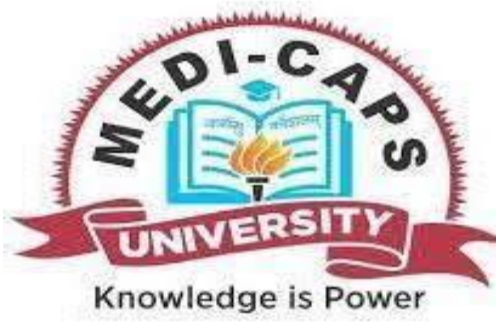
MEDI - CAPS UNIVERSITY, INDORE



DEPARTMENT

OF

COMPUTER SCIENCE & ENGINEERING

Lab Manual

Subject: Internet and Web technology

Subject Code: CS3ES12

(SOFTWARE WORKSHOP- I)

SESSION: 2021-22

INDEX

SL. No.	EXPERIMENT NAME	Page No.
1	Write a program HTML to Create menu & formatting using CSS. Formatting content using CSS stylesheet, background images, colors and properties, also apply some mouse over event.	4-11
2	Write a Program to Create User registration form and Login Page using simple HTML, Basic HTML application using TAG like hyperlink, lists, tables, images, simple HTML forms, website's structure	12-15
3	Write a Program to create a static webpage using table tags of HTML	16-20
4	Write a Program to create a HTML page, which has properly aligned paragraphs with image along with it	21-26
5	Write a Program to Create a form with various fields and appropriate front and validations using any one of the scripting languages	27-40
6	Write a Program to develop and demonstrate the usage of inline, internal and external style sheet using CSS.	41-47
7	Write a program to design a simple calculator using JavaScript	48-52
8	Write a program JavaScript to validate the following fields of the Registration page	53-60
9	Write a program JavaScript to create and read value from cookie on the web pages	61-65
10	Write an XML file which will display the Book information which includes the following: (a) Title of the book (b) Author Name (c) ISBN number (d) Publisher name (e) Edition (f) Price	66-73
11	Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings	74-76
12	Write a PHP program to display a digital clock which displays the current time of the server.	77-80
13	Write a Servlet application to print the current date and time	81-83
14	Write a JSP application to print the current date and time	84-87

SOFTWARE AND HARDWARE REQUIREMENTS FOR A BATCH OF 30 STUDENTS

HARDWARE:

- Intel Desktop Systems: 35 nos.

SOFTWARE:

- **System Software:** Microsoft Windows 7 Academic Get Genuine Legalization License
- **Application Software's:** MS Office, Notepad/ Notepad+/ Atom, Antivirus (Symantec Endpoint Protection 12.1.5), , XAMPP Server.

EXPERIMENT-1

TITLE: Write a HTML Program to Create menu & formatting using CSS. Formatting content using CSS stylesheet, background images, colors and properties, also apply some mouse over event.

Outcome: To Learn how to combine basic HTML elements to create Web pages.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1 Develop web pages using HTML and using cascading styles sheets.		√	√									

1. DESCRIPTION ABOUT EXPERIMENT:

1.1. Hyper Text Markup Language (HTML): It is use for design a client-side web pages, this is a static page it means you can only view Html page not give request and not get response from server using html page. Using Html pages browser get user information through form (This is an Html element). Html provides so many elements (like <p>, <img>, <h1>) to design a web page.

1.2. HTML Common tags:

- HTML is the building block for web pages. HTML is a format that tells a computer how to display a web page.
- The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.
- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension.

1.3. HTML Tags:

- HTML tags are used to mark-up HTML elements
- .HTML tags are surrounded by the two characters < and >.
- The surrounding characters are called angle brackets.
- HTML tags normally come in pairs like and the first tag in a pair is the start tag, the second tag is the end tag.
- The text between the start and end tags is the element content.
- HTML tags are not case sensitive, <B>means the same as<b>.

1.3 The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, , <p> etc.
<p>	This tag represents a paragraph.
<h1> to <h6>	Defines header 1 to header 6
 	Inserts a single line break
<hr>	Defines a horizontal rule
<b>	It is used to make a text bold.
<img>	It is used to insert an image within an HTML document.
	It is used to align the content in center
	It defines a division or section within HTML document.
<meta>	It defines metadata of an HTML document.
<!-->	Defines a comment

1.4. Cascading Style Sheets (CSS): It is a rule-based language that applies styling to HTML elements. We write CSS rules in Html elements (<p>, <img>), and modify properties of those elements such as color, background color, width, border thickness, font size, etc.

1.5. CSS rule, is made up of two parts:

- **Selector Part:** Identifies the HTML elements that the rule will be applied to, identified by the actual element name, Ex: <body>, <p>, <h1>, etc.
- **Declaration part**: It contains property and value. Example: suppose that we want size of our text 10px then it declare as font-size:10px. Here font-size is properties and 10px is there value, and all this declaration is called declaration.

Syntax:

```
h1 {
    color:red;
    font-size:10px;
}
```

2. PROCEDURE:

STEP 1: Start with content. As a starting point, we'll write up raw text content and see what browsers do with it.

STEP 2: Give the document structure. You'll learn about HTML element syntax and the elements that give a document its structure.

STEP 3: Identify text elements. You'll describe the content using the appropriate text elements and learn about the proper way to use HTML.

STEP 4: Add an image. By adding an image to the page, you'll learn about attributes and empty elements.

STEP 5: Change the page appearance with a style sheet. This exercise gives you a taste of formatting content with Cascading Style Sheets.

3. PROGRAM:

```
<!DCOTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

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

<link href="mytheme.css" rel="stylesheet"type="text/css"/>

<title> Web Programming</title>

</head>

<body>

<div id="Header"></div>

<div id="logo">

<img src="Logo.jpeg" alt="MU" style="width:200px; height:200px;">

</div>

<div id="title">

MEDI–CAPS UNIVERSITY, INDORE

</div>

<div id="Menu">

<a href="#"> Home </a>||<a href="#"> Admission </a>||<a href="#"> Result </a>||<a
href="#"> Contact Information </a>

</div>

</body>

</html>
```

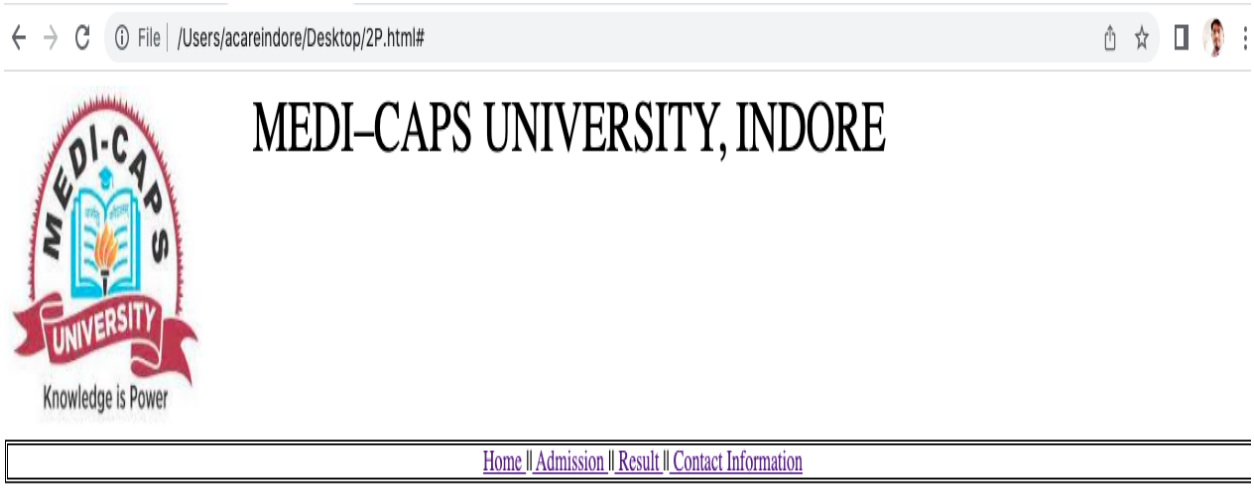
// CSS Style

```
body
{
/* background-image:url(bg1.png);*/
   background-color:lightblue;
}
div#Header
{
margin-left:50px;
margin-right:50px;
}
div#logo
{
width:10%;
height:100px;
background-color:red;
position:absolute;
}
div#title
{
border:2px
position:absolute;
top:15px;
```

```
margin-left:5%;
margin-right:10%;
width:80%;
height:80px;
font-size:40px;
text-align:center;
}
div#content
{
borderL:dashed;
position:absolute;
top:80px;
text-align:center;
}
div#menu
{
position:absolute;
top:200px;
border:double;
width:100%;
margin-top:4px;
text-align:center;
}
```

```
a:hover

{

backgroung-color:yellow;

}

a:hover

{

background-color:yellow;

}

/*

a:link {color:green;}

a:visited {color:green;}

a:hover {color:red;}

a:active {color:yellow;}

*/
```

4. OUTPUT:



Result: *Thus the program has been executed successfully.*

EXPERIMENT-2

TITLE: Write a program to Create User registration form and Login Page using simple HTML, Basic HTML application using TAG like hyperlink, lists, tables, images, simple HTML forms, website's structure.

Outcome: To learn the Create user registration form and login page using html.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Create User registration form and Login Page using html.		√										

1. DESCRIPTION ABOUT EXPERIMENT:

- Create and add HTML form elements and their associated elements. Here we will create nested form-related or associated elements inside opening and closing Form tags.
- You can also 'add action' attribute to your <form> tag

1.1. **HTML FORM**: An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc. An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

SYNTAX:

```
<form action = "URL" method = "GET or POST">
        // form elements like submit or reset buttons, input, text area etc.
</form>
```

2. PROCEDURE:

STEP 1: Choose an HTML editor

STEP 2: Create your HTML file

STEP 3: Add text fields and create your form

STEP 4: Add placeholders

STEP 5: Edit your HTML registration form

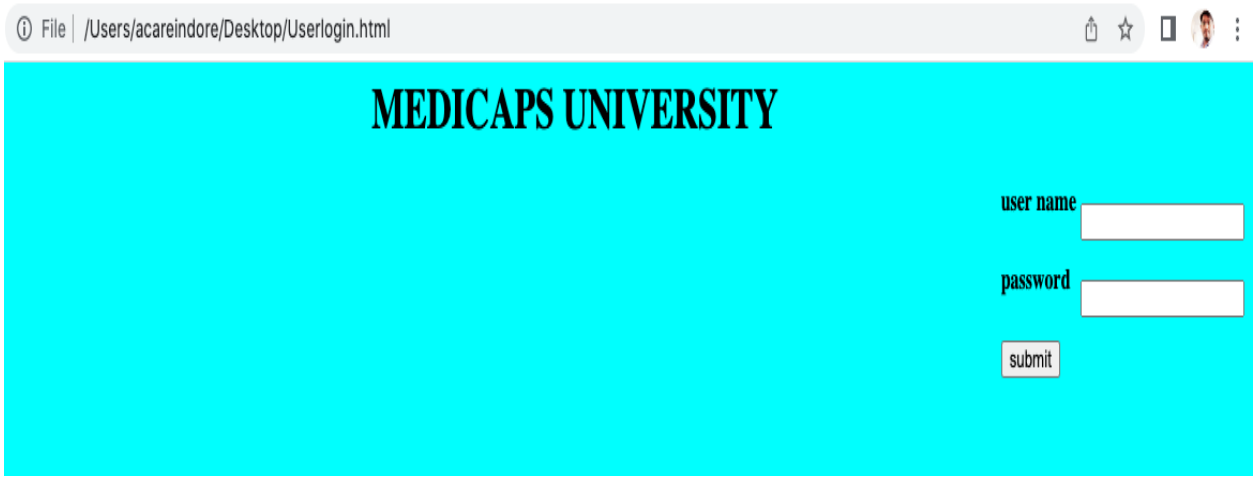
3. PROGRAM:

// Login.html

```
<!DOCTYPE html>

<html>

<head>

        <title> Login Page </title>

</head>

<body bgcolor="cyan"> <center>

<strong><h1> MEDICAPS UNIVERSITY </h1></strong></center>

<right>

<table align="right">

<tr>

        <td><h4>user name</td>

        <td><input type="text" ></td>

        <td></td>

</tr>

<tr>

        <td><h4>password</td>

        <td><input type="password"></td>

        <td></td>

</tr>

<tr>

<td>
```

```
<form method="post" action="catalog.html" >
<input type="submit" value="submit" >
</form>
</td>
</html>
```

4. OUTPUT:



Result: *Thus the program has been executed successfully.*

EXPERIMENT-3

TITLE: Write a program to create a Static Webpage Using Table Tags of Html

Outcome: To learn the development process to create a static webpage using table tags.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Create a static webpage using table tags of html.		√										

1. DESCRIPTION ABOUT EXPERIMENT:

1.1. TABLE IN HTML

- A table is a representation of data arranged in rows and columns. Really, it's more like a spreadsheet. In HTML, with the help of tables, you can arrange data like images, text, links and so on into rows and columns of cells.
- The use of tables in the web has become more popular recently because of the amazing HTML table tags that make it easier to create and design them.
- To create a table in HTML you will need to use tags. The most important one is the <table> tag which is the main container of the table. It shows where the table will begin and where it ends.

1.2. ATTRIBUTE OF TABLE:

- **border:** This attribute used for give border around table and border for all cell.
- **colspan:** This attribute defines the number of columns. It is used inside <td> tag.
- **rowspan:** This attribute defines the number of rows. It is used inside <td> tag**.**
- **id:** This attribute are used for provide unique id for table.

1.3.COMMON HTML TABLE TAGS:

Tag	Description
<tr>	represents rows
<td>	used to create data cells
<th>	used to add table headings
<caption>	used to insert captions
<thead>	adds a separate header to the table
<tbody>	shows the main body of the table
<tfoot>	creates a separate footer for the table

2. PROCEDURE

STEP 1: Define the thickness of the border

STEP 2: Merge columns Define the width of a cell

STEP 3: Add text fields and create your form

3. SYNTAX OF TABLE:

```
<table>
        <tr>
          <td>Cell 1</td>
          <td>Cell 2</td>
          <td>Cell 3</td>
        </tr>
    <tr>
          <td>Cell 4</td>
          <td>Cell 5</td>
          <td>Cell 6</td>
    </tr>
</table>
```

4. PROGRAM:

```
 <!DOCTYPE html>
<html>
<body>
<table border="3" cellpadding="2">
<caption align="top"> <b> Specification Table with Hours and Marks <b><caption>

<tr>
      <th rowspan="2"> Unit No. </th>
      <th rowspan="2"> Subject Title </th>
      <th rowspan="2"> Teaching Hours </th>
     <th colspan="4"> Distribution of Theory Marks</th>
</tr>

<tr>
<td> Internal </td>
<td> Assignment </td>
<td> Attendance </td>
<td> TOTAL MARKS</td>
</tr>
<tr>

<TR>
        <TD> <center>I </TD>
        <TD>Internet and Web Technology</TD>
        <TD><center>40</TD>
        <TD><center>40</TD>
        <TD><center>10</TD>
        <TD><center>10</TD>
        <TD><center>100</TD>
      </TR>
```

```
 <TR>
        <TD> <center>II </TD>
        <TD>Database Mangement</TD>
        <TD><center>40</TD>
        <TD><center>40</TD>
        <TD><center>10</TD>
        <TD><center>10</TD>
        <TD><center>100</TD>
 </TR>

<TR>
        <TD> <center>III </TD>
        <TD>Software Engineering</TD>
        <TD><center>40</TD>
        <TD><center>40</TD>
        <TD><center>10</TD>
        <TD><center>10</TD>
        <TD><center>100</TD>
 </TR>

 <TR>
            <TD> <center> IV </TD>
            <TD>Data Science</TD>
            <TD><center>40</TD>
            <TD><center>40</TD>
            <TD><center>10</TD>
            <TD><center>10</TD>
            <TD><center>100</TD>
</TR>
```

```
<TR>
        <TD> <center>V </TD>
        <TD>Cloud Computing</TD>
        <TD><center>40</TD>
        <TD><center>40</TD>
        <TD><center>10</TD>
        <TD><center>10</TD>
        <TD><center>100</TD>
</TR>

</html>
```

5. OUTPUT:

Specification Table with Hours and Marks

Unit No.	Subject Title	Teaching Hours	Distribution of Theory Marks			
			Internal	Assignment	Attendance	TOTAL MARKS
I	Internet and Web Technology	40	40	10	10	100
II	Database Mangement	40	40	10	10	100
III	Software Engineering	40	40	10	10	100
IV	Data Science	40	40	10	10	100
V	Cloud Computing	40	40	10	10	100

Result: *Thus the program has been executed successfully.*

EXPERIMENT-4

TITLE: Write a Program to create a HTML page, which has properly aligned paragraphs with image along with it.

Outcome: To learn the create a html page aligned paragraphs with image.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Create a HTML page, which has properly aligned paragraphs with image.		√	√									

1. DESCRIPTION ABOUT EXPERIMENT: -

1.1. HTML Form Validation

- The data entered into a form needs to be in the right format and certain fields need to be filled in order to effectively use the submitted form.
- Username, password, contact information are some details that are mandatory in forms and thus need to be provided by the user. Below is a code in HTML, CSS, and JavaScript to validate a form. HTML is used to create the form.
- JavaScript to validate the form. CSS to design the layout of the form.

Syntax of Form:

```
<form action = "URL" method = "GET or POST">

        // form elements like submit or reset buttons, input, text area etc.
</form>
```

2. PROCEDURE:

STEP 1: Start the Program

STEP 2: Create HTML tags

STEP 3: Create Forms

STEP 4: Create radio button and input box

STEP 5: Execute the form in browser

STEP 6: Enter the values

STEP 7: Stop

3. PROGRAM:

```
<!DOCTYPE html>

<html>

<head>

<title> Application Form Using HTML </title>

</head>

<form>

<fieldset>

<legend><h3>PERSONAL INFORMATION </h3></legend>

<img src="sawan.jpg" alt="Person" style="width:304px;height:228px" align="right">

First name:

<input type="text" name="firstname" placeholder="Enter First name"
required><br><br><br>

Last name:

<input type="text" name="lastname" placeholder=" Enter Last Name" required><br><br>

Gender:

<input type="radio" name="gender" value="male"> Male

<input type="radio" name="gender" value="female"> Female <br><br><br>

Birthday:

<input type="datetime-local" name="bdaytime"><br><br><br>

Country :<input list="Country" name="Country">

<datalist id="Country">

<option value="India">
```

<option value="Japan">

<option value="Australia">

<option value="United States">

<option value="United Kingdom">

</datalist>

E-mail:

<input type="email" name="email">

Select your favorite color:

<input type="color" name="favcolor" value="#ff0000">

Address:
<textarea rows='5' cols='40'></textarea>

</fieldset>

<fieldset>

<legend><h3> **EDUCATIONAL DETAILS** </h3></legend>

Degree :

<select name="degree">

<option value="B.E"> B.TECH </option>

<option value="M.E">M.TECH</option>

<option value="PhD"> PhD </option>

</select>

Certified Courses, any: <textarea rows='5' cols='40'></textarea>

</fieldset>

<fieldset>

<legend><h3> **WORK EXPERIENCE** </h3></legend>

Knowledge in :<input type="checkbox" name="HTML" value="HTML"> HTML

<input type="checkbox" name=".NET" value=".NET"> .NET

<input type="checkbox" name="JAVA" value="JAVA"> JAVA

<input type="checkbox" name="PHP" value="PHP"> PHP

Years of Experience :

<input type="number" name="years" min="0" max="50" step="1" value="0">

</fieldset>

<input type="submit" title="Enter the above details and click me" />

</form>

</body>

</html>

4. OUTPUT:

PERSONAL INFORMATION

First name: Vikash

Last name: Sawan

Gender: Male Female

Birthday: dd/mm/yyyy

Country : India

E-mail: vikash.sawan@medicaj

Select your favorite color:

Address:
Rau Indore

EDUCATIONAL DETAILS

Degree : M.TECH

Certified Courses, any:

WORK EXPERIENCE

Knowledge in : HTML .NET JAVA PHP

Years of Experience : 5

Submit

Result: *Thus the program has been executed successfully*

EXPERIMENT-5

TITLE: To write a Program to Create a form with various fields and appropriate front and validations using any one of the scripting languages.

Outcome: : To learn the develop to form with various fields.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Creating a form with various validation fields.		√	√	√								

1. DESCRIPTION ABOUT EXPERIMENT:

- Forms are used in webpages for the user to enter their required details that further send it to the server for processing.
- A form is also known as a web form or HTML form. Examples of form use are prevalent in e-commerce websites, online banking, online surveys to name a few.

1.1. J**avaScript**: It is used to create client-side dynamic pages. JavaScript is an object-based scripting language which is lightweight and cross-platform. JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

1.2. **Function:** An important part of JavaScript is the ability to create new functions within <script> and </script> tag. Declare a function in JavaScript using function keyword. The keyword function precedes the name of the function.

Syntax of Function:

```
function functionName (parameter or not)
    {
        //code to be executed;
    }
```

1.3.**Html Form**: An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc. An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

Syntax of Form:

```
<form action = "URL" method = "GET or POST">
        // form elements like submit or reset buttons, input, text area etc.
</form>
```

2. PROCEDURE:

STEP 1: Start the Program

STEP 2: Declare and Initialize the Variables

STEP 3: Create the function validate

STEP 4: Check the document values

STEP 5: Validate document form

STEP 6: Write coding for HTML

STEP 7: Execute in browser

STEP 8: Stop

3. PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
<title>Sample Registration Form with Javascript Validations</title>
<script language="javascript" type="text/javascript">
function validate() {
if (document.frm.user.value == "")
{
alert("Please enter the Name");
document.frm.user.focus();
return false;
}
if (document.frm.age.value == "")
{
alert("Please select the Age");
document.frm.age.focus();
return false;
}
if ((document.frm.s1[0].checked == false) && (document.frm.s1[1].checked == false))
{
alert("Please select the Gender");
return false;
```

```
}
if (document.frm.email.value == "")
{
alert("Please enter the email");
document.frm.email.focus();
return false;
}
if (document.frm.pass.value == "")
{
alert("Please enter the Password");
document.frm.pass.focus();
return false;
}
if ((document.frm.pass.length< 7) && (document.frm.pass.length> 15))
{
alert("Password Should be of length between 7 to 15");
document.frm.pass.focus();
return false;
}
if (document.frm.phone.value == "")
{
alert("Please enter the Phone Number");
document.frm.phone.focus();
return false;
```

```
}
if (document.frm.address.value == "")
{
alert("Please enter the Address");
document.frm.address.focus();
return false;
}
if (document.frm.city.value == "")
{
alert("Please enter the city");
document.frm.city.focus();
return false;
}
if (document.frm.state.value == "")
{
alert("Please enter the state");
document.frm.state.focus();
return false;
}
if (document.frm.country.value == "")
{
alert("Please select the Country");
document.frm.country.focus();
return false;
```

```
}
if (document.frm.agree.checked == "")
{
alert("Select terms and conditions check box");
return false;
}
else
{
return true;
}
}
function Numchk(num) {
if (isNaN(num.value))
{
alert('Entered Value:"' +num.value+ '" has invalid characters. Field accepts only numeric characters');
num.value = "";
}
}
function charchk(char) {
var letters = /^[A-Za-z]+$/;
if(!char.value.match(letters))
{
alert('Entered Value:"' +char.value+ '" has invalid characters. Field accepts only alphabet characters');
```

```
char.value = "";
}
}
</script>
</head>
<body bgcolor="pink">
<form name="frm" method="get" action="success.html">
<table width="590" align="center" cellpadding="1" cellspacing="1">
<tr>
<td colspan="3" bgcolor="#2267c6" height=40px style="text-align: center;">REGISTRATION
FORM</td>
</tr>
<tr>
<td colsspan="3" height=20px><font color=red>*</font> Indicates Require Field</td>
</tr><br><br>
<tr>
<td>Name<font color="red">*</font>
</td>
<tdalign="center">
</td>
<td>
<input type="text" name="user" title="Name" id="txtUser" onKeyUp="charchk(this)">
```

```
</td>
</tr>
<tr>
<td height="5px"></td>
</tr>
<tr>
<td>Age<font color="red">*</font>
</td>
<tdalign="center">
</td>
<td>
<select size="1" name="age" style="width: 50%;">
<option value="">---Please Select--</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
</select>
</td>
</tr>
```

```
<tr>
<td height="5px;"></td>
</tr>
<tr>
<td>Gender<font color="red">*</font>
</td>
<tdalign="center"></td>
<td>
<input type="radio" name="s1" value="Yes">Male
<input type="radio" name="s1" value="No">Female
</td>
</tr>
<tr>
<td height=5px></td>
</tr>
<tr>
<td height=5px></td>
</tr>
<tr>
<td>Email<font color="red">*</font>
</td>
<tdalign="center">:</td>
<td>
<input type="email" name="email" title="Email">
```

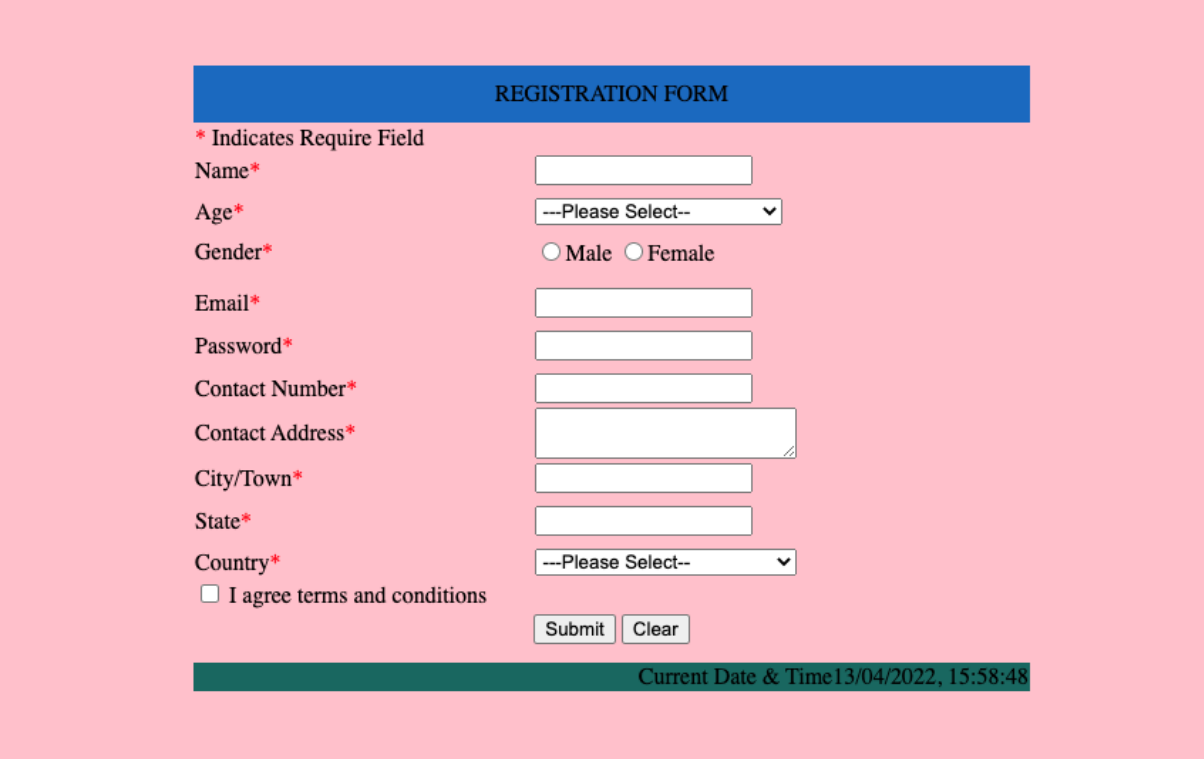
```
</td>
</tr>
<tr>
<td height=5px></td>
</tr>
<tr>
<td class="headsubtitle">Password<font color="red">*</font>
</td>
<tdalign="center"></td>
<td>
<input type="password" name="pass" title="password">
</td>
</tr>
<tr>
<td style="height:5px"></td>
</tr>
<tr>
<td>Contact Number<font color="red">*</font>
</td>
<tdalign="center"></td>
<td>
<input type="text" name="phone" title="Phone" onKeyUp="Numchk(this)">
</td>
</tr>
```

```
<tr>
<td>Contact Address<font color="red ">*</font>
</td>
<tdalign="center" valign="middle"></td>
<td>
<textarea rows="2" cols="35" name="address" style="width:
53%;"></textarea>
</td>
</tr>
<tr>
<td>City/Town<font color="red ">*</font>
</td>
<tdalign="center"></td>
<td>
<input type="text " name="city" title="city"onKeyUp="charchk(this)">
</td>
</tr>
<tr>
<td height=5px></td>
</tr>
<tr>
<td>State<font color="red">*</font>
</td>
<tdalign="center"></td>
<td>
```

```
<input type="text" name="state" title="state" onKeyUp="charchk(this)">
</td>
</tr>
<tr>
<td height=5px></td>
</tr>
<tr>
<td>Country<font color="red">*</font>
</td>
</td>
<tdalign="center "></td>
<td>
<select size="1" name="country" style="width: 53%;">
<option value="">---Please Select--</option>
<option value="Australia ">Australia</option>
<option value="india ">india</option>
<option value="italy ">italy</option>
<option value="ireland ">ireland</option>
</select>
</td>
</tr>
<tr>
<td>
<input type="checkbox" name="agree"> I agree terms and conditions
```

```
</td>

<tr>

</tr>

<tr>

<td align="center " colspan="3">

<input type="submit" name="add" value="Submit" onClick="return
validate();">

<input type="reset" name="reset" value="Clear">

</td>

</tr>

<tr>

<td height=10px></td>

</tr>

<tr>

<td id="footer" name= "footers" colspan=3 bgcolor=#22067c6
align=right></td>

</tr>

</table>

</form>

</body>

<script>

document.getElementById("footer").innerHTML = "Current Date & Time" +
new Date().toLocaleString();

</script>

</html>
```

4. OUTPUT:



Result: *Thus the program has been executed successfully.*

EXPERIMENT-6

TITLE: Develop and demonstrate the usage of inline, internal and external style sheet using CSS.

Outcome: To learn the design a web page using CSS.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Create a web page using CSS		√	√									

1. DESCRIPTION ABOUT EXPERIMENT:

1.1.Cascading Style Sheet (CSS):

- It is used to set the style in web pages that contain HTML elements. It sets the background color, font-size, font-family, color, etc. Property of elements on a web page.
- **There are three types of CSS which are given below:**
  - i. Inline CSS Internal or Embedded CSS
  - ii. External CSS
  - iii. Inline CSS

i. **Internal or Embedded CSS:** This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.

ii. **External CSS:** External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, etc.). CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages.

iii. **Inline CSS:** It contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.

1. **PROCEDURE:**

STEP 1: Use different font styles

STEP 2: Control the repetition of image with background-repeat and no-repeat property

STEP 3: Define style for links as a: link, a: active, a: hover, a: visited

STEP 4: Add customized cursors for links.

2. **PROGRAM:**

//sample.html

```
<!DOCTYPE html>
<html>
 <head>
 <style type="text/css">
 body
 {
 background-image:url('cse.jpg');
 background-repeat:no-repeat;
 background-position:center center;
 background-attachment:fixed;
 background-color:pink;
 }
 a:link { text-decoration:none;color:orange; }
 a:visited { text-decoration:none;color:red; }
 a:hover { text-decoration:underline;color:blue; }
 a:active { text-decoration:underline;color:purple; }
 h3 { color:green; }
 .c1{cursor:crosshair}
 .c2{cursor:pointer}
 .c3{cursor:move}
```

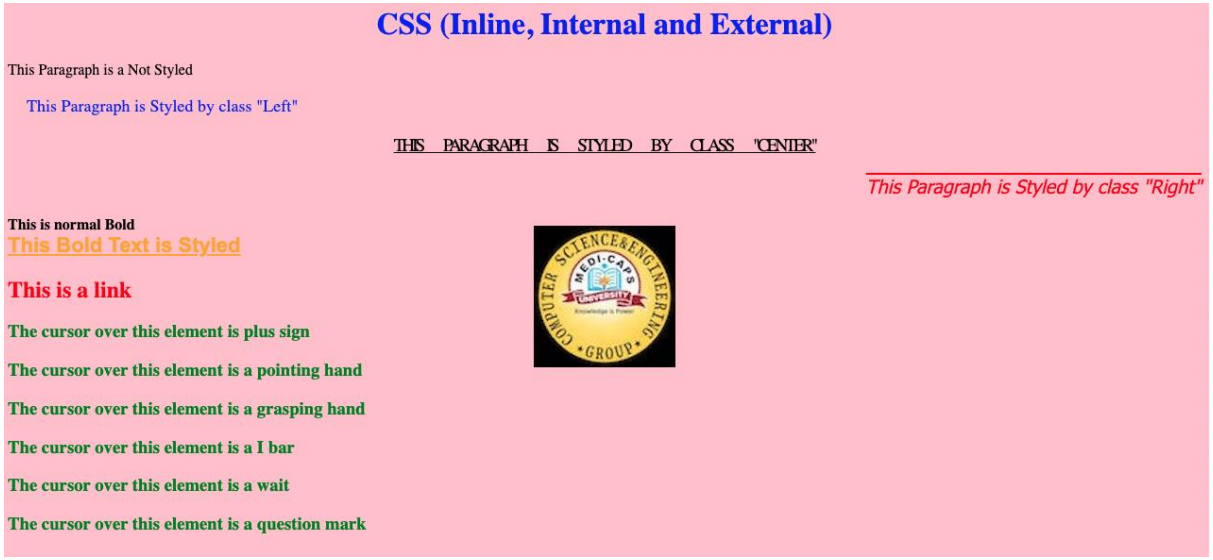
```
.c4{cursor:text}

.c5{cursor:wait}

.c6{cursor:help}

</style>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body bgcolor="cyan">

<h1 style="color:blue;text-align:center;"> CSS (Inline, Internal and External) </h1>

<p>This Paragraph is a Not Styled</p>

<p class="left">This Paragraph is Styled by class "Left"</p>

<p class="center">This Paragraph is Styled by class "Center"</p>

<p class="right">This Paragraph is Styled by class "Right"</p>

<b>This is normal Bold</b> <br>

<b id="headline">This Bold Text is Styled </b>


<h2><b><a href=" ">This is a link</a></b></h2>

<h3 class="c1">The cursor over this element is plus sign</h3>

<h3 class="c2">The cursor over this element is a pointing hand</h3>

<h3 class="c3">The cursor over this element is a grasping hand</h3>

<h3 class="c4">The cursor over this element is a I bar</h3>

<h3 class="c5">The cursor over this element is a wait</h3>

<h3 class="c6">The cursor over this element is a question mark</h3>

</html>
```

//style.css

```
p.left
{
text-align:left;
color:blue;
font-family:Cambria;
font-size:large;
text-indent:20px;
}
p.center
{
 text-align:center;
 text-decoration:underline;
 text-transform:uppercase;
 letter-spacing:-3px;
 word-spacing:20px;
 font-size:larger;
}
p.right
{
 text-align:right;
 color:red;
```

```
 font-family:Tahoma;
 font-size:15pt;
 text-decoration:overline;
 font-style:italic;
}
b#headline
{
 color:orange;
 font-size:22px;
 font-family:arial;
 text-decoration:underline;
}
```

4. OUTPUT-1: background-repeat:no-repeat;



4.1 OUTPUT-2: background-repeat:repeat;

Result: *Thus the program has been executed successfully.*

EXPERIMENT-7

TITLE: Write a program to design a simple calculator using JavaScript

Outcome: To learn the design a simple calculator using JavaScript.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Design calculator using JavaScript.		√	√	√								

1. DESCRIPTION ABOUT EXPERIMENT:

- **JavaScript Calculator:** As we know, the Calculator is a portable device used in our daily life to perform various mathematical functions such as addition, subtraction, multiplication, division, root, etc.
- However, we have scientific or sophisticated calculators used to solve complex tasks such as trigonometry functions, degrees, exponential operators, log functions, hyperbolic functions, square root, and so on.
- In this topic, we will create a calculator program in JavaScript.

1.1. J**avaScript**: It is used to create client-side dynamic pages. JavaScript is an object-based scripting language which is lightweight and cross-platform. JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

1.2. **Function:** An important part of JavaScript is the ability to create new functions within <script> and </script> tag. Declare a function in JavaScript using function keyword. The keyword function precedes the name of the function.

Syntax of Function:

```
function functionName (parameter or not)
        {
                //code to be executed;
        }
```

1.3.**Html Form**: An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc. An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

Syntax of Form:

```
<form action = "URL" method = "GET or POST">
        // form elements like submit or reset buttons, input, text area etc.
</form>
```

1. PROCEDURE:

STEP 1: You add two or more numbers (any number of digits) using the number keys given on the calculator.

STEP 2: You then perform fundamental arithmetic functions, i.e., addition, subtraction, multiplication and division on these numbers.

STEP 3: The calculator should be able to calculate and give out the correct solutions.

- This is the premise of your calculator tool. So, you need a keypad, a display unit and function elements.
- The “equals to” button will evaluate the answer while the “clear” button will ensure the removal of all the inputs on the calculator display.

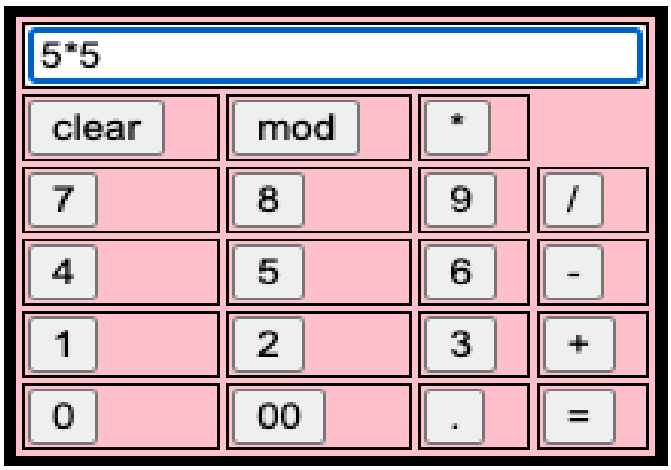
2. PROGRAM:

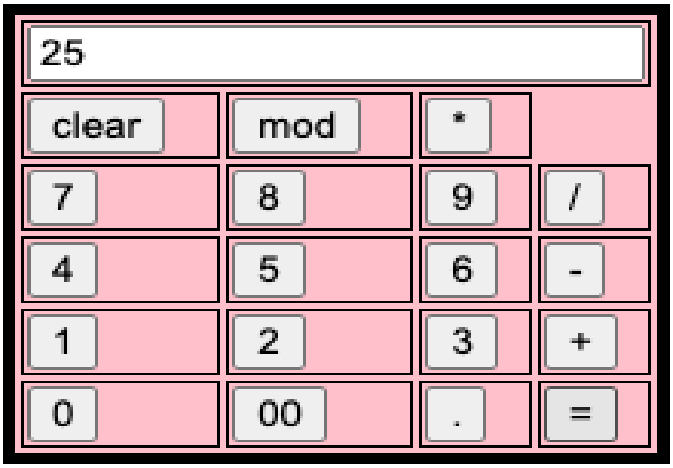
```
<!DOCTYPE html>
<html>
<head>
 <title>Calculator</title>
<script language="javascript">
var inputstring="";
function updatestring(value)
{
inputstring=inputstring+value;
document.calculator.input.value=inputstring;
}
</script>
</head>
 <body>
<form name="calculator">
<table border="4" align="center" bgcolor="pink" bordercolor="black">
<tr>
 <td colspan="4"><input type="text" name="input" maxlength="15" size="27"></td>
</tr>
<tr> <td><input type="button" value="clear" onclick="input.value=' ';inputstring=' '
"></td>
```

```
<td><input type="button" value="mod" onclick="updatestring('%')"></td>
<td><input type="button" value="*" onclick="updatestring('*')"> </td> </tr>
<tr>
<td><input type="button" value="7" onclick="updatestring('7')"> </td>
<td><input type="button" value="8" onclick="updatestring('8')"> </td>
 <td><input type="button" value="9" onclick="updatestring('9')"> </td>
 <td><input type="button" value="/" onclick="updatestring('/')"> </td> </tr>
<tr>
<td><input type="button" value="4" onclick="updatestring('4')"> </td>
<td><input type="button" value="5" onclick="updatestring('5')"> </td>
 <td><input type="button" value="6" onclick="updatestring('6')"> </td>
 <td><input type="button" value="-" onclick="updatestring('-')"> </td> </tr>
<tr>
<td><input type="button" value="1" onclick="updatestring('1')"> </td>
 <td><input type="button" value="2" onclick="updatestring('2')"> </td>
<td><input type="button" value="3" onclick="updatestring('3')"> </td>
 <td><input type="button" value="+" onclick="updatestring('+')"> </td> </tr>
<tr>
<td><input type="button" value="0" onclick="updatestring('0')"> </td>
<td><input type="button" value="00" onclick="updatestring('00')"> </td>
 <td><input type="button" value="." onclick="updatestring('.')"> </td>
<td><input type="button" value="=" onclick="input.value=eval(inputstring);"> </td>
</tr>
```

```
</table>

</form>

</body>

</html>
```

3. OUTPUT:





EXPERIMENT-8

TITLE: Write JavaScript to validate the following fields of the Registration page.

Outcome: To learn the create a validate fields of the registration page.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Design a validate fields of the registration page		√	√									

1. DESCRIPTION ABOUT EXPERIMENT:

- In order to validate the fields of login and registration pages JavaScript is used. JavaScript is programming code that can be inserted into HTML pages.
- JavaScript inserted into HTML pages, can be executed by all modern web browsers. JavaScript is mainly used for validating the elements in a form submitted by the user. This JavaScript code can react to user events

1.1. J**avaScript**: It is used to create client-side dynamic pages. JavaScript is an object-based scripting language which is lightweight and cross-platform. JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

1.2. **Function:** An important part of JavaScript is the ability to create new functions within <script> and </script> tag. Declare a function in JavaScript using function keyword. The keyword function precedes the name of the function.

Syntax of Function:

```
function functionName (parameter or not)
    {
        //code to be executed;
    }
```

1.3. **Html Form**: An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc. An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

Syntax of Form:

```
<form action = "URL" method = "GET or POST">
    // form elements like submit or reset buttons, input, text area etc.
</form>
```

2. PROCEDURE:

STEP 1: First Name (Name should contains alphabets and the length should not be less than 6 characters).

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

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

STEP 4: Mobile Number (Phone number should contain 10 digits only).

STEP 5: Last Name and Address (should not be Empty).

3. PROGRAM:

```
// After clicking OK button the page is redirected to success.html


<!DOCTYPE html>
<html>
 <head><title>Registration Form Validation</title></head>
 <body bgcolor="#E4F0F8">
 <script type='text/javascript'>
 function formValidator()
 {
 // Make quick references to our fields
 var firstname = document.getElementById('firstname');
 var lastname = document.getElementById('lastname');
 var email = document.getElementById('email');
 var pass = document.getElementById('pass');
 var addr = document.getElementById('addr');
 var mobileno = document.getElementById('mobileno');
 // Check each input in the order that it appears in the form!
 if(notEmpty(firstname, "can not be null")){
 if(isAlphabet(firstname, "Please enter only letters for your Firstname")){
 if(lengthRestriction(firstname, 5)){
 if(isAlphabet(lastname, "Please enter only letters for your Lastname")){
 if(emailValidator(email, "Please enter a valid email address")){
 if(lengthRestriction(pass, 5)){
 if(isAlphanumeric(pass, "please enter Numbers and Letters Only for password")){
 if(notEmpty(addr, "please enter the address")){
```

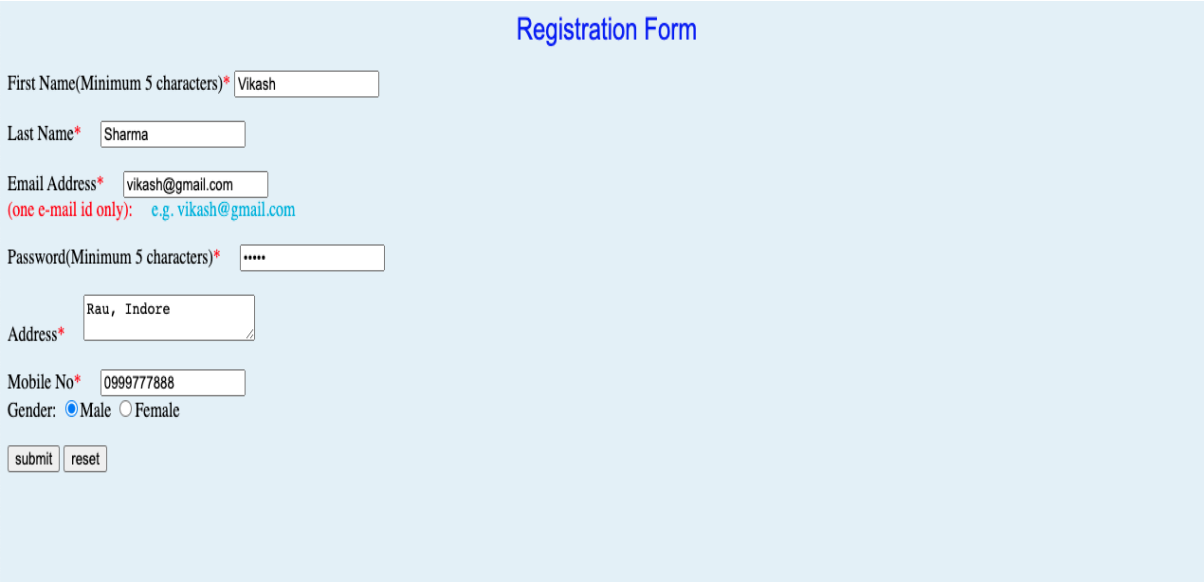
```
if(isNumeric(mobileno, "Please enter a valid mobileno")){
if(lengthRestriction1(mobileno, 10,10)){
return true;
} } } }
}
}
}
} } }
return false;
}
function notEmpty(elem, helperMsg){
if(elem.value.length == 0){
alert(helperMsg);
elem.focus(); // set the focus to this input
return false;
}
return true;
}
function isNumeric(elem, helperMsg){
var numericExpression = /^[0-9]+$/;
if(elem.value.match(numericExpression)){
return true;
}else{
alert(helperMsg);
elem.focus();
return false;
}
```

```
 }
function isAlphabet(elem, helperMsg){
 var alphaExp = /^[a-zA-Z]+$/;
 if(elem.value.match(alphaExp)){
 return true;
 }else{
 alert(helperMsg);
 elem.focus();
 return false;
 }
 }
function isAlphanumeric(elem, helperMsg){
 var alphaExp = /^[0-9a-zA-Z]+$/;
 if(elem.value.match(alphaExp)){
 return true;
 }else{
 alert(helperMsg);
 elem.focus();
 return false;
 }
 }
function lengthRestriction(elem, min){
 var uInput = elem.value;
 if(uInput.length >= min){
 return true;
 }else{
 alert("Please enter minimum " +min+ " characters");
```

```
elem.focus();
return false;
}
}
function emailValidator(elem, helperMsg)
{
var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/;
if(elem.value.match(emailExp))
{
return true;
}
else{
alert(helperMsg);
elem.focus();
return false;
}
}
function lengthRestriction1(elem, min, max) {
var uInput = elem.value;
if(uInput.length >= min && uInput.length <= max)
{
return true;
}
else {
alert("Please enter 10 numbers only");
elem.focus();
return false;
```

```
}

}

</script>

<center><font color="blue" size="5" face="arial">Registration
Form</font></center><br />

<form onsubmit='return formValidator()' action="right.html">

First Name(Minimum 5 characters)<font color="red">* </font>

<input type='text' id='firstname' /><br /><br />

Last Name<font color="red"><font color="red">* </font> </font>
   

<input type='text' id='lastname' /><br /><br />

Email Address<font color="red">* </font>    

<input type='text' id='email' /><br />

<font color="red">(one e-mail id only):</font>    

<font color="redblue">e.g. vikash@gmail.com</font><br /><br/>

Password(Minimum 5 characters)<font color="red">* </font>    

<input type='password' id='pass'><br /><br/>

Address<font color="red">* </font>    

<textarea rows="2" cols="20" id='addr' /></textarea> <br /> <br/>

Mobile No<font color="red">* </font>    

<input type='text' id='mobileno' /><br />

Gender: <input type='radio' name="gender">Male

<input type='radio' name="gender">Female<br/><br />

<input type='Submit' value='submit' />

<input type='Reset' value='reset' />

</form>

</body>

</html>
```

4. OUTPUT:



Result: *Thus the program has been executed successfully.*

EXPERIMENT-9

TITLE: Write JavaScript to create and read value from cookie on the web pages

Outcome: To learn the understand Cookies let you store user information in web pages.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 2 Creates a web page cookie you store user information in web pages.		√	√									

1. DESCRIPTION ABOUT EXPERIMENT:

- The web servers host the website. The client-server makes a request for data from the webserver and the webserver fetches the required pages and responds to the client by sending the requested pages.
- The web server communicates with the client-server using HTTP (Hypertext Transfer Protocol).
- HTTP is a stateless protocol which means the server needs not to retain the user information once the transaction ends and the connection is closed. The web browser is an example of a client-server which communicates with the web server using HTTP.
- HTTP prevents long engagement of the client with the webserver and the connection is closed automatically once the request is serviced. But often it is required to store the user information for future references.
- One of the most common uses of cookies is for authentication. Cookies serve the purpose of retaining user information even when the connection is lost. Cookies are data, stored in text files, on the computer.

- **Working Principle:** When the client or web browser establishes a connection with the webserver, the webserver sends some data to the browser in the form of a cookie. The browser may accept or reject the cookie. If the browser accepts it, the cookie gets stored in the hard disk of the client device.
- The CGI scripts on the server can read and write cookie values that are stored on the client, so when the client visits the same website again it retrieves the cookie data from the browser. JavaScript can be used to manipulate cookies using the cookie property of the Document object.
- JavaScript can read, create, modify, and delete the cookies for the current web page. The code below demonstrates how JavaScript can be used to create and read a value from the cookie.
- Create cookie using JavaScript: This function creates a cookie using the field-name, field-value, and expiry date. The path is left blank such that it applies to the current webpage. However, we can specify any other webpage or directory name.
- **Read cookie using JavaScript:** This function retrieves the cookie data stored in the browser. The cookie string is automatically encoded while sending it from the server to the browser. Hence it needs to be decoded before the actual data can be retrieved. Next, the decoded string is split into an array to get all the cookie name-value pairs. Loop through the array to find the field-name and respective field-values. If the cookie is found, the value is returned else the function returns the empty string.

2. PROCEDURE:

STEP 1: Expires: Specifies when the cookie will expire. If left empty the cookie expires immediately when the connection is lost.

STEP 2: Domain: Specifies the domain name of the website.

STEP 4: Name=Value: Cookies are stored in the form of name-value pairs.

STEP 5: Path: Specifies the webpage or directory that sets the cookie.

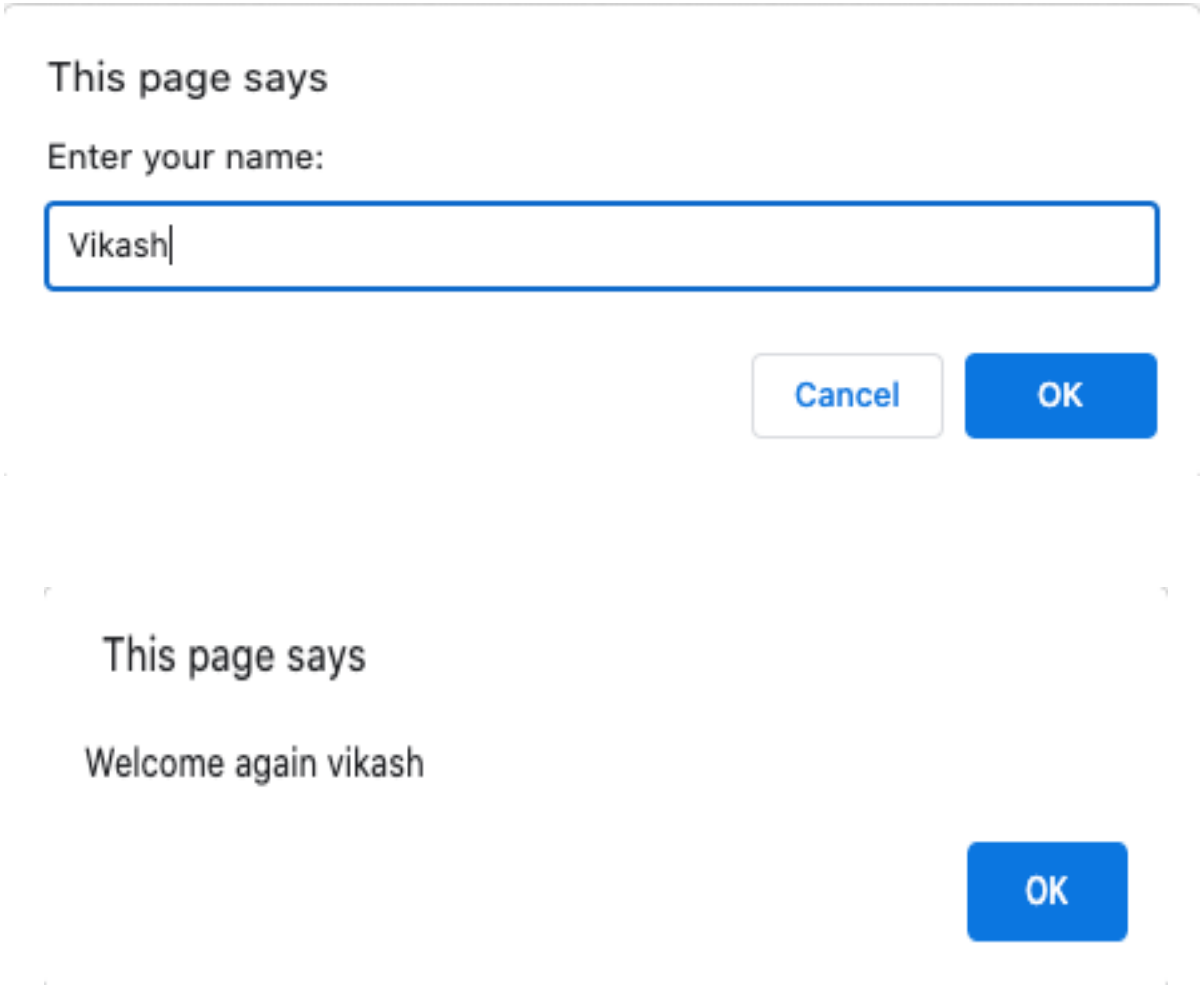
STEP 6: Secure: Specifies whether the cookie can be retrieved by any server (secure or non-secure).

3. PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
 <title>
        Create and read cookies
        using JavaScript
 </title>

 <script type="text/javascript">
        function createCookie(fieldname, fieldvalue, expiry) {
                var date = new Date();
                date.setTime(date.getTime()+ (expiry*24*60*60*1000));
                var expires = "expires=" + date.toGMTString();
                document.cookie = fieldname + "=" + fieldvalue
                                                + ";" + expires + ";path=/";
        }


        function readCookie(cname) {
                var name = cname + "=";
                var decoded_cookie =
                        decodeURIComponent(document.cookie);
                var carr = decoded_cookie.split(';');
                for(var i=0; i<carr.length;i++){
                        var c = carr[i];
                        while(c.charAt(0)==' '){
                                c=c.substring(1);
                        }
                        if(c.indexOf(name) == 0) {
                                return c.substring(name.length, c.length);
                        }
                }
```

```
            return "";
        }


        function runApp() {
            var user = readCookie("username");
            if(user != ""){
                alert("Hello "+user);
            }else{
                user=prompt("Enter your name: ", "");
                if(user!= "" && user!=null){
                    createCookie("username", user, 30);
                }
            }
        }

 </script>
</head>
<body onload="runApp()"></body>
</html>
```

4. OUTPUT:


This page says
Enter your name:
Vikash
Cancel
OK
This page says
Welcome again vikash
OK

EXPERIMENT-10

TITLE: Write an XML file which will display the Book information which includes the following:

a) Title of the book

b) Author Name

c) ISBN number

d) Publisher name

e) Edition

f) Price

Write a Document Type Definition (DTD) to validate the above XML file. Display the XML file as follows. The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XML schemas XSL and CSS for the above purpose.

Outcome: To learn the create a xml file which will display the book information.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 3 Create to XML file which will display the Book information.		√	√	√								

1. DESCRIPTION ABOUT EXPERIMENT:

- **DTD vs XML Schema:** The DTD provides a basic grammar for defining an XML Document in terms of the metadata that comprise the shape of the document. An XML Schema provides this, plus a detailed way to define what the data can and cannot contain. It provides far more control for the developer over what is legal, and it provides an Object-Oriented approach, with all the benefits this entails.
- Many systems interfaces are already defined as a DTD. They are mature definitions, rich and complex. The effort in re-writing the definition may not be worthwhile. DTD is also established, and examples of common objects defined in a DTD abound on the Internet -- freely available for re-use. A developer may be able to use these to define a DTD more quickly than they would be able to accomplish a complete re-development of the core elements as a new schema.
- Finally, you must also consider the fact that the XML Schema is an XML document. It has an XML Namespace to refer to, and an XML DTD to define it. This is all overhead. When a parser examines the document, it may have to link this all in, interpret the DTD for the Schema, load the namespace, and validate the schema, etc., all before it can parse the actual XML document in question. If you're using XML as a protocol between two systems that are in heavy use, and need a quick response, then this overhead may seriously degrade performance.

2. PROCEDURE:

STEP 1: Define the style rules for the text elements such as title, color, background font-size, etc.

STEP 2: Define each element such as books title, author. ISBN, publisher, edition, price etc. Using the display property of XML.

STEP 5: Identify the titles and bold them.

STEP 6: Linking XML with XSLT

Syntax of linking XML with XSLT

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

3. **PROGRAM:**

// books.dtd

```
<!ELEMENT details (title, author, ISBN_Number, publisher, edition, price) >

<!ELEMENT title (#PCDATA)>

<!ELEMENT author (#PCDATA)>

<!ELEMENT ISBN_Number (#PCDATA)>

<!ELEMENT publisher (#PCDATA)>

<!ELEMENT edition (#PCDATA)>

<!ELEMENT price (#PCDATA)>
```

// books.css

```
.thbook

{

background-color:gray;

}

.bg

{

background-color:red;

}
```

// books.xml

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

<!DOCTYPE book SYSTEM "books.dtd">

<book>

<details>

<title> Web Technology </title>

<author> Ralph Moseley </author>

<ISBN_Number>2550</ISBN_Number>

<publisher> Wiley </publisher>

<edition>5</edition>

<price>500/-</price>

</details>

<details>

<title> Java Programming</title>

<author> E Balagurusamy </author>

<ISBN_Number>2555</ISBN_Number>

<publisher>Pearson</publisher>

<edition>6</edition>

<price>600/-</price>

</details>

<details>

<title> DBMS</title>

<author> Navathe </author>
```

<ISBN_Number>**2560**</ISBN_Number>

<publisher> **Ramez Elmasri** </publisher>

<edition>**5**</edition>

<price>**700/-<**/price>

</details>

<details>

<title> **Software Engineering**</title>

<author> **Sommerville** </author>

<ISBN_Number>**2570**</ISBN_Number>

<publisher> **McGraw-Hill** </publisher>

<edition>**4**</edition>

<price>**800/-<**/price>

</details>

<details>

<title> **Automata**</title>

<author> **Gabe Ibanez** </author>

<ISBN_Number>**2575**</ISBN_Number>

<publisher>**Willay**</publisher>

<edition>**3**</edition>

<price>**900/-**</price>

</details>

<details>

<title> **Operating System**</title>

<author> **Abraham Silberschatz** </author>

<ISBN_Number>**2580**</ISBN_Number>

<publisher>**Willay**</publisher>

<edition>**10**</edition>

<price>**1000**/-</price>

</details>

</book>

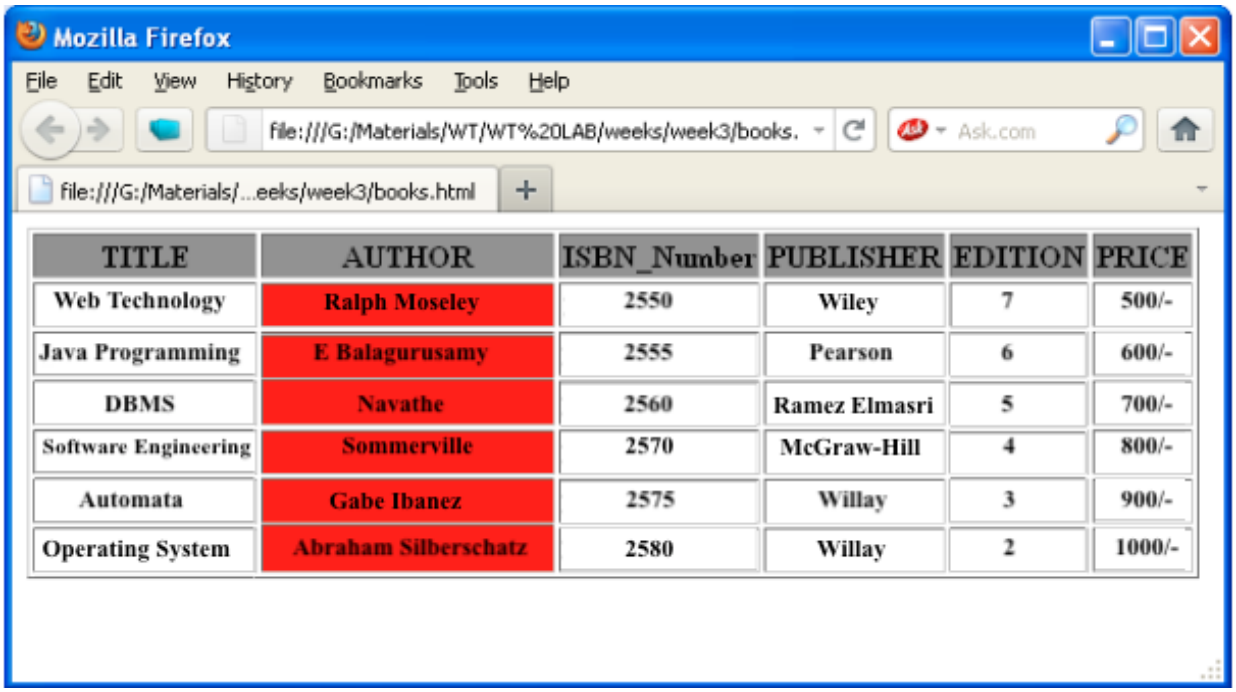
//books.html

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="books.css">
</head>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
{
// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else
{
```

```
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
document.write("<table border='1'>");
var x=xmlDoc.getElementsByTagName("details")
document.write("<tr><th class='thb'>");
document.write("TITLE</th><th class='thb'>AUTHOR</th><th
class='thb'>ISBN_Number</th><th class='thb'>PUBLISHER</th><th
class='thb'>EDITION</th><th class='thb'>PRICE</th></tr>");
for (i=0;i<x.length;i++)
{
document.write("<tr><td>");
document.write(x[i].getElementsByTagName("title")[0].childNodes[0].nodeValue);
document.write("</td><th class='bg'>");
document.write(x[i].getElementsByTagName("author")[0].childNodes[0].nodeValue.toU
pperCase());
document.write("</th><td>");
document.write(x[i].getElementsByTagName("ISBN_Number")[0].childNodes[0].nodeV
alue);
document.write("</td><td>");
document.write(x[i].getElementsByTagName("publisher")[0].childNodes[0].nodeValue);
```

```
document.write("</td><td>");

document.write(x[i].getElementsByTagName("edition")[0].childNodes[0].nodeValue);

document.write("</td><td>");

document.write(x[i].getElementsByTagName("price")[0].childNodes[0].nodeValue);

document.write("</td></tr>");

}

document.write("</table>");

</script>

</body>

</html>
```

4. OUTPUT:



TITLE	AUTHOR	ISBN_Number	PUBLISHER	EDITION	PRICE
Web Technology	Ralph Moseley	2550	Wiley	7	500/-
Java Programming	E Balagurusamy	2555	Pearson	6	600/-
DBMS	Navathe	2560	Ramez Elmasri	5	700/-
Software Engineering	Sommerville	2570	McGraw-Hill	4	800/-
Automata	Gabe Ibanez	2575	Willay	3	900/-
Operating System	Abraham Silberschatz	2580	Willay	2	1000/-

EXPERIMENT-11

TITLE: Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

Outcome: To learn the Count number of visitors on web page.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 4 Count number of visitors on web page.		√	√	√								

1. DESCRIPTION ABOUT EXPERIMENT:

- A PHP session is used to store data on a server rather than the computer of the user. Session identifiers or SID is unique numbers which are used to identify every user in a session-based environment.
- The SID is used to link the user with his information on the server like posts, emails etc.
- A session mechanism can be used to store page views which increment on each refresh and show the count on a webpage.
- A session is user specific and for every user, a separate session is created along with a separate session variable which is associated with that session.
- Using this mechanism, for every user the session variable is set to 1 initially for the first visit. On consecutive visits, the value of this session variable is incremented and displayed on the output webpage.

Syntax of PHP:

```
<? php
        // Code to be executed;
?>
```

2. **PROCEDURE:**

STEP1: Session_start (): It is a first step which is used to start the session. It is a standard call. The session_start () should be used whenever the session variable is used.

STEP 2: $_SESSION[‘views’]: This is the session variable which is used to store views count for a user’s session. ‘views’ is the session name. The session name should be always be enclosed within the single quote.

STEP 3: isset (): It is a standard php function which returns true or false depending upon whether the passed parameter is set or not.

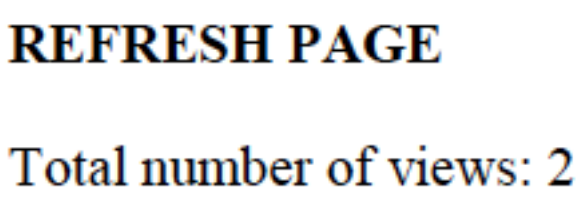
STEP 4: Create a file with .php extension in wamp (inside www directory) & XAMP (inside htdocs directory)

STEP5: Run the file in PHP environment like localhost

3. PROGRAM:

```
<?php

        echo "<h3> REFRESH PAGE </h3>";

        $fname="counter.txt";

        $fp = fopen($fname,"r");

        $hits= fscanf($fp,"%d");

        fclose($fp);

        $hits[0]++;

        $fp = fopen($fname,"w");

        fprintf($fp,"%d",$hits[0]);

        fclose($fp);

        echo "Total number of views: ".$hits[0];

?>
```

4. OUTPUT



Result: *Thus the program has been executed successfully.*

EXPERIMENT-12

TITLE: Write a PHP program to display a digital clock which displays the current time of the server.

Outcome: To learn the develop php program to display a digital clock.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 4 Create a php program to display a digital clock		√	√	√								

1. DESCRIPTION ABOUT EXPERIMENT:

- PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.)

Syntax of PHP:

```
<? php
        // Code to be executed;
?>
```

1.1. **PHP date () Function**: The PHP date () function converts timestamp to a more readable date and time format.

Syntax:

date (format, timestamp)

2. PROCEDURE:

STEP 1: Create the basic structure of the digital clock.

STEP 2: add the time, am/pm, and date options.

STEP 3: The third step is to add a php code to the time setting.

STEP 4: activate the date option in the digital clock.

3. PROGRAM:

```
<!DOCTYPE HTML>

<html>

<head>

        <meta http-equiv="refresh" content="1"/>

<style>

p {

        color:yellow;

        font-size:90px;

        position: absolute;

        top: 50%;

        left: 50%;

        transform: translate(-50%, -50%);

}

Body {

background-color:maroon;

}

</style>

        <p> <?php echo date(" h: i : s A");?> </p>

</head>

</html>
```

4. OUTPUT

Result: *Thus the program has been executed successfully.*

EXPERIMENT-13

TITLE: Write a Servlet application to print the current date and time.

Outcome: To learn the display current date and time using servlet.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 5 Develop the servlet application to display the current date and time.		√	√		√							

1. DESCRIPTION ABOUT EXPERIMENT:

- One of the most important advantages of using Servlet that can be used in most of the core Java methods available. This chapter will explain the java.util package provided by the Java Date class that encapsulates the current date and time.
- Date class supports two constructors. The first constructor to initialize the object the current date and time

Syntax:

Date ()

- The following constructor accepts a parameter that is equal to the number of milliseconds elapsed since January 1, 1970 at midnight.

Syntax:

Date (long millisec)

2. **PROCEDURE:**

STEP 1: Create the basic structure of the digital clock.

STEP 2: add the time, am/pm, and date options.

STEP 3: The third step is to add a servlet code to the time setting.

STEP 4: activate the date option in the digital clock.

3. **PROGRAM:**

// datesrv.java

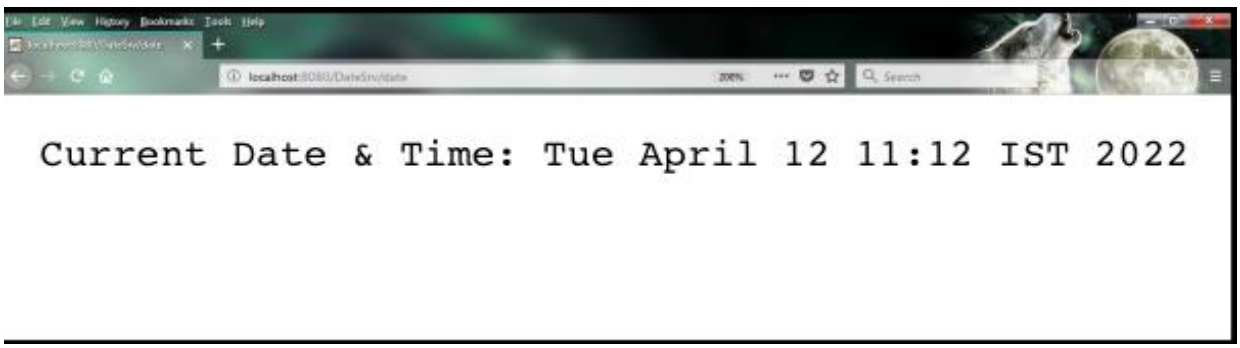
```
import java.io.*;

import javax.servlet.*;

public class DateSrv extends GenericServlet

{

// implement service()

public void service(ServletRequest req, ServletResponse res) throws IOException,

ServletException

        {

        //set response content type

        res.setContentType("text/html");

        //get stream obj

        PrintWriter pw = res.getWriter();

        //write req processing logic

        java.util.Date date = new java.util.Date();

        pw.println("<h2>"+"Current Date & Time: " +date.toString()+"</h2>");

        //close stream object
```

```
pw.close();
}
}
```

//web.xml

```
<?xml version="1.0"?>
<web-app>
<servlet>
<servlet-name>Date</servlet-name>
<servlet-class>DateSrv</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Date</servlet-name>
<url-pattern>/date</url-pattern>
</servlet-mapping>
</web-app>
```

4. OUTPUT



Result: *Thus the program has been executed successfully.*

EXPERIMENT-14

TITLE: Write a JSP application to print the current date and time.

Outcome: To learn the display current date and time using JSP.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 5 Develop the JSP application to display the current date and time.		√	√		√							

1. **DESCRIPTION ABOUT EXPERIMENT:**
   - JSP is a most powerful programming language, by which we can do many things and Java is an industry preferable language. So, it has a huge field of features. Here we discuss one of the best features of Java, that is how to represent the current date and time using Java.
   - There are many ways to do this, there are many classes by which it can be possible to display the current Date and Time.
   - **Using Date class:** There is a class called Date class which can represent the current date and time in GMT form. We can get the IST form by adding 5 hours and 30 minutes to the GMT form. This class comes under the util package of Java.

2. **PROCEDURE:**

   STEP 1: Create the basic structure of the digital clock.

   STEP 2: add the time, am/pm, and date options.

   STEP 3: The third step is to add a servlet code to the time setting.

   STEP 4: activate the date option in the digital clock.

3. PROGRAM:

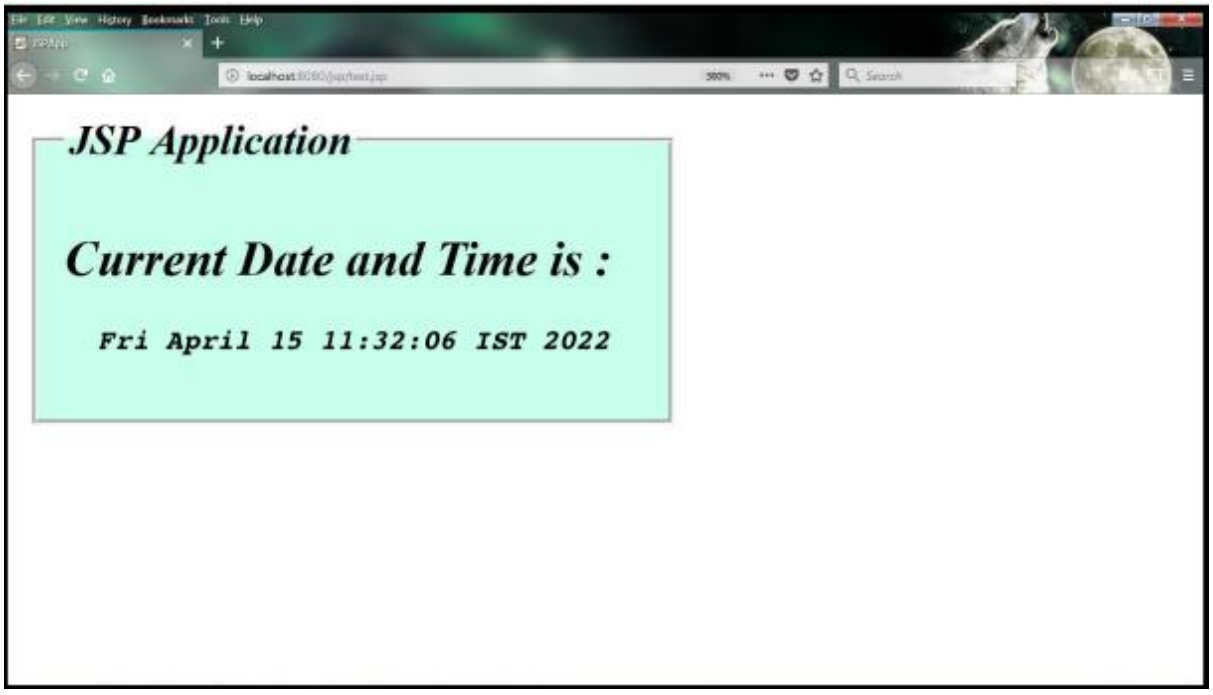
// test.jsp

```
<html>
        <head><title>JSP Application</title></head>
    <body>
    <form>
        <fieldset style="width:50%; background-color: #ccffeb;">
        <legend><b><i>JSP Application<i><b></legend>
        <h3>Current Date and Time is :</h3>
        <% java.util.Date d = new java.util.Date();
        out.println(d.toString()); %>
        </fieldset>
    </form>
    </body>
</html>
```

// web.xml

```
<web-app>
<servlet>
<servlet-name>JSP</servlet-name>
<jsp-file>/test.jsp</jsp-file>
</servlet>
<servlet-mapping>
<servlet-name>xyz</servlet-name>
<url-pattern>/test</url-pattern>
</servlet-mapping>
</web-app>
```

4. OUTPUT



Result: *Thus the program has been executed successfully.*