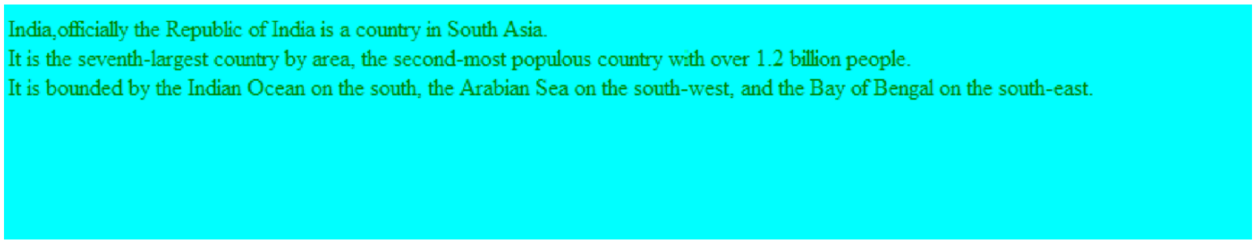
PROGRAM:

Sample.html

```
<html>
<head>
<title>CSS EXTERNAL STYLESHEET</title>
<link rel="stylesheet" type="text/css" href="style1.css"
title="style1"/>
<link rel="alternate stylesheet" type="text/css" href="style2.css"
title="style2"/>
</head>
<body>
<p>India,officially the Republic of India is a country in South Asia.
It is the seventh-largest country by area, the second-most populous
country with over 1.2 billion people.  It is bounded by the Indian
Ocean on the south, the Arabian Sea on the south-west, and the Bay of
Bengal on the south-east. </p>
</body>
</html>
```

Style1.css
```
 body{background-color:cyan}
 p{font-size:16;font-style:bold;color:green}
```

Style2.css
```
 p{font-family:arial;font-size:10;color:red}
```

OUTPUT:

```
India,officially the Republic of India is a country in South Asia.
It is the seventh-largest country by area, the second-most populous country with over 1.2 billion people.
It is bounded by the Indian Ocean on the south, the Arabian Sea on the south-west, and the Bay of Bengal on the south-east.
```

RESULT: Thus the above program to implement external stylesheet has been executed successfully.

EX.NO:3a
DATE:

FORM VALIDATION USING JAVASCRIPT

AIM:
To write a program using javascript for validating a form.

ALGORITHM:
1. Start.
2. Create a form in html which includes text field for name ,password.
3. Form also includes submit button.
4. When the button is clicked , onclick event call the function that validates the form.
5. End.

CODING:
```
<!DOCTYPE html>
<html lang="en"><head>
<meta charset="utf-8">
<title>JavaScript Form Validation using a sample registration
form</title>
<meta name="keywords" content="example, JavaScript Form Validation,
Sample registration form" />
<meta name="description" content="This document is an example of
JavaScript Form Validation using a sample registration form. " />

<script >
function formValidation()
{
var uid = document.registration.userid;
var passid = document.registration.passid;
var uname = document.registration.username;
var uadd = document.registration.address;
var ucountry = document.registration.country;
var uzip = document.registration.zip;
var uemail = document.registration.email;
var umsex = document.registration.msex;
var ufsex = document.registration.fsex;
if(userid_validation(uid,5,12))
{
      if(passid_validation(passid,7,12))
      {
            if(allLetter(uname))
            {
                  if(alphanumeric(uadd))
                  {
                        if(countryselect(ucountry))
                        {
```

```
                                        if(allnumeric(uzip))
                                        {
                                                if(ValidateEmail(uemail))
                                                {
                                                        if(validsex(umsex,ufsex))
                                                        {
                                                        }
                                                }
                                        }
                                }
                        }
                }
        }
}
return false;
}

function userid_validation(uid,mx,my)
{
var uid_len = uid.value.length;
if (uid_len == 0 || uid_len >= my || uid_len < mx)
{
alert("User Id should not be empty / length be between "+mx+" to
"+my);
uid.focus();
return false;
}
return true;
}


function passid_validation(passid,mx,my)
{
var passid_len = passid.value.length;
if (passid_len == 0 ||passid_len >= my || passid_len < mx)
{
alert("Password should not be empty / length be between "+mx+" to
"+my);
passid.focus();
return false;
}
return true;
}
```

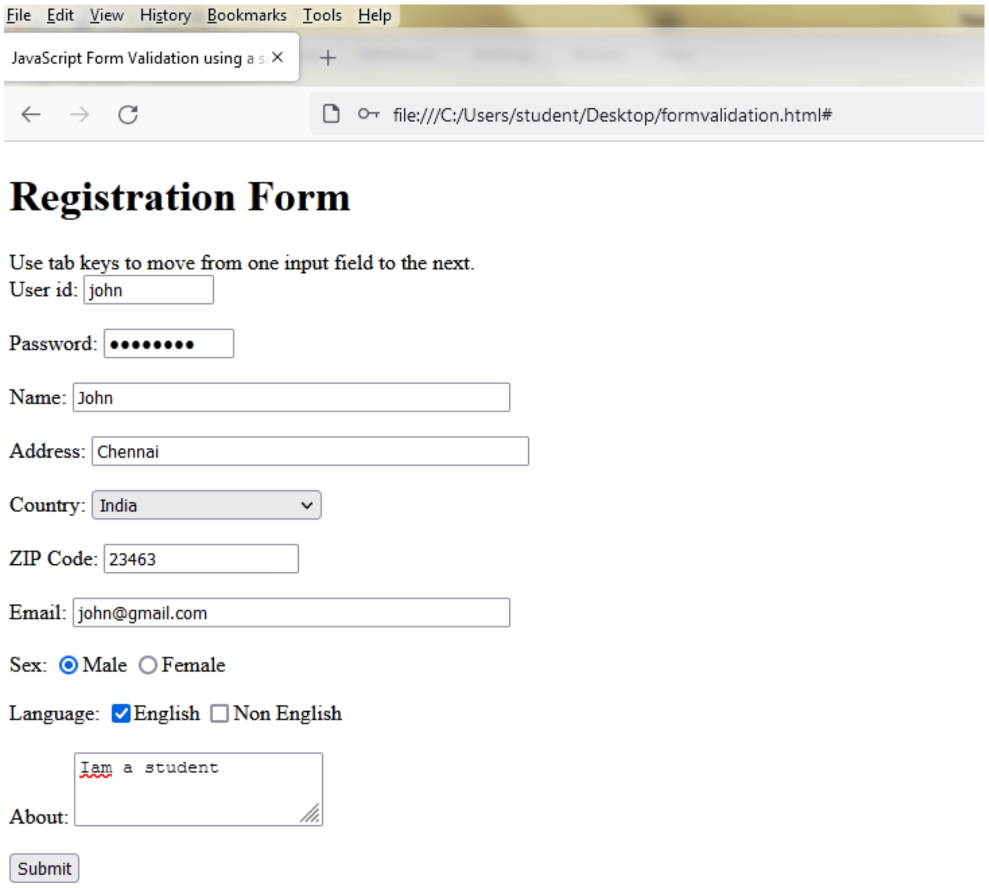
```
function allLetter(uname)
{
var letters = /^[A-Za-z]+$/;
if(uname.value.match(letters))
{
return true;
}
else
{
alert('Username must have alphabet characters only');
uname.focus();
return false;
}
}


function alphanumeric(uadd)
{
var letters = /^[0-9a-zA-Z]+$/;
if(uadd.value.match(letters))
{
return true;
}
else
{
alert('User address must have alphanumeric characters only');
uadd.focus();
return false;
}
}


function countryselect(ucountry)
{
if(ucountry.value == "Default")
{
alert('Select your country from the list');
ucountry.focus();
return false;
}
else
{
return true;
}
}
```

```
function allnumeric(uzip)
{
var numbers = /^[0-9]+$/;
if(uzip.value.match(numbers))
{
return true;
}
else
{
alert('ZIP code must have numeric characters only');
uzip.focus();
return false;
}
}


function ValidateEmail(uemail)
{
var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
if(uemail.value.match(mailformat))
{
return true;
}
else
{
alert("You have entered an invalid email address!");
uemail.focus();
return false;
}
}


function validsex(umsex,ufsex)
{
x=0;

if(umsex.checked)
{
x++;
} if(ufsex.checked)
{
x++;
}
if(x==0)
{
```

```
alert('Select Male/Female');
umsex.focus();
return false;
}
else
{
alert('Form Succesfully Submitted');
window.location.reload()
return true;
}
}

</script>

</head>
<body onload="document.registration.userid.focus();">
<h1>Registration Form</h1>
Use tab keys to move from one input field to the next.
<form name='registration' onSubmit="return formValidation();">

<label for="userid">User id:</label>
<input type="text" name="userid" size="12" /><br><br>
<label for="passid">Password:</label>
<input type="password" name="passid" size="12" /><br><br>
<label for="username">Name:</label>
<input type="text" name="username" size="50" /><br><br>
<label for="address">Address:</label>
<input type="text" name="address" size="50" /><br><br>
<label for="country">Country:</label>
<select name="country">
<option selected="" value="Default">(Please select a country)</option>
<option value="AF">Australia</option>
<option value="AL">Canada</option>
<option value="DZ">India</option>
<option value="AS">Russia</option>
<option value="AD">USA</option>
</select><br><br>
<label for="zip">ZIP Code:</label>
<input type="text" name="zip" /><br><br>
<label for="email">Email:</label>
<input type="text" name="email" size="50" /><br><br>
<label id="gender">Sex:</label>
<input type="radio" name="msex" value="Male" /><span>Male</span>
<input type="radio" name="fsex" value="Female"
/><span>Female</span><br><br>
```

```
<label>Language:</label>
<input type="checkbox" name="en" value="en" checked
/><span>English</span>
<input type="checkbox" name="nonen" value="noen" /><span>Non
English</span><br><br>
<label for="desc">About:</label>
<textarea name="desc" id="desc"></textarea><br><br>
<input type="submit" name="submit" value="Submit" />

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

OUTPUT:


File Edit View History Bookmarks Tools Help
JavaScript Form Validation using a s
file:///C:/Users/student/Desktop/formvalidation.html#
Submit

Registration Form

Use tab keys to move from one input field to the next.
User id: john

Password: ••••••••

Name: John

Address: Chennai

Country: India

ZIP Code: 23463

Email: john@gmail.com

Sex: [✓ Male] Female

Language: [✓ English] Non English

About: Iam a student

RESULT:
Thus the form was validated successfully using javascript.

EX.NO: 3b
DATE:

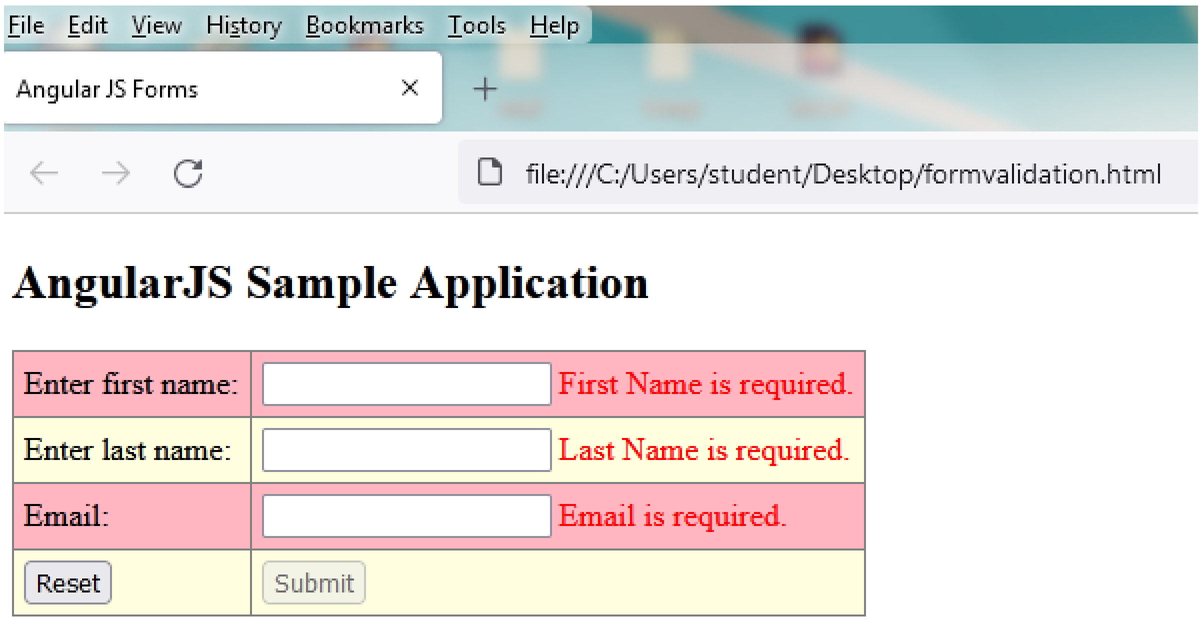
Angular JS application – Form validation

AIM:
To write a form using Angular JS for validation.
ALGORITHM:
1. Start.
2. Create a form in html which includes text field for name ,password.
3. Form also includes submit button.
4. When the button is clicked , onclick event call the function that validates the form.
5. End.

CODING:

```
<!DOCTYPE html>
 <html>
   <head>
      <title>Angular JS Forms</title>
      <script src = "http://ajax.googleapis.com/ajax/libs/angularjs/1.3.1
4/angular.min.js"></script>

 <style>
         table, th , td {
            border: 1px solid grey;
            border-collapse: collapse;
            padding: 5px;
         }

         table tr:nth-child(odd) {
            background-color: lightpink;
         }

         table tr:nth-child(even) {
            background-color: lightyellow;
         }
      </style>
   </head>
   <body>

      <h2>AngularJS Sample Application</h2>
      <div ng-app = "mainApp" ng-controller = "studentController">

         <form name = "studentForm" novalidate>
            <table border = "0">
               <tr>
                  <td>Enter first name:</td>
                  <td><input name = "firstname" type = "text" ng-
model = "firstName" required>
                     <span style = "color:red" ng-
show = "studentForm.firstname.$dirty && studentForm.firstname.$invalid">

                        <span ng-
show = "studentForm.firstname.$error.required">First Name is required.</s
pan>
                     </span>
                  </td>
               </tr>

               <tr>
                  <td>Enter last name: </td>
                  <td><input name = "lastname"  type = "text" ng-
model = "lastName" required>
                     <span style = "color:red" ng-
show = "studentForm.lastname.$dirty && studentForm.lastname.$invalid">
                        <span ng-
show = "studentForm.lastname.$error.required">Last Name is required.</spa
n>
                     </span>
                  </td>
               </tr>
               <tr>
                  <td>Email: </td><td><input name = "email" type = "email
" ng-model = "email" length = "100" required>
                     <span style = "color:red" ng-
show = "studentForm.email.$dirty && studentForm.email.$invalid">
                        <span ng-
show = "studentForm.email.$error.required">Email is required.</span>
                        <span ng-
show = "studentForm.email.$error.email">Invalid email address.</span>
                     </span>
                  </td>
               </tr>
                <tr>
                  <td>
                     <button ng-click = "reset()">Reset</button>
                  </td>
                  <td>
                     <button ng-
disabled = "studentForm.firstname.$dirty &&
                        studentForm.firstname.$invalid || studentForm.las
tname.$dirty &&
                        studentForm.lastname.$invalid || studentForm.emai
l.$dirty &&
                        studentForm.email.$invalid" ng-
click="submit()">Submit</button>
                  </td>
               </tr>
     </table>
         </form>
      </div>
        <script>
         var mainApp = angular.module("mainApp", []);
         mainApp.controller('studentController', function($scope) {
            $scope.reset = function(){
               $scope.firstName = "First Name";
               $scope.lastName = "Last Name";
               $scope.email = "xxxx@yyyy.com";
            }
             $scope.reset();
         });
      </script>
     </body>
</html>
```

OUTPUT:


File Edit View History Bookmarks Tools Help
Angular JS Forms
file:///C:/Users/student/Desktop/formvalidation.html

AngularJS Sample Application

Enter first name:	First Name is required.
Enter last name:	Last Name is required.
Email:	Email is required.
Reset	Submit

Result:

Thus the Angular JS application form was validated successfully.

EX.NO: 4
DATE:

APPLICATION USING REACT JS

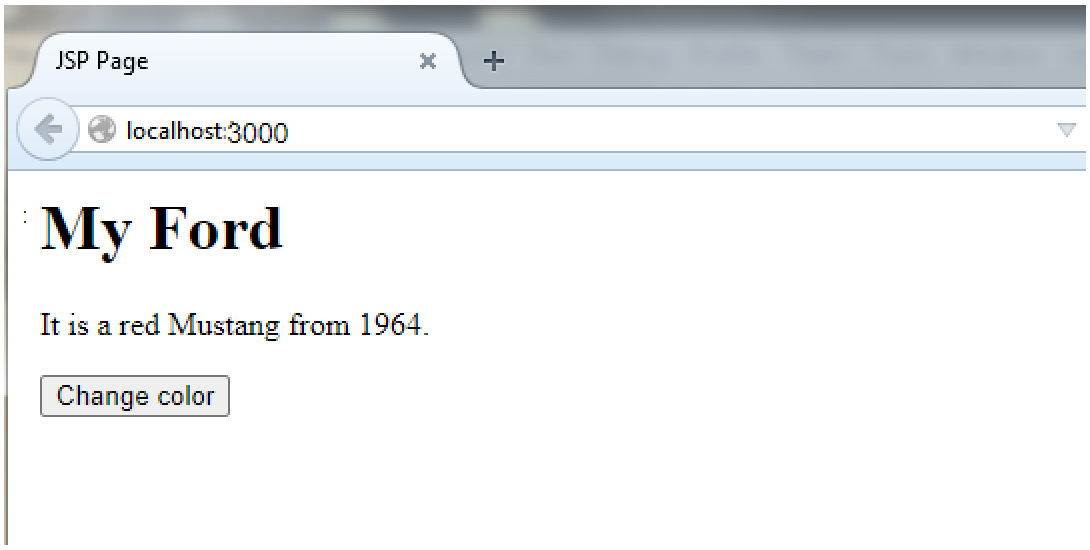
AIM:
To write a application using React JS for updating the state.

ALGORITHM:
1. Start.
2. Create a state.
3. When the button is clicked , onclick event changes the state of the component.
4. End.

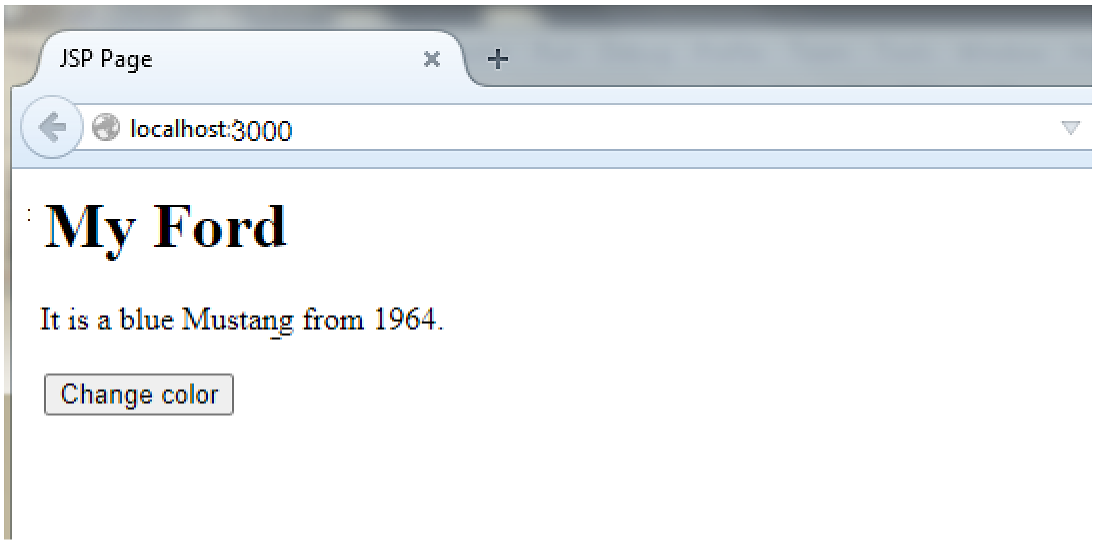
CODING:

```
import React from 'react';
import ReactDOM from 'react-dom';

class Car extends React.Component {
constructor(props) {
super(props);
this.state = {
     brand: "Ford",
     model: "Mustang",
     color: "red",
     year: 1964
};
}
changeColor = () => {
this.setState({color: "blue"});
}
render() {
return (
     <div>
            <h1>My {this.state.brand}</h1>
            <p>
            It is a {this.state.color}
            {this.state.model}
            from {this.state.year}.
            </p>
            <button
            type="button"
            onClick={this.changeColor}
            >Change color</button>
     </div>
);
}
}

ReactDOM.render(<Car />, document.getElementById('root'));
```

OUTPUT:



```
My Ford

It is a red Mustang from 1964.

Change color
```



```
My Ford

It is a blue Mustang from 1964.

Change color
```

Result:
Thus the program is executed successfully.

EX.NO: 5.a
DATE:

SERVLET WITH HTML

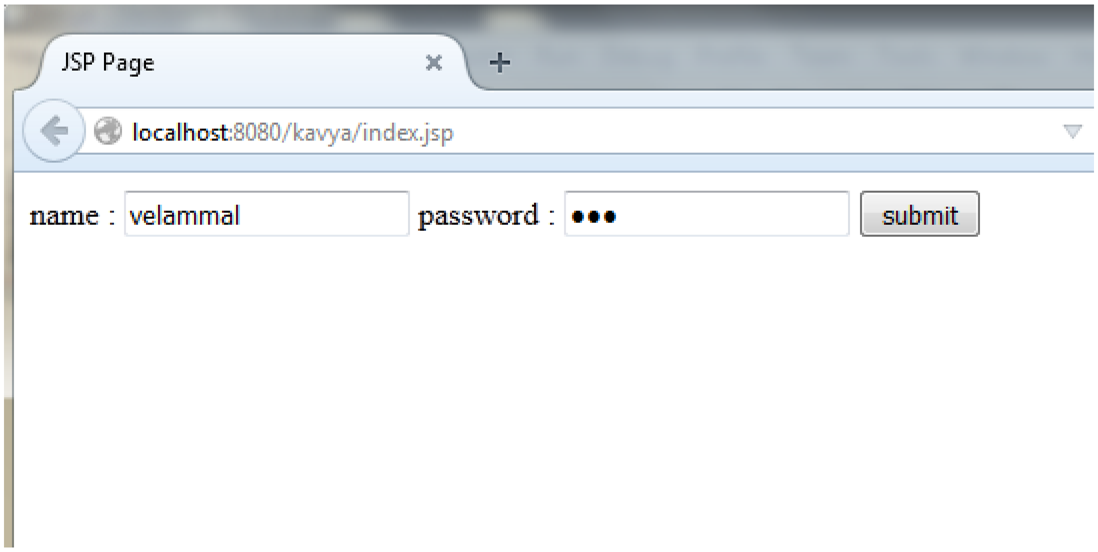
AIM:
To write a HTML code which invokes servlet.

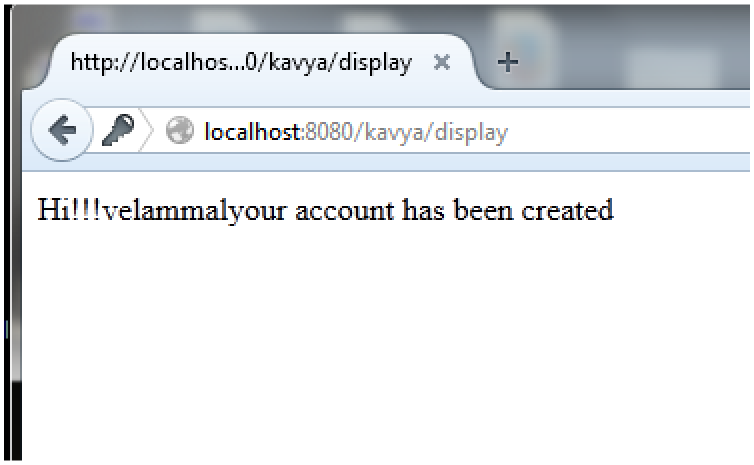
ALGORITHM:
1. Start the program.
2. Create a html page which gets two numbers from the user.
3. Use post method to pass the input to the servlet on submit.
4. Create the servlet code to add the values posted by the html.
5. Stop the program.

CODING:
HTML code
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<form action ="display" method ="post">
name : <input type ="text" name="name">
password : <input type ="password" name ="pass">
<input type ="submit" value="submit">
</form>
</body>
</html>
```

Servlet code
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name="display", urlPatterns={"/display"})
public class display extends HttpServlet {

        protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            String a = request.getParameter("name");
            out.println("Hi!!!" +a+ "your account has been created");
        } finally {
            out.close();
        }
    }
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
  }

}
```

OUTPUT:


JSP Page
localhost:8080/kavya/index.jsp
name : velammal password : ●●● submit


http://localhos...0/kavya/display
localhost:8080/kavya/display

Hi!!!velammalyour account has been created

RESULT:
Thus the HTML page invokes servlet successfully.

EX.NO: 5.b
DATE:

COOKIES

AIM:

To implement the concept of cookies in a servlet program.

ALGORITHM:

1.Write a basic HTML program.

2.Store the values typed/chosen by the user and store it in cookies

3.Retrieve the data fro cookies using the servlet program and display it.

4.Stop

PROGRAM:

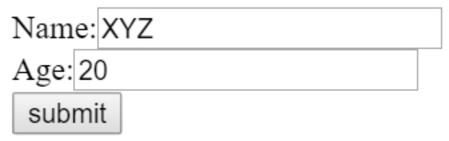
Html file

```
<html>
<head>
<title>Abc</title>
</head>
<body>
<form action="NewServlet" method="post">
Name:<input type="text" name="user"/><br>
Age:<input type="text" name="age"/><br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

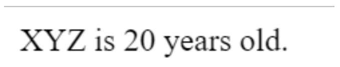
Servlet Program

```
import java.io.*;
importjavax.servlet.*;
importjavax.servlet.http.*;
public class NewServlet extends HttpServlet{
public void doPost(HttpServletRequestreq, HttpServletResponse
res)throws IOException,ServletException
    {
        String user=req.getParameter("user");
        Cookie cookie=new Cookie("m",user);
        String age=req.getParameter("age");
        Cookie cookie1=new Cookie("m",age);
res.addCookie(cookie);
res.setContentType("text/html");
PrintWriter out=res.getWriter();
out.println(user+" is "+age+" years old.");
cookie.setMaxAge(0);
    }
}
```

OUTPUT:



Name: XYZ
Age: 20



XYZ is 20 years old.

RESULT: Thus,the code was executed successfully.

EX.NO: 6

DATE:

JDBC CONNECTION WITH SERVLET

AIM:

To write a program for JDBC connection with servlet.

ALGORITHM:

1.Start.

2.Create a html program.

3.Create a jsp program.

4.Create a java program to establish connection.

5.Stop.

PROGRAM:

insert.html

```
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<form name="Insertion" action="insert.jsp">
        Name: <input type="text" name="firstName">
<input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>
```

insert.jsp

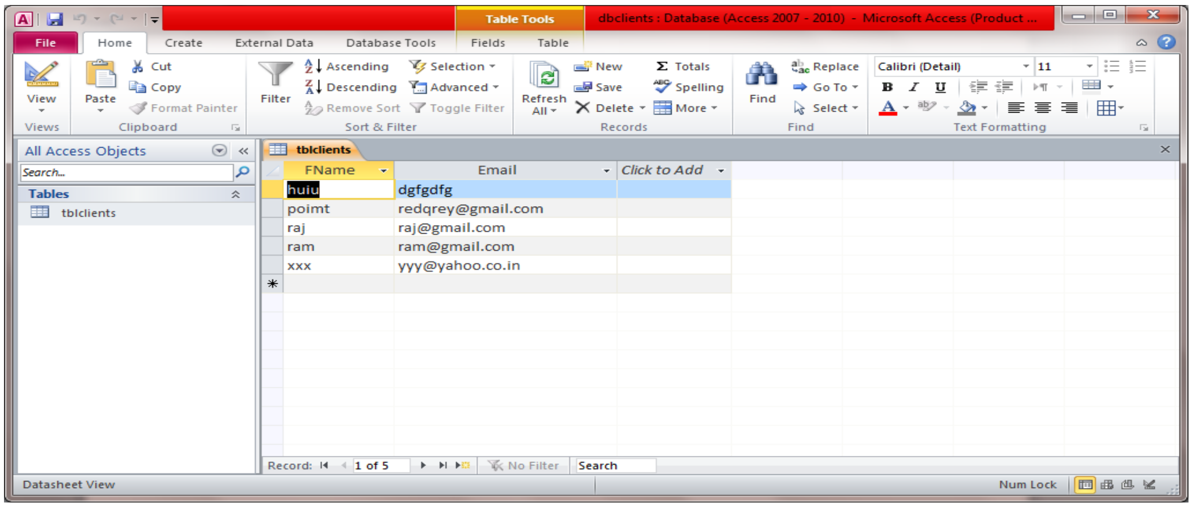
```
<%@ page import="org.pack.MyDB" %>
<html>
<body style="Background-color:#CCC0FF">

<% out.println("<h2>Clients List</h2>"); %>
<h3>----------------------------------------------------------------------
------------</h3>
<% String firstName = request.getParameter("firstName");
   String email = request.getParameter("email");
MyDBdbclient=new MyDB();
dbclient.insertdata(firstName,email);
out.println(dbclient.displayWithbrowser());
dbclient.closecon();
%>
<h3>----------------------------------------------------------------------
------------</h3>
</body>
</html>
```

MyDB.java

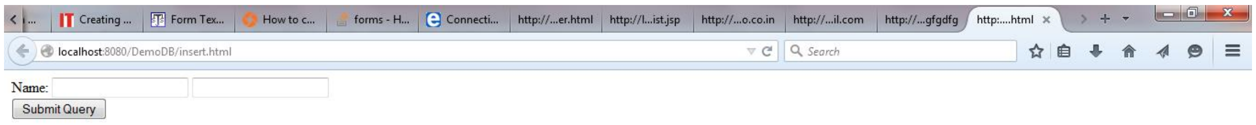
```
packageorg.pack;
importjava.sql.*;

public class MyDB {
private  Connection con;
private Statement statement;
private  ResultSetrs;
publicMyDB(){

try{
Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/M
YSQL", "root","vec");
statement=con.createStatement();
}catch(ClassNotFoundException e){System.out.println("Unable to
connect");
}catch(Exception se){System.out.println("sql error!");}
   }

public String displayWithbrowser(){
            String dbdata="<table>";
try{
```

```
rs=statement.executeQuery("Select * From tblclients");
while(rs.next()){
dbdata+="<tr>"+"<td style='width:300px'>" + rs.getString(1) +
"</td><td style='width:300px'>" + rs.getString(2) +  "</td></tr>";

            }

}catch(SQLException se){System.out.println("sql error!");}
returndbdata+"</table>";
   }

publicintinsertdata(String name, String email){
try{
statement.executeUpdate("INSERT INTO tblclients(FName,Email)
Values('"+name+"','"+email+"')");

}catch(SQLException se){System.out.println("sql error!");}
return 1;
 }

public void closecon(){
try{
con.close();
}catch(SQLException se){System.out.println("sql error!");}

  }
}
```

OUTPUT:

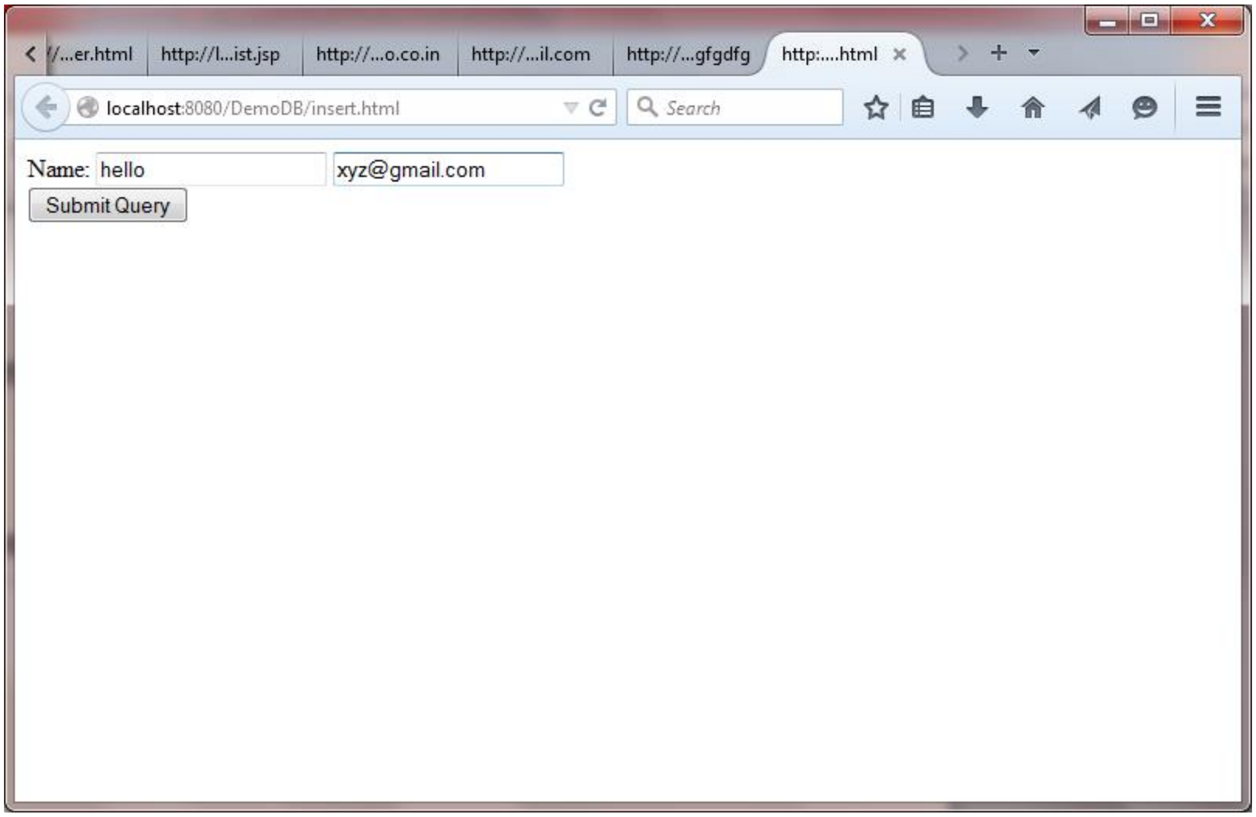


FName	Email
huiu	dgfgdfg
poimt	redqrey@gmail.com
raj	raj@gmail.com
ram	ram@gmail.com
xxx	yyy@yahoo.co.in



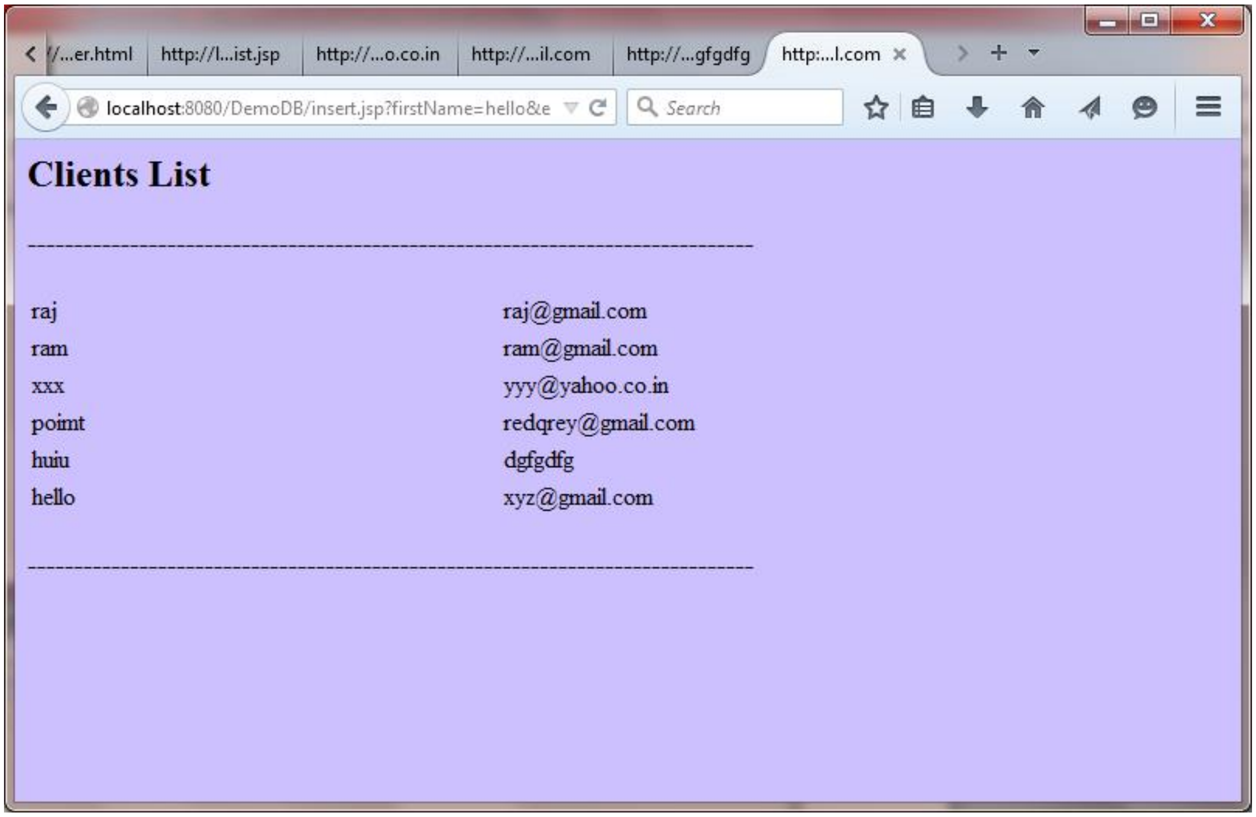
Name:

Submit Query



Name: hello xyz@gmail.com

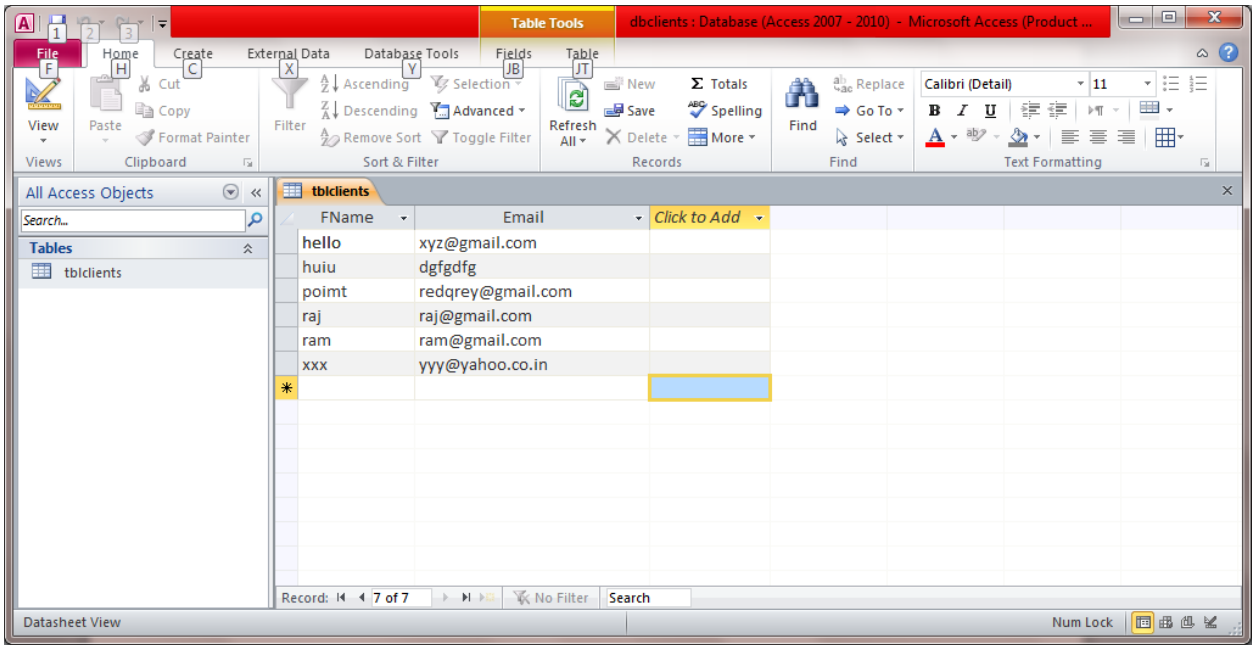
Submit Query



localhost:8080/DemoDB/insert.jsp?firstName=hello&e

Clients List

```
--------------------------------------------------------------------------------

raj                                   raj@gmail.com
ram                                   ram@gmail.com
xxx                                   yyy@yahoo.co.in
poimt                                 redqrey@gmail.com
huiu                                  dgfgdfg
hello                                 xyz@gmail.com

--------------------------------------------------------------------------------
```



dbclients : Database (Access 2007 - 2010) - Microsoft Access (Product ...

tblclients

FName	Email
hello	xyz@gmail.com
huiu	dgfgdfg
poimt	redqrey@gmail.com
raj	raj@gmail.com
ram	ram@gmail.com
xxx	yyy@yahoo.co.in

Record: 7 of 7

RESULT:

Thus the above program was executed successfully.

EX.NO: 7

DATE:

BASIC CALCULATOR USING JSP

AIM:

To write a HTML code which invokes JSP.

ALGORITHM:

1. Start the program.

2. Create a html page which gets two numbers from the user.

3. Use submit buttonto pass the input to the JSP on submit.

4. Create the JSP code to perform arithmetic operations to the values posted by the html.

5. Stop the program.

CODING

HTML code

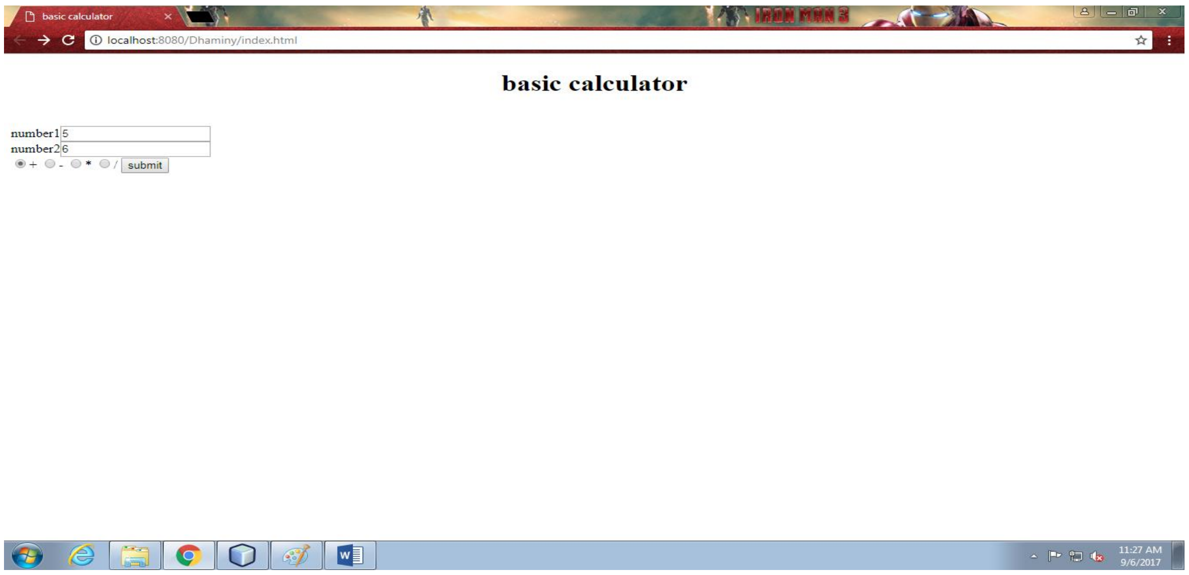
```
<html>
<head>
<title> BASIC CALCULATOR</title>
</head>
<body>
<h1><center>basic calculator</center></h1>
<form action="eg.jsp"><br>
<label>
                number1<input type="text" name="number1"
value="num1"><br>
</label>
<label>
                number2<input type="text" name="number2"
value="num2"><br>
</label>
<input type="radio" name="r1" value="add">+
<input type="radio" name="r1" value="sub">-
<input type="radio" name="r1" value="mul">*
<input type="radio" name="r1" value="div">/
<input type="submit" value="submit">
```

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

JSP code

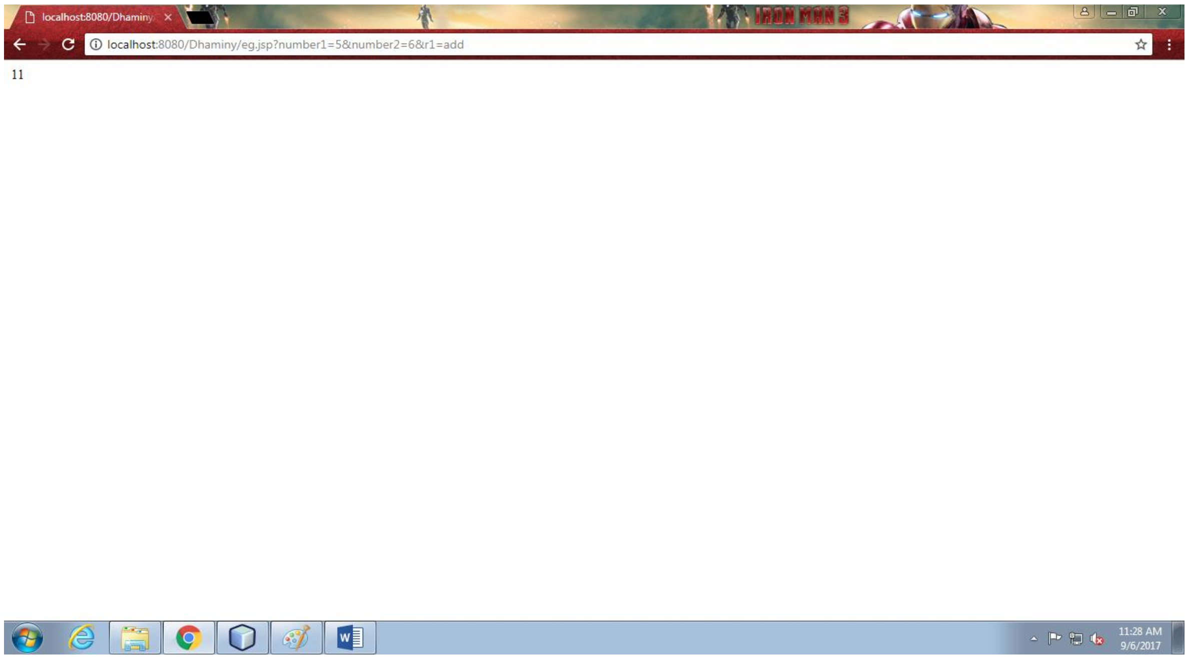
```
<html>
<body>
<% String s1=request.getParameter("number1");
        String s2=request.getParameter("number2");
        int a,b;
        a=Integer.parseInt(s1);
        b=Integer.parseInt(s2);
        String operation=request.getParameter("r1");
        if(operation.equals("add"))
        {
            int x;
            x=a+b;
            out.println(x);
        }
        else if(operation.equals("sub"))
        {
            int x;
            x=a-b;
            out.println(x);
        }
        else if(operation.equals("mul"))
        {
            int x;
            x=a*b;
            out.println(x);
        }
        else
        {
            int x;
            x=a/b;
            out.println(x);
        }
        %>
</body>
</html>
```

OUTPUT:



basic calculator

number1 5
number2 6
+ - * / submit



```
11
```

RESULT:

Thus the HTML page invokes JSP successfully

EX.NO: 8

DATE:

XML SCHEMA - XSLT / XSL

AIM:

Program using XML-Schema-XSLT/XSL

ALGORITHM:

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

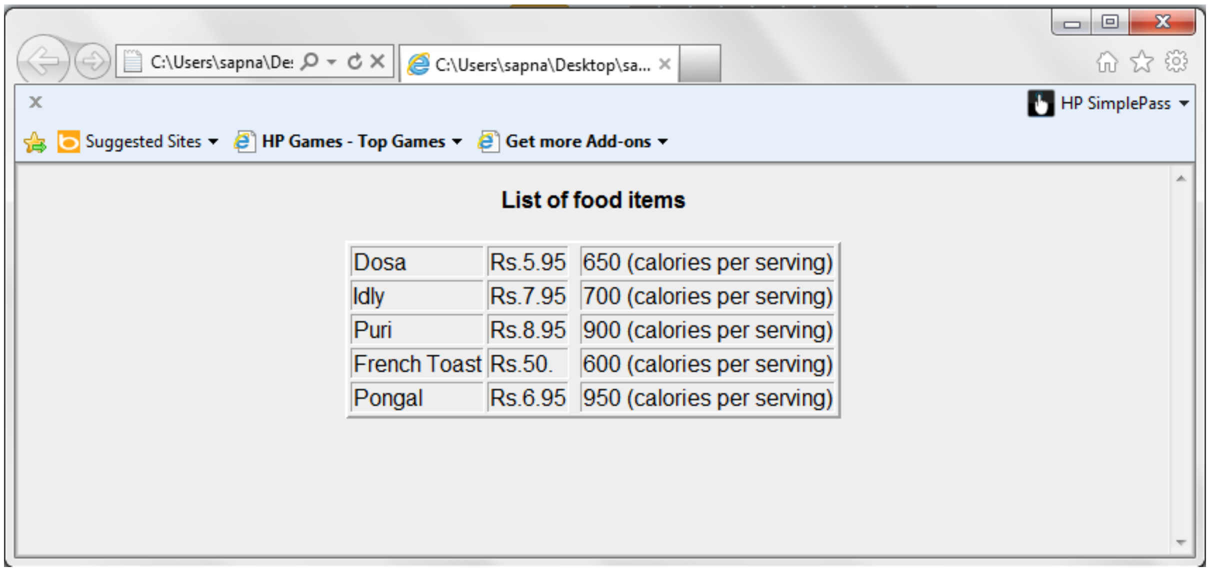
XML CODE

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="eg.xsl"?>
<breakfast_menu>
      <food>
            <name>Dosa</name>
            <price>Rs.5.95</price>
            <calories>650</calories>
      </food>
      <food>
            <name>Idly</name>
            <price>Rs.7.95</price>
            <calories>700</calories>
      </food>
      <food>
            <name>Puri</name>
            <price>Rs.8.95</price>
            <calories>900</calories>
      </food>
      <food>
            <name>French Toast</name>
            <price>Rs.50.</price>
            <calories>600</calories>
      </food>
      <food>
            <name>Pongal</name>
            <price>Rs.6.95</price>
            <calories>950</calories>
      </food>
</breakfast_menu>
```

XSLT CODE

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<html xsl:version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml">
<body style="font-family:Arial;font-size:12pt;background-
color:#EEEEEE">
<center>
<h4>List of food items</h4>
<table border="2 px">
<xsl:for-each select="breakfast_menu/food">
      <tr>
<td><xsl:value-of select="name"/></td>
<td><xsl:value-of select="price"/></td>
<td><xsl:value-of select="description"/></td>
<td><xsl:value-of select="calories"/>(calories per serving)</td>
      </tr>
</xsl:for-each>
      </table>
      </center>
</body>
</html>
```

OUTPUT:



List of food items

Dosa	Rs.5.95	650 (calories per serving)
Idly	Rs.7.95	700 (calories per serving)
Puri	Rs.8.95	900 (calories per serving)
French Toast	Rs.50.	600 (calories per serving)
Pongal	Rs.6.95	950 (calories per serving)

RESULT:

Thus Programs using XML-Schema-XSLT/XSL was developed and successfully executed

EX.NO: 9.a

DATE:

FORM VALIDATION USING PHP REGULAR EXPRESSION

AIM:

To write a PHP Program to validate a form

ALGORITHM:

Step 1: Start the Program

Step 2:Create a regular expression pattern

Step 3:Check whether the input string matches the regular expression pattern

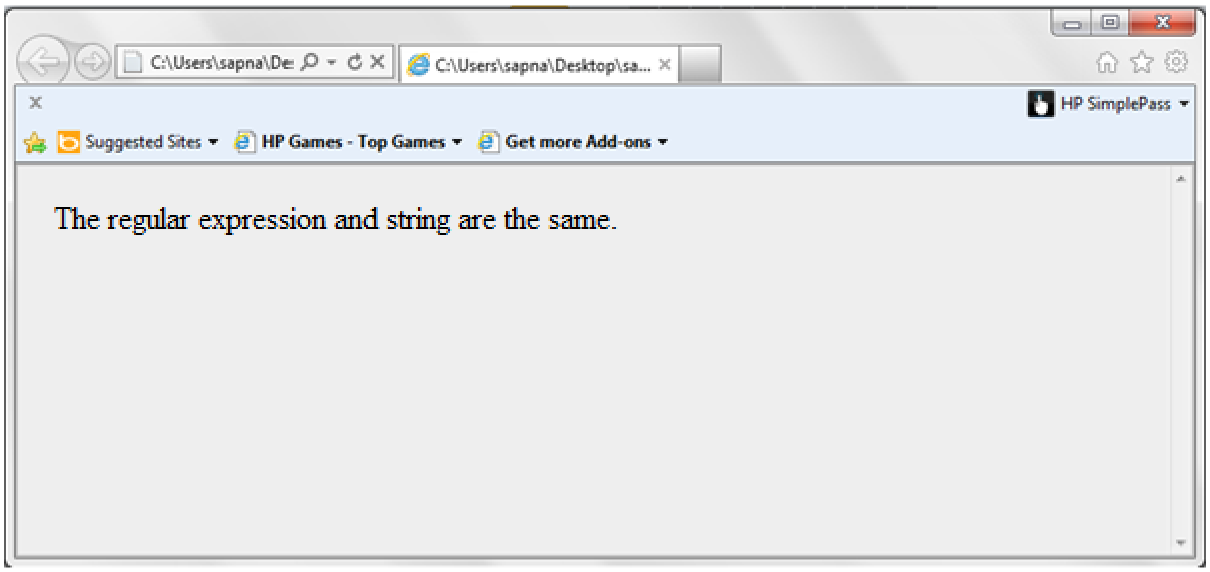
Step 4:If it matches display that the string are same, otherwise display the input does not match with the regular expression.

Step 5:Stop

CODE:

```
<?php
$regExp = "/[a-zA-Z]+ \d+/";
if (preg_match($regExp, "Januaury 26"))
// This statement matches the regular expression with the stri
ng, If mathces then if statement executes, otherwise else stat
ment.
{
    echo "The regular expression and string are the same.";
} else
{
    echo "The regex pattern does not match with the string.";

}

?>
```

OUTPUT:



```
The regular expression and string are the same.
```

RESULT:

Thus the above program was executed successfully.

EX.NO: 9.b

DATE:

PHP APPLICATION CONNECTED TO THE DATABASE

AIM:

To write a program in PHP connection with MySQL.

ALGORITHM:

1.Start.

2.Create a html program.

3.Create a PHP program and connect it to MySQL database.

4.Insert the form data into the table in MySQL.

5.Stop.

PROGRAM:

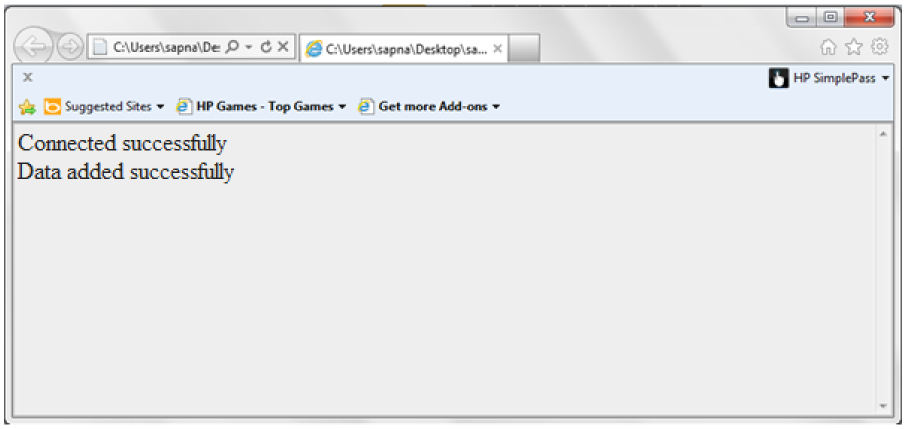
Form.html

```
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
</head>
<body>
<form name="Insertion" action="s2.php" method="POST">
        User Name: <input type="text" name="uname">
<input type="password" name="pwd"><br>
<input type="submit">
</form>
</body>
</html>
```

s2.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "cse";
$database = "db";
// Create connection
$conn = new mysqli($servername, $username, $password,
$database);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

echo "Connected successfully<br>";

$name = $_POST['uname'];
$pwd = $_POST['pwd'];

$sql = "INSERT INTO tbl (user, pass)
VALUES ('$name','$pwd')";
if ($conn->query($sql) === TRUE) {
  echo "Data added successfully";
} else {
  echo "Error adding data: " . $conn->error;
}

?>
```

OUTPUT:



```
Connected successfully
Data added successfully
```

RESULT:

Thus the program in PHP to connec to MySQL was executed successfully

EX.NO: 10

DATE:

WEB SERVICE BASED CALCULATOR

AIM:

To demonstrate web service based calculator.

ALGORITHM:

Step 1. Start

Step 2. Create a new project under Java Web→Web Application.

Step 3. Right click on the source package and create new Web Service.

Step 4. Define a method add () with its parameters and operations.

Step 5. Right click on the project name and select Clean and Build. After completion, right click again on the project name and select deploy.

Step 6. Right click on Web Services and select Test Web Service to run the program.

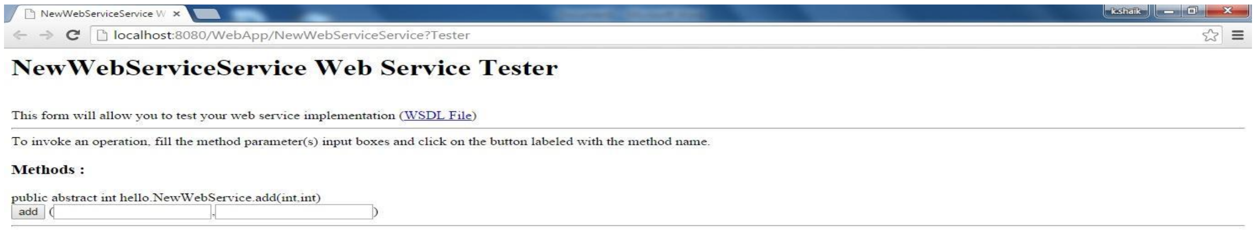
Step 7. End

CODING:

NewWebService.java

```
package hello;

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

@WebService()

public class NewWebService {
@WebMethod(operationName = "add")

public int add(@WebParam(name = "i")
int i, @WebParam(name = "j")
int j) {

int k=i+j;
return k;

    }
}
```

OUTPUT:



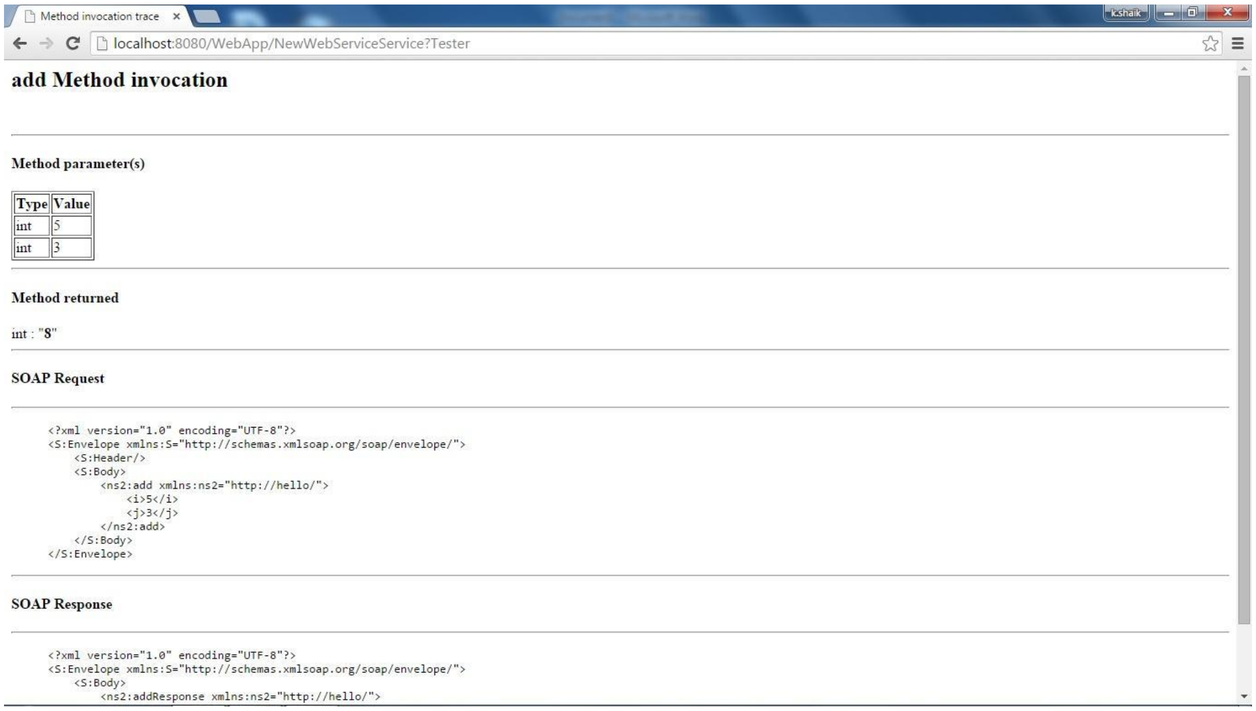
NewWebServiceService Web Service Tester

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 hello.NewWebService.add(int,int)
add (,)



add Method invocation

Method parameter(s)

Type	Value
int	5
int	3

Method returned

int : "8"

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://hello/">
            <i>5</i>
            <j>3</j>
        </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://hello/">
```

RESULT:

Thus, the program to demonstrate webserver based calculator was executed and the output was verified successfully.

CONTENT BEYOND SYLLABUS

EX.NO:11

DATE:

INSTALLATION OF APACHE TOMCAT SERVER

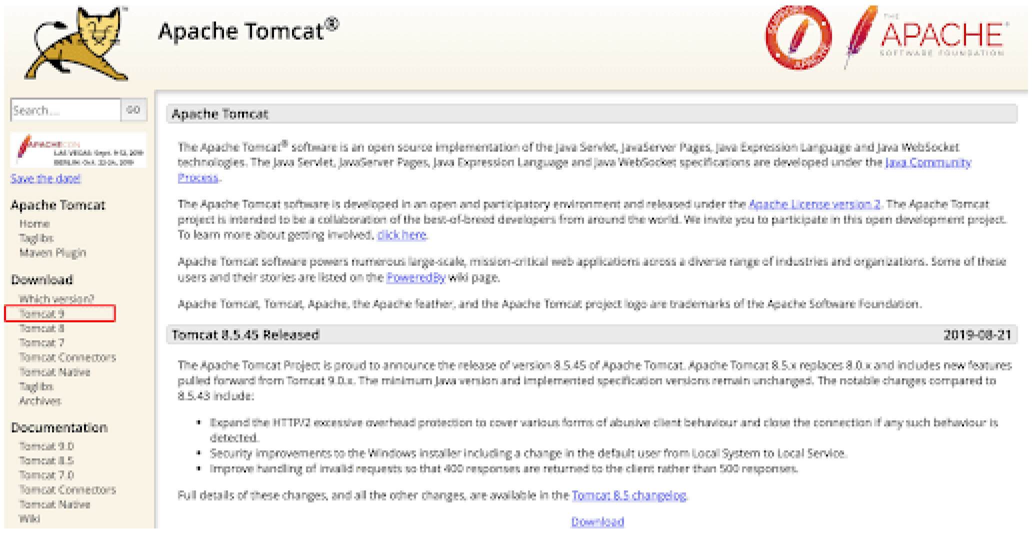
Steps to Install Apache Tomcat Server:

Prerequisites

JDK or JRE will need to be installed on the Windows Server before you can configure Tomcat 9 on the server. OpenJDK and Amazon Corretto are two examples of open-source Java Development Kit providers.

Installing Tomcat 9

Open your browser and head over to https://tomcat.apache.org.
Scroll down a little to locate and click on the Tomcat 9 link located within the left menu bar.



Apache Tomcat

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process.

The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, click here.

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat 8.5.45 Released 2019-08-21

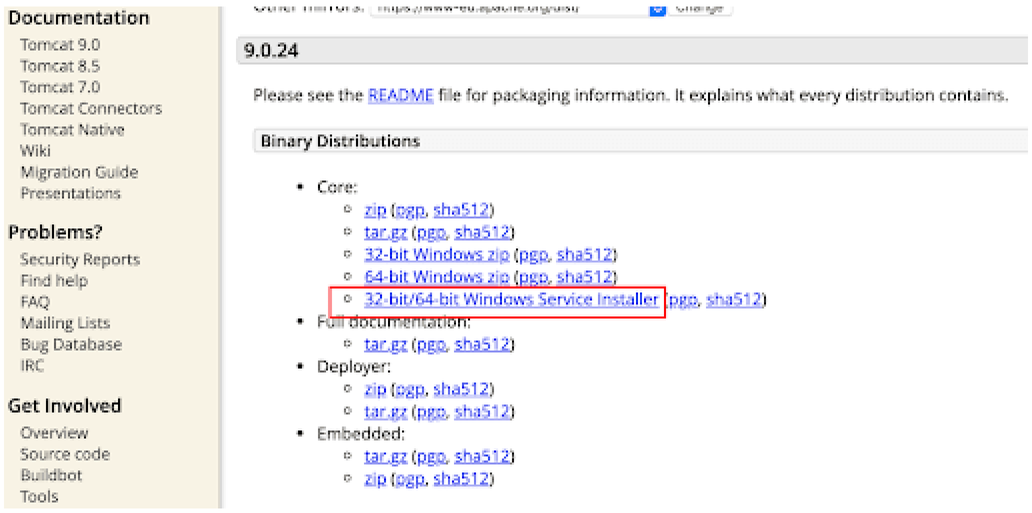
The Apache Tomcat Project is proud to announce the release of version 8.5.45 of Apache Tomcat. Apache Tomcat 8.5.x replaces 8.0.x and includes new features pulled forward from Tomcat 9.0.x. The minimum Java version and implemented specification versions remain unchanged. The notable changes compared to 8.5.43 include:

- Expand the HTTP/2 excessive overhead protection to cover various forms of abusive client behaviour and close the connection if any such behaviour is detected.
- Security improvements to the Windows installer including a change in the default user from Local System to Local Service.
- Improve handling of invalid requests so that 400 responses are returned to the client rather than 500 responses.

Full details of these changes, and all the other changes, are available in the Tomcat 8.5 changelog.

Download

Next, locate the 32-bit/64-bit Windows Service Installer link and click on it.


Documentation

- Tomcat 9.0
- Tomcat 8.5
- Tomcat 7.0
- Tomcat Connectors
- Tomcat Native
- Wiki
- Migration Guide
- Presentations

Problems?

- Security Reports
- Find help
- FAQ
- Mailing Lists
- Bug Database
- IRC

Get Involved

- Overview
- Source code
- Buildbot
- Tools

9.0.24

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

This link will open the Windows Service Installer automatically. Once the installer window pops up, you will initiate the install by clicking on the Start Download button. Once the download completes, double click the .exe installer to begin the installation.

The install wizard should only take a few moments to complete. After the install wizard has finished, it will ask you to make a few different decisions along the way.

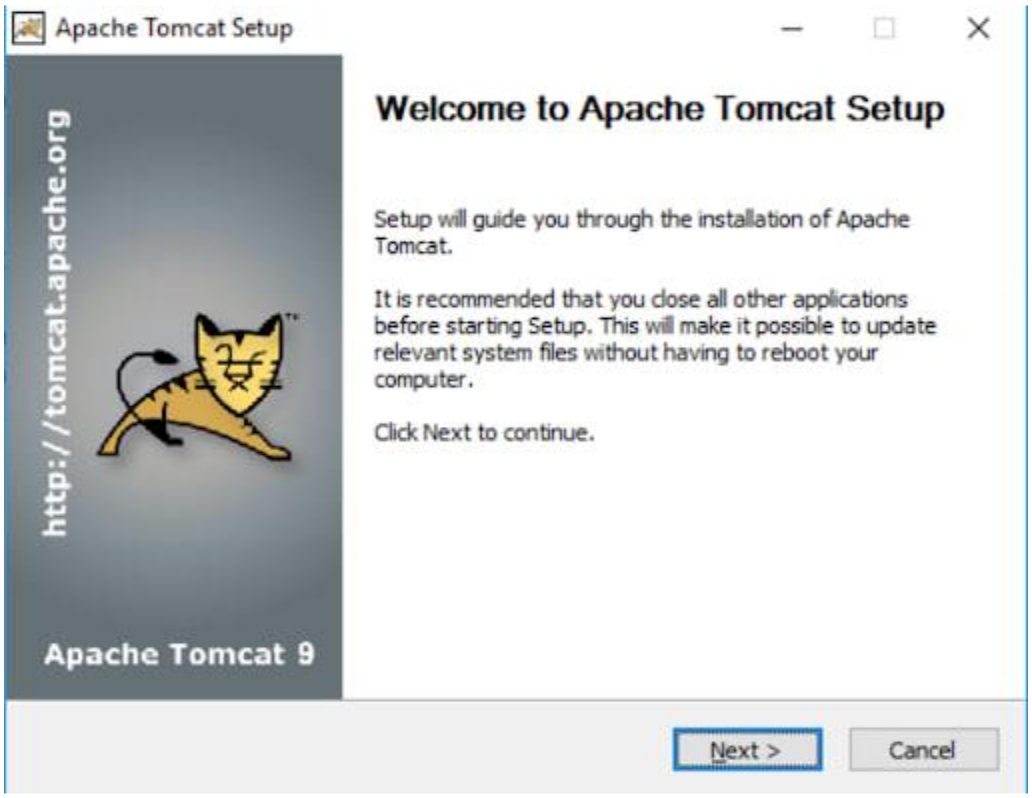
Below are the following choices you will be asked to make to install Tomcat 9.

Please note you can configure Tomcat 9 how you would like but for the purpose of this demonstration, we will be installing it with default settings.

Installation

Step 1.

The first page of the install explains what the installer will do and what to expect. Go ahead and click Next on the first page.


Apache Tomcat Setup

Welcome to Apache Tomcat Setup

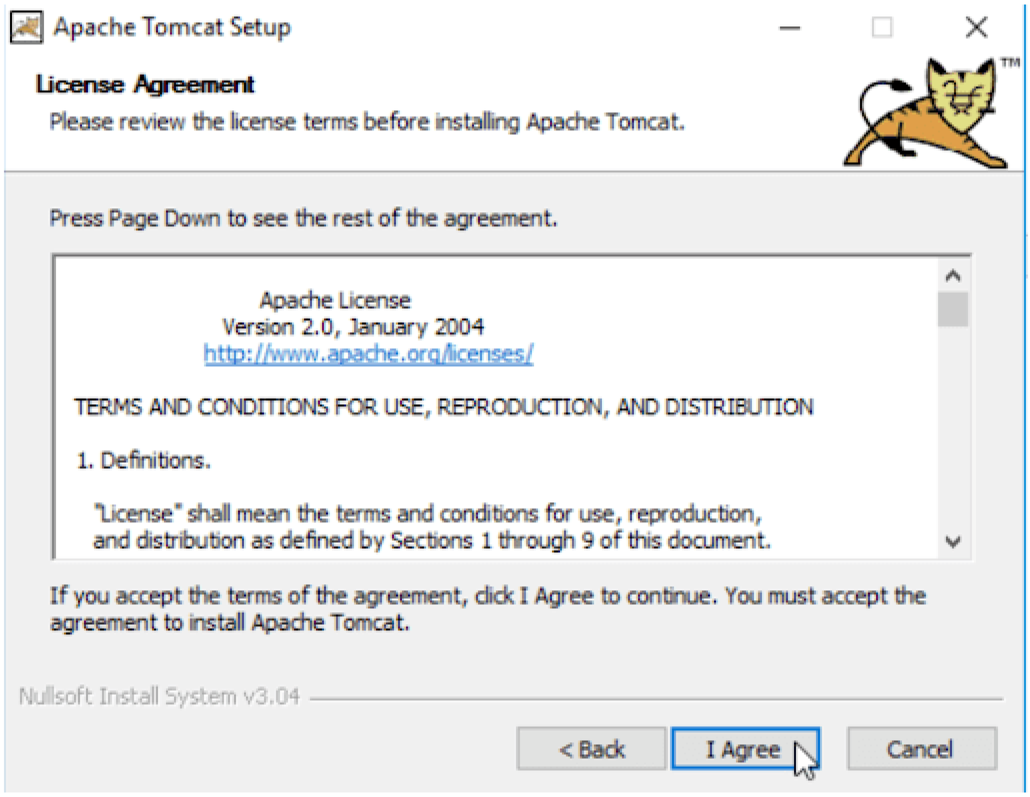
Setup will guide you through the installation of Apache Tomcat.

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.

Step 2.

Before you can start the install, you must agree to the Apache License Agreement for the Tomcat 9 service


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.

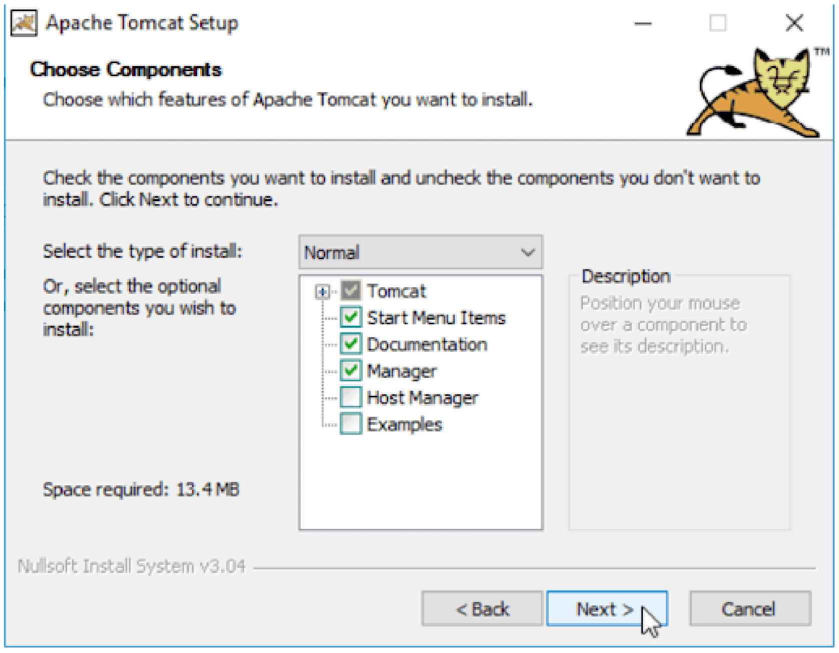
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

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

Step 3.

Click on "Select the type of install" dropdown list and choose the "Full" install option and then click Next.


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.

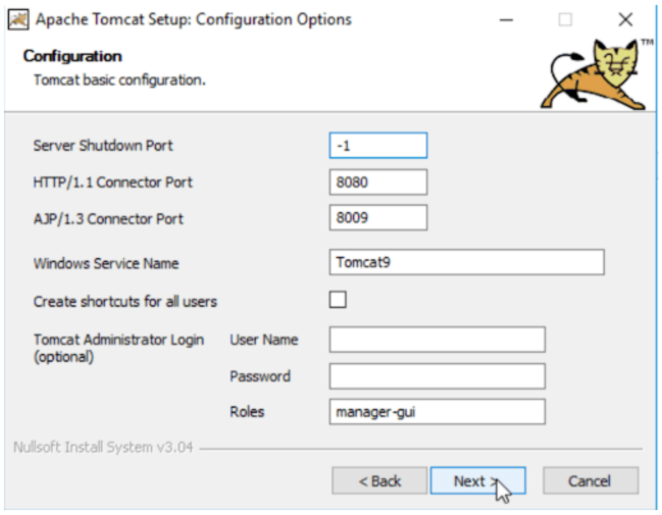
Space required: 13.4 MB

Nullsoft Install System v3.04

Step 4.

The next screen is the configuration screen. This screen will allow you to set up any default ports that you want the service to connect through, and will also allow you to set an Administrator username and password.

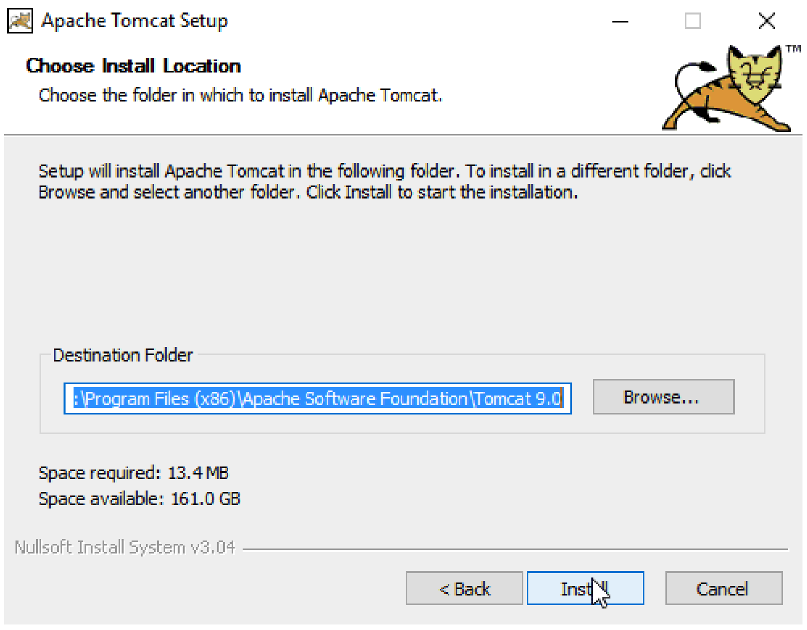
Please note that you can configure these settings later if you choose to.
Choose your selections and click Next.


Apache Tomcat Setup: Configuration Options
Configuration
Tomcat basic configuration.
Nullsoft Install System v3.04
< Back
Next >
Cancel

Server Shutdown Port		-1
HTTP/1.1 Connector Port		8080
AJP/1.3 Connector Port		8009
Windows Service Name		Tomcat9
Create shortcuts for all users		☐
Tomcat Administrator Login (optional)	User Name	
	Password	
	Roles	manager-gui

Step 5.

Next, you will choose the location where you want to install the Tomcat 9 service. Click Next.



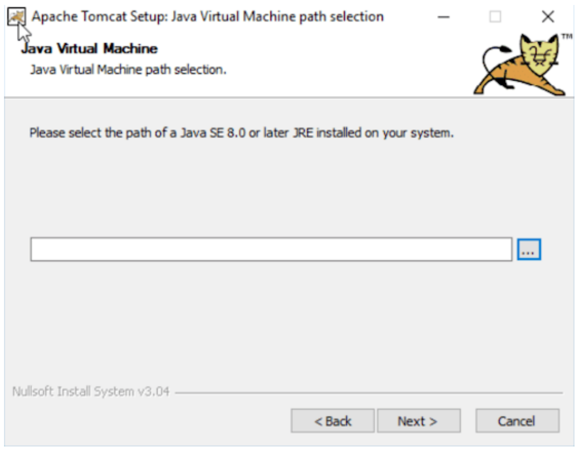
Setup will install Apache Tomcat in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

:\Program Files (x86)\Apache Software Foundation\Tomcat 9.0

Space required: 13.4 MB
Space available: 161.0 GB

Step 6.

To correctly install Tomcat 9 on your server, the Wizard will want you to choose the location of any other Java related software on your server.


Apache Tomcat Setup: Java Virtual Machine path selection

Java Virtual Machine

Java Virtual Machine path selection.

Please select the path of a Java SE 8.0 or later JRE installed on your system.

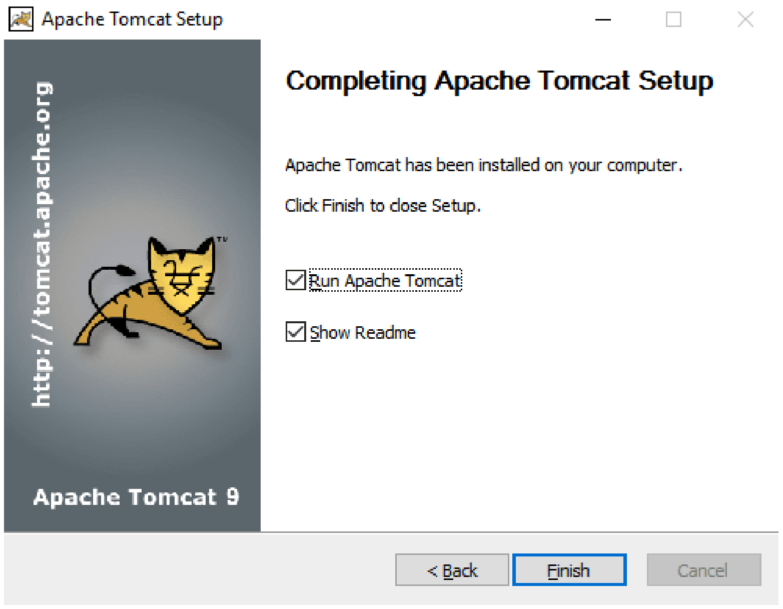
Nullsoft Install System v3.04

Step 7.

Once the install is complete, click Next.

Lastly, uncheck the Readme checkbox.

If you prefer to start Apache Tomcat 9 after the install, leave the checkbox marked. If not, then you will want to uncheck that checkbox.


Apache Tomcat Setup

Completing Apache Tomcat Setup

Apache Tomcat has been installed on your computer.

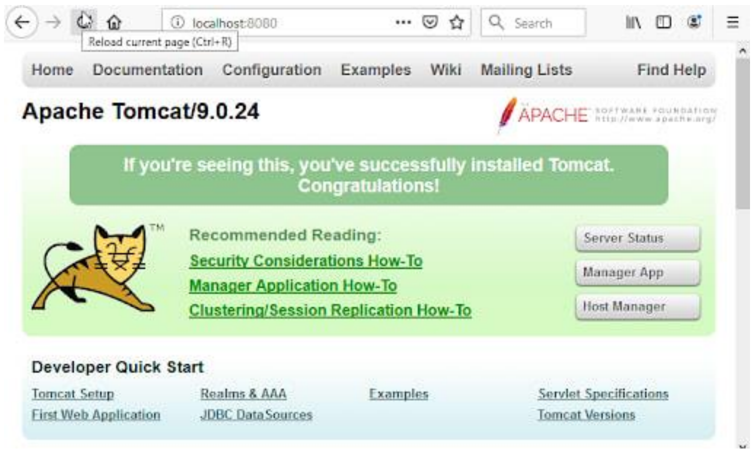
Click Finish to close Setup.

☑ Run Apache Tomcat

☑ Show Readme

To make sure the service is running, go to the Windows startup menu and type services.cmd. From the list of available services, find the Apache Tomcat 9 service, right-click on the service name, select Start, and make sure the service starts successfully. You should see a “running” status next to the service name.

To test the Tomcat 9 install and verify it is running on your server, open and point your browser to localhost:8080 (or whatever custom port you put into the configuration).



Apache Tomcat/9.0.24

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

The page that loads will tell you if Tomcat 9 was successfully installed.

EX.NO: 12

DATE:

LOCK SERVLET

AIM:

To write a program to lock servlet.

ALGORITHM:

1.Start.

2.Create a program to lock servlet itself to a particular server IP address and port number.

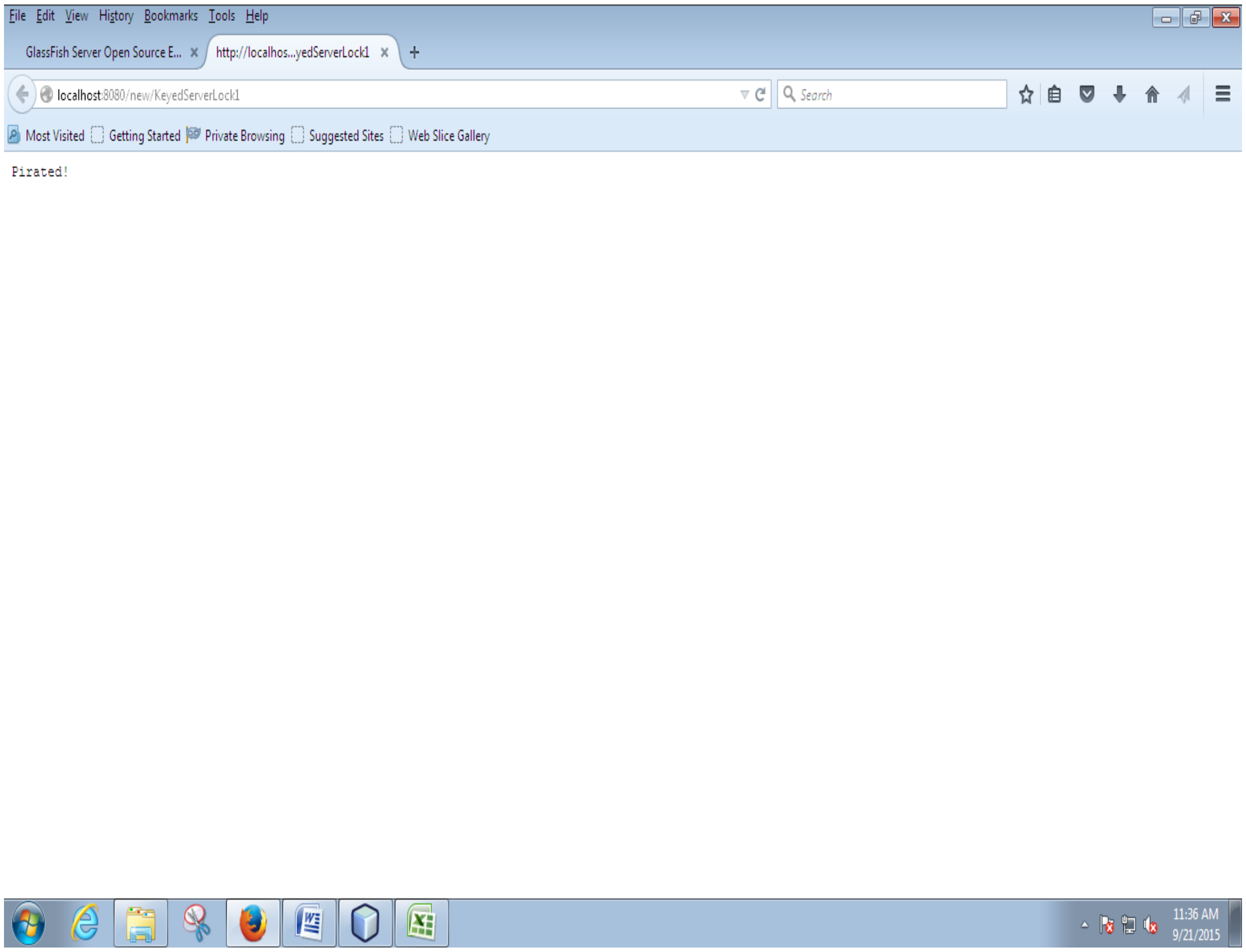
3.Use an init parameter key that is appropriate for its servlet IP address and port before it unlocks itself and handles a request.

4.Stop the program.

CODING:

```
import java.io.*;
import java.net.*;
import java.util.*;
import javax.servlet.*;
public class KeyedServerLock extends GenericServlet
{
 public void service(ServletRequest req, ServletResponse res)throws
ServletException, IOException
{
 res.setContentType("text/plain");
 PrintWriter out = res.getWriter();
 String key = getInitParameter("key");
 String host = req.getServerName();
 int port = req.getServerPort();
 if (! keyFitsServer(key, host, port))
{
 out.println("Pirated!");
 }
 else
 {
 out.println("Valid");
 }
 }
 private boolean keyFitsServer(String key, String host, int port)
 {
 if (key == null) return false;
 long numericKey = 0;
try
 {
 numericKey = Long.parseLong(key);
 }
 catch (NumberFormatException e)
{
 return false;
 }
 byte hostIP[];
 try
{
hostIP = InetAddress.getByName(host).getAddress();
}
catch (UnknownHostException e)
{
return false;
}
long servercode = 0;
for (int i = 0; i < 4; i++)
{
servercode <<= 8;
servercode |= hostIP[i];
}
servercode <<= 32;
servercode |= port;
 long accesscode = ~numericKey;
return (servercode == accesscode);
}
}
```

OUTPUT:


File Edit View History Bookmarks Tools Help
GlassFish Server Open Source E...
http://localhos...yedServerLock1
localhost:8080/new/KeyedServerLock1
Search
Most Visited
Getting Started
Private Browsing
Suggested Sites
Web Slice Gallery
11:36 AM
9/21/2015

```
Pirated!
```

RESULT: Thus the above program was executed successfully.