

List of Experiments

1.	Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient
2.	Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
3.	Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT-SHRINKING" in BLUE color. Then the font size decreases to 5pt.
4.	Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems: a. Parameter: A string Output: The position in the string of the left-most vowel b. Parameter: A number Output: The number with its digits in the reverse order
5.	Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
6.	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.
7.	Write a PHP program to display a digital clock which displays the current time of the server.
8.	Write the PHP programs to do the following: a. Implement simple calculator operations. b. Find the transpose of a matrix. c. Multiplication of two matrices. d. Addition of two matrices.
9.	Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following: a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList. b. Search for a word in states that begins with k and ends in s. Perform a caseinsensitive comparison. [Note: Passing re.Ias a second parameter to method compile performs a case-insensitive comparison.] Store this word in element1 of statesList. c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.

	d. Search for a word in states that ends in a. Store this word in element 3 of the list.
10.	Write a PHP program to sort the student records which are stored in the database using selection sort.

Study Experiment / Project:

Develop a web application project using the languages and concepts learnt in the theory and exercises listed in part A with a good look and feel effects. You can use any web technologies and frameworks and databases.

Note:

1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report (15-20 pages) that must include the following
 - a. Introduction
 - b. Requirement Analysis
 - c. Software Requirement Specification
 - d. Analysis and Design
 - e. Implementation
 - f. Testing

Conduction of Practical Examination:

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 30 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:

a) Part A: Procedure + Conduction + Viva: $10 + 35 + 5 = 50$ Marks

b) Part B: Demonstration + Report + Viva voce = $15 + 10 + 05 = 30$ Marks

Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

Content Sheet

Experiment No.	Experiment	Page No.
1.	Java script Simple Calculator	5
2.	Table of squares and cubes	8
3.	Text growing and shrinking in Java script	10
4.	HTML5 with embedded Java script using functions	14
5.	XML design with CSS	17
6.	Tracking number of visitors using PHP	20
7.	Digital clock using PHP	22
8.	Illustrating numerical operations using PHP	26
9.	Searching using Regular expressions using PHP	38
10.	Database access using PHP and PDO and sorting using Selection sort	42
	About XAMP	48
	Lab Working Environment	49
	Viva Questions and Answers	51

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient

Explanation:

This program demonstrates the use of embedding Javascript inside a webpage designed using HTML. The usefulness of Java script in client-side computations for small tasks, displaying results in its own graphical window, rendering the output to the web pages dynamically and requesting for input from the user are illustrated through this program.

Program :

HTML File (1.html)

```
<html>
<script type=text/javascript src=1.js>
</script>
<body>
<table border>
<tr >
<td colspan=4><input type=text id=res size=16 onfocus="this.blur()"></td>
</tr>
<tr>
<td><input type=button id=b1 value=0 size=2 onclick=f('0')></td>
<td><input type=button id=b2 value=1 size=2 onclick=f('1')></td>
<td><input type=button id=b3 value=2 size=2 onclick=f('2')></td>
<td><input type=button id=b4 value=+ size=2 onclick=f('+')></td>
</tr>
<tr>
<td><input type=button id=b5 value=3 size=2 onclick=f('3')></td>
<td><input type=button id=b6 value=4 size=2 onclick=f('4')></td>
<td><input type=button id=b7 value=5 size=2 onclick=f('5')></td>
<td><input type=button id=b8 value=- size=2 onclick=f('-')></td>
</tr>
<tr>
<td><input type=button id=b9 value=6 size=2 onclick=f('6')></td>
<td><input type=button id=b10 value=7 size=2 onclick=f('7')></td>
<td><input type=button id=b11 value=8 size=2 onclick=f('8')></td>
<td><input type=button id=b12 value=* size=2 onclick=f('*')></td>
</tr>
<tr>
<td><input type=button id=b13 value=9 size=2 onclick=f('9')></td>
<td><input type=button id=b14 value=+/- size=2 onclick=f('--')></td>
<td><input type=button id=b15 value='=' size=2 onclick=f('=')></td>
<td><input type=button id=b16 value='/' size=2 onclick=f('/')></td>
</tr>
<tr>
<td colspan=4 ><input type=button value=Clear size=16 onclick=f('c')></td>
</tr>
```

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

Java script Code: (1.js)

```
function f(d)
{
    if(d=='c')
    {
        document.getElementById('res').value="";
        return;
    }
    res=document.getElementById('res').value;
    if(res==0 && d==0)
        return;
    if(d=='+' || d=='-' || d=='*' || d=='/')
    {
        opr=d;
        num=parseFloat(res);
        document.getElementById('res').value=d;
        return;
    }
    if(d=='=')
    {
        num1=parseFloat(res);
        switch(opr)
        {
            case '+': ans=num+num1; break;
            case '-': ans=num-num1; break;
            case '*': ans=num*num1; break;
            case '/': ans=parseInt(num/num1); break;
        }
        document.getElementById('res').value=ans;
        return;
    }
    if(d=='--')
    {
        document.getElementById('res').value*=-1;
        return;
    }
    if(!isNaN(document.getElementById('res').value))
        document.getElementById('res').value+=d;
    else
        document.getElementById('res').value=d;
}
```

Javascript functions used:

document.getElementById() - The getElementById() method returns the element that has the ID attribute with the specified value. This method is one of the most common methods in the HTML DOM, and is used almost every time you want to manipulate, or get info from, an element on your document.

Returns *null* if no elements with the specified ID exists. An ID should be unique within a page. However, if more than one element with the specified ID exists, the getElementById() method returns the first element in the source code.

parseFloat()- parseFloat() function parses a string and returns a floating point number. This function determines if the first character in the specified string is a number. If it is, it parses the string until it reaches the end of the number, and returns the number as a number, not as a string.

parseInt() – parseInt() function parses a string and returns an integer. The radix parameter is used to specify which numeral system to be used, for example, a radix of 16 (hexadecimal) indicates that the number in the string should be parsed from a hexadecimal number to a decimal number.

If the radix parameter is omitted, JavaScript assumes the following:

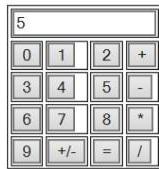
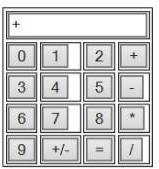
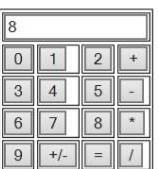
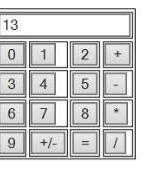
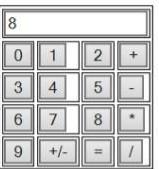
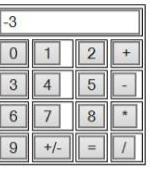
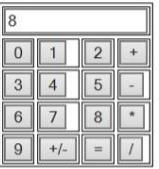
If the string begins with "0x", the radix is 16 (hexadecimal)

If the string begins with "0", the radix is 8 (octal). This feature is deprecated

If the string begins with any other value, the radix is 10 (decimal)

isNaN()- The isNaN() function determines whether a value is an illegal number (Not-a-number). This function returns true if the value equates to NaN. Otherwise it returns false.

Output:**Initial view:**

Add				
Subtract				
Multiply				
Divide				

2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.

Explanation:

This program aims at demonstrating the usefulness of Java script in generating dynamic web pages using Java script. Accepting user input as web page loads is also demonstrate in this program. This script also illustrates validation using Java script.

Program

HTML File: (2.html)

```
<html>
<script type="text/javascript src=2.js>
</script>
<body onload=sc()>
</body>
</html>
```

Java script Code: (2.js)

```
function sc()
{
    rng=prompt('Enter the range');
    res=rng.split("-");
    if(res.length!=2)
    {
        alert("invalid range ");
        return;
    }
    first=parseInt(res[0]);
    second=parseInt(res[1]);
    if(first>second)
    {
        alert("invalid range ");
        return;
    }
    str="




```

Functions used:

prompt() - The prompt() method displays a dialog box that prompts the visitor for input. A prompt box is often used if you want the user to input a value before entering a page. **Note:** When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. Do not overuse this method, as it prevents the user from accessing other parts of the page until the box is closed.

The prompt() method returns the input value if the user clicks "OK". If the user clicks "cancel" the method returns null

split() – The split() method is used to split a string into an array of substrings, and returns the new array. If an empty string ("") is used as the separator, the string is split between each character. The split() method does not change the original string.

alert() – The alert() method displays an alert box with a specified message and an OK button. An alert box is often used if you want to make sure information comes through to the user. The alert box takes the focus away from the current window, and forces the browser to read the message. Do not overuse this method, as it prevents the user from accessing other parts of the page until the box is closed.

document.write() - The write() method writes HTML expressions or JavaScript code to a document. The write() method is mostly used for testing: If it is used after an HTML document is fully loaded, it will delete all existing HTML.

Output:

Between a range of values 0-10

localhost needs some info from you.

Enter the range

0-10

OK

Cancel

Output:

localhost/pwlab/ckr/2.html

Number	Square	Cube
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

If range is same:

localhost needs some info from you.

Enter the range

5-5

OK

Cancel

Output:

localhost/pwlab/ckr/2.html

Number	Square	Cube
5	25	125

Invalid range:

localhost needs some info from you.

Enter the range

7-5

OK

Cancel

Output:

This site says...

invalid range

Don't let this page create more messages

OK

3. Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.

Explanation:

This program brings out the event handling capabilities of Java script. Timer event handling using Java script is explained in this program. Further, the use of style attributes in HTML and Java script is demonstrated. This program also brings out usefulness of Java script in handling dynamic positioning of elements in HTML page. These give the life to a web page and make it engaging.

Program:**HTML (3.html)**

```
<html>
<script type="text/javascript src=3.js>
</script>
<body onload=fnts()>
<p id="p1" style="font-size:12pt;color=black"> TEXT-GROWING </p>
</body>
</html>
```

Java script code (3.js)

```
var grow=true;
function fnts()
{
    fntsize=document.getElementById("p1").style.fontSize;
    document.getElementById("p1").style.color="red";

    ifntsize=parseInt(fntsize);
    window.setTimeout(fntGS,100,ifntsize);

}

function fntGS(ifs)
{
    if(grow)
    {
        ifs=ifs+1;
        if(ifs<=50)
        {
            document.getElementById("p1").style.fontSize=ifs+"pt";
        }
        else
        {
            grow=false;
            document.getElementById("p1").style.color="blue";
            document.getElementById("p1").innerHTML="TEXT-SHRINKING";
        }
    }
}
```

```

else
{
    ifs=ifs-1;
    if(ifs<5)
        return;
    document.getElementById("p1").style.fontSize=ifs+"pt";

}
window.setTimeout(fntGS,100,ifs);
}

```

Functions used:

window.setTimeout() - The setTimeout() method calls a function or evaluates an expression after a specified number of milliseconds. **Tip:** 1000 ms = 1 second. The function is only executed once.

Output

Growing text	Shrinking Text
	

Enhancement:

The position of text can also be moved along with its font size

Enhanced HTML File (3e.html)

```

<html>
<script type=text/javascript src=3e.js>
</script>
<body onload=fnts()>
<p id="p1" style="font-size:12pt;color:black;position:absolute;top:10px;left:10px"> TEXT-
GROWING </p>
</body>
</html>

```

Enhanced Java script code: (3e.js)

```

var grow=true;
function fnts()
{
    fntsize=document.getElementById("p1").style.fontSize;
    document.getElementById("p1").style.color="red";
    x=document.getElementById("p1").style.left;
    x=parseInt(x);
    y=document.getElementById("p1").style.top;
    y=parseInt(y);

    ifntsize=parseInt(fntsize);
    window.setTimeout(fntGS,100,ifntsize,x,y);
}

```

```

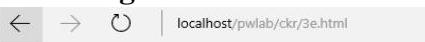
}

function fntGS(ifs)
{
    if(grow)
    {
        ifs=ifs+1;
        x=x+1;
        y=y+1;
        if(ifs<=50)
        {
            document.getElementById("p1").style.fontSize=ifs+"pt";
            document.getElementById("p1").style.top=y+"px";
            document.getElementById("p1").style.left=y+"px";
        }
        else
        {
            grow=false;
            document.getElementById("p1").style.color="blue";
            document.getElementById("p1").innerHTML="TEXT-SHRINKING";
        }
    }
    else
    {
        ifs=ifs-1;
        x=x+1;
        y=y+1;

        if(ifs<5)
            return;
        document.getElementById("p1").style.fontSize=ifs+"pt";
        document.getElementById("p1").style.top=y+"px";
        document.getElementById("p1").style.left=y+"px";
    }
    window.setTimeout(fntGS,100,ifs,x,y);
}

```

Output

Growing Text with movement	Shrinking text with movement
	

4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:

a. Parameter: A string

Output: The position in the string of the left-most vowel

b. Parameter: A number

Output: The number with its digits in the reverse order

Explanation:

This program demonstrates the structure of HTML5 files and semantic structure followed. Writing of functions in Java script and form based event handling in Java script is also illustrated. The dynamic processing of Java script without reloading HTML5 pages is also shown.

HTML5 file: (4.html)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Demonstrating Java script
functions</title> </head>
<script type=text/javascript src=4.js>
</script>
<body>
<section>
<h1>Finding left most vowel</h1>
<p>Enter a string: <input type=text id=t1></p>
<input type=button value=Find onclick = alert( findLMV(
document.getElementById('t1').value ))>

</section>
<hr>
<section>
<h1>Reverse of a number</h1>
<p>Enter a number: <input type=text id=t2></p>
<input type=button value=Reverse
onclick=alert(reverse(document.getElementById('t2').value))>

</section>
```

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

Java script (4.js)

```
function findLMV(str)
{
    for(i=0;i<str.length;i++)
    {
        if(str.charAt(i)=='a' || str.charAt(i)=='e' || str.charAt(i)=='i' || str.charAt(i)=='o' ||
        str.charAt(i)=='u')
            return ("Left most vowel of " + str + " is at location " + (i+1) );
    }
    return ("No vowels found for string "+ str);
}

function reverse(num)
{
    rnum=0;
    temp=num;
    if(isNaN(num))
    {
        return "invalid number";
    }
    while(num!=0)
    {
        rnum *= 10;
        rnum+= num % 10;
        num -= num % 10;
        num = Math.floor(num/10);
    }
    return "Reverse of num "+ temp + " is " + rnum;
}
```

Functions used:

charAt() - The charAt() method returns the character at the specified index in a string. The index of the first character is 0, the second character is 1, and so on.

Math.floor() - The floor() method rounds a number DOWNWARDS to the nearest integer, and returns the result. If the passed argument is an integer, the value will not be rounded.

Output:

Input	Output
-------	--------

<p>← → ⌂ localhost/pwlab/ckr/4.html</p> <p>Finding left most vowel</p> <p>Enter a string: <input type="text" value="semantics"/> x</p> <p><input type="button" value="Find"/></p> <hr/> <p>Reverse of a number</p> <p>Enter a number: <input type="text"/></p> <p><input type="button" value="Reverse"/></p>	<p>This site says...</p> <p>Left most vowel of semantics is at location 2</p> <p><input type="button" value="OK"/></p>
<p>← → ⌂ localhost/pwlab/ckr/4.html</p> <p>Finding left most vowel</p> <p>Enter a string: <input type="text" value="sys"/> x</p> <p><input type="button" value="Find"/></p> <hr/> <p>Reverse of a number</p> <p>Enter a number: <input type="text"/></p> <p><input type="button" value="Reverse"/></p>	<p>This site says...</p> <p>No vowels found for string sys</p> <p><input type="button" value="OK"/></p>
<p>← → ⌂ localhost/pwlab/ckr/4.html</p> <p>Finding left most vowel</p> <p>Enter a string: <input type="text"/></p> <p><input type="button" value="Find"/></p> <hr/> <p>Reverse of a number</p> <p>Enter a number: <input type="text" value="1087"/></p> <p><input type="button" value="Reverse"/></p>	<p>This site says...</p> <p>Reverse of num 1087 is 7801</p> <p><input type="button" value="OK"/></p>
<p>← → ⌂ localhost/pwlab/ckr/4.html</p> <p>Finding left most vowel</p> <p>Enter a string: <input type="text"/></p> <p><input type="button" value="Find"/></p> <hr/> <p>Reverse of a number</p> <p>Enter a number: <input type="text" value="aa"/></p> <p><input type="button" value="Reverse"/></p>	<p>This site says...</p> <p>invalid number</p> <p><input type="button" value="OK"/></p>

5. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Branch, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

Explanation:

eXtensible Markup Language (XML) is used in many aspects of web development. It is often used to separate data from presentation. XML does not carry any information about how to be displayed. The same XML data can be used in many different presentation scenarios. Because of this, with XML, there is a full separation between data and presentation. In many HTML applications, XML is used to store or transport data, while HTML is used to format and display the same data.

An XML document with correct syntax is called "Well Formed". An XML document validated against a DTD is both "Well Formed" and "Valid". A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD). XML files suffer though from poor usability by nature as they are not friendly to look at in the browser. Hence they are rendered either using CSS or XSLT (eXtensible Style Sheet Language Transformation from XML to XHTML). CSS does not support new elements to be added or records to be sorted. XSLT is template based and support manipulating contents of XML as well as ordering of XML elements.

External DTD for student: (5.dtd)

```
<!ELEMENT vtu (student+)>
<!ELEMENT student (name,yoj,branch,email*)>
<!ELEMENT name (f,m,l)>
<!ELEMENT yoj (#PCDATA)>
<!ELEMENT branch (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT f (#PCDATA)>
<!ELEMENT m (#PCDATA)>
<!ELEMENT l (#PCDATA)>
<!ATTLIST student usn CDATA #REQUIRED>
<!ATTLIST student college CDATA #IMPLIED>
```

XML data according to DTD of the student

```
(5.xml) <?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/css" href="5.css"?>
<!DOCTYPE vtu SYSTEM "5.dtd"> <vtu>

<student usn="1CD10CS001" college="CITECH">
<name>
<f>Asim</f>
<m> M </m>
<l> L </l>
</name>
<yoj> 2014 </yoj>
<branch> CSE </branch>
```

```
<email> asim@gmail.com </email>
</student>
<student usn="1CD10CS002" college="CITECH">
<name>
  <f>Amar</f>
  <m> M </m>
  <l> L </l>
</name>
<yoj> 2014 </yoj>
<branch> CSE </branch>
<email> amar@gmail.com</email>
</student>
<student usn="1CD10CS003" college="CITECH">
<name>
  <f>Azad</f>
  <m> M </m>
  <l> L </l>
</name>
<yoj> 2014 </yoj>
<branch> CSE </branch>
<email> azad@gmail.com</email>
</student>
</vtu>
```

CSS file for display (5.css)

```
vtu {display:table; border: solid 1px;}
student {display:table-row; border: solid 1px;}
student:before {display:table-cell; border: solid 1px; content:attr(usn);}
student:after {display:table-cell; border: solid 1px; content:attr(college);}
f,m,l,yoj,branch,email {display:table-cell; border: solid 1px;}
```

Output

Checking for well-formedness:

```
xmllint -noout <xml filename>
```

Checking for validity

```
xmllint -noout -valid <xml filename>
```

Raw XML display(without CSS)

XML display with CSS

Enhancement:

XLST can be used than CSS to display all the students in proper format with headings, and student data can be sorted in descending order of USN

XSLT file: (5.xsl)

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml"> <xsl:template match="vtu">
<html>
<body>
```

```

<table border="2pt">
<tr>
<th> USN </th>
<th> First Name </th>
<th> Middle Name </th>
<th> Last Name </th>
<th> Year of Joining </th>
<th> Branch </th>
<th> College </th>
<th> Email-Id </th>
</tr>
<xsl:for-each select="student">
<xsl:sort select="@usn" order="descending"/>
<tr>
<td> <xsl:value-of select="@usn" /> </td>
<td> <xsl:value-of select="name/f" /> </td>
<td> <xsl:value-of select="name/m" /> </td>
<td> <xsl:value-of select="name/l" /> </td>
<td> <xsl:value-of select="yoj" /> </td>
<td> <xsl:value-of select="branch" /> </td>
<td> <xsl:value-of select="@college" /> </td>
<td> <xsl:value-of select="email" /> </td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Output (with modified XSLT file)

6. 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.

Explanation:

With server side technologies, server executes the program (PHP) and sends output to browser. This program using PHP gives the most commonly used script in web-site design i.e keeping track of the number of visitors of a web page. File handling capabilities of PHP

also demonstrated.

HTML file (6.html)

```
<html>
<body >
<iframe src="6.php" frameborder="0" width="2000"></iframe>
</body>
</html>
```

PHP program (6.php)

```
<?php
$f=fopen("count.txt","r+");
$line=fgets($f);
fseek($f,0);
$line++;
echo "Number of times visited the web page ".$line;
fwrite($f,$line);
fclose($f);
?>
```

Functions/Libraries used:

fopen()-method to open files is with the fopen() function. The first parameter of fopen() contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened. The file may be opened in one of the following modes:

Modes	Description
r	Open a file for read only. File pointer starts at the beginning of the file
w	Open a file for write only. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	Open a file for write only. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	Creates a new file for write only. Returns FALSE and an error if file already exists
r+	Open a file for read/write. File pointer starts at the beginning of the file
w+	Open a file for read/write. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	Open a file for read/write. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	Creates a new file for read/write. Returns FALSE and an error if file already exists

fgets() - The fgets() function is used to read a single line from a file.

fseek() - The fseek() function seeks in an open file. This function moves the file pointer from its current position to a new position, forward or backward, specified by the number of bytes. This function returns 0 on success, or -1 on failure. Seeking past EOF will not generate an error.

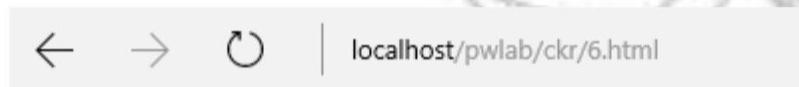
Syntax: fseek(file,offset,whence)

Parameter	Description
-----------	-------------

file	Required. Specifies the open file to seek in
offset	Required. Specifies the new position (measured in bytes from the beginning of the file)
whence	Optional. (added in PHP 4). Possible values: SEEK_SET - Set position equal to offset. Default SEEK_CUR - Set position to current location plus offset SEEK_END - Set position to EOF plus offset (to move to a position before EOF, the offset must be a negative value)

fwrite() - The fwrite() function is used to write to a file. The first parameter of fwrite() contains the name of the file to write to and the second parameter is the string to be written.

Output:



Number of times visited the web page 28

Enhancement:

Displaying number of visitors in digital font uploaded in server and client should not upload any new fonts in its machine

Modified PHP program

```
<?php
header("Content-Type: image/png");
$im = @imagecreate(1000, 800);

//set the background color of the image
$background_color = imagecolorallocate($im, 255, 255, 255);

$f=fopen("count.txt","r+");
$line=fgets($f);
fseek($f,0);
$line++;
$str=sprintf("Number of times visited the web page %04d",$line);
fwrite($f,$line);
fclose($f);

$font="C:/WINDOWS/Fonts/DS-DIGIB.TTF";

$red = imagecolorallocate($im, 255, 0, 0);

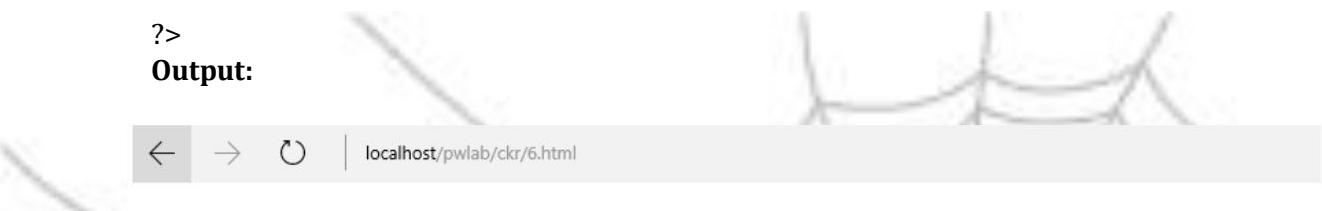
imagettfttext($im, 32, 0, 20, 80, $red,$font, $str);
```

```

imagepng($im);
//frees any memory associated with the image
imagedestroy($im);

?>
Output:

```

localhost/pwlab/ckr/6.html

NUMBER OF TIMES VISITED THE WEB PAGE 0029

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

Explanation

This program fetches the current time from the server. The automatic page refreshing ability of PHP programs based on timers is demonstrated. This feature can be exploited in any online exam scenarios which are time bound and server time should be shown continuously.

Program

HTML file (7.html)

```

<html>
  <body>
    <iframe src="7.php" frameborder="0" width="2000"></iframe>
  </body>
</html>

```

PHP program (7.php)

```

<?php
  date_default_timezone_set('Asia/Calcutta');
  header("Refresh: 1");
  $d=date('H:i:s');
  echo $d;
?>

```

Functions used:

date_default_timezone_set (string \$timezone_identifier) - Sets the default timezone used by all date/time functions in a script.

date()-The date() function formats a local date and time, and returns the formatted date string.

Syntax: date(*format,timestamp*);

Parameter	Description
	Required. Specifies the format of the outputted date string. The following characters can be used:
	d - The day of the month (from 01 to 31)
	D - A textual representation of a day (three letters)
	j - The day of the month without leading zeros (1 to 31)
	l (lowercase 'L') - A full textual representation of a day
	N - The ISO-8601 numeric representation of a day (1 for Monday, 7 for Sunday)
	S - The English ordinal suffix for the day of the month (2 characters st, nd, rd or th. Works well with j)
	w - A numeric representation of the day (0 for Sunday, 6 for Saturday)
	z - The day of the year (from 0 through 365)
	W - The ISO-8601 week number of year (weeks starting on Monday)
	F - A full textual representation of a month (January through December)
	m - A numeric representation of a month (from 01 to 12)
	M - A short textual representation of a month (three letters)
	n - A numeric representation of a month, without leading zeros (1 to 12)
	t - The number of days in the given month
<i>format</i>	L - Whether it's a leap year (1 if it is a leap year, 0 otherwise)
	o - The ISO-8601 year number
	Y - A four digit representation of a year
	y - A two digit representation of a year
	a - Lowercase am or pm
	A - Uppercase AM or PM
	B - Swatch Internet time (000 to 999)
	g - 12-hour format of an hour (1 to 12)
	G - 24-hour format of an hour (0 to 23)
	h - 12-hour format of an hour (01 to 12)
	H - 24-hour format of an hour (00 to 23)
	i - Minutes with leading zeros (00 to 59)
	s - Seconds, with leading zeros (00 to 59)
	u - Microseconds (added in PHP 5.2.2)
	e - The timezone identifier (Examples: UTC, GMT, Atlantic/Azores)
	I (capital i) - Whether the date is in daylights savings time (1 if Daylight Savings Time, 0 otherwise)
	O - Difference to Greenwich time (GMT) in hours (Example: +0100)
	P - Difference to Greenwich time (GMT) in hours:minutes (added in PHP 5.1.3)
	T - Timezone abbreviations (Examples: EST, MDT)

- Z - Timezone offset in seconds. The offset for timezones west of UTC is negative (-43200 to 50400)
- c - The ISO-8601 date (e.g. 2013-05-05T16:34:42+00:00)
- r - The RFC 2822 formatted date (e.g. Fri, 12 Apr 2013 12:01:05 +0200)
- U - The seconds since the Unix Epoch (January 1 1970 00:00:00 GMT)

header()-The header() function sends a raw HTTP header to a client. It is important to notice that header() must be called before any actual output is sent (In PHP 4 and later, you can use output buffering to solve this problem):

Syntax: header(string,replace,http_response_code)

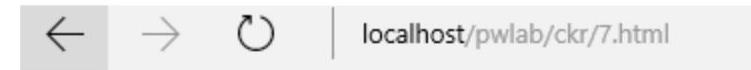
Parameter	Description
string	Required. Specifies the header string to send
replace	Optional. Indicates whether the header should replace previous or add a second header. Default is TRUE (will replace). FALSE (allows multiple headers of the same type)
http_response_code	Optional. Forces the HTTP response code to the specified value (available in PHP 4.3 and higher)

Output:



localhost/pwlab/ckr/7.html

08:42:08 am



localhost/pwlab/ckr/7.html

20:40:35 pm

Enhancement:

The PHP script can be enhanced to graphically display the clock, rendering a true digital clock effect.

Modified PHP code:

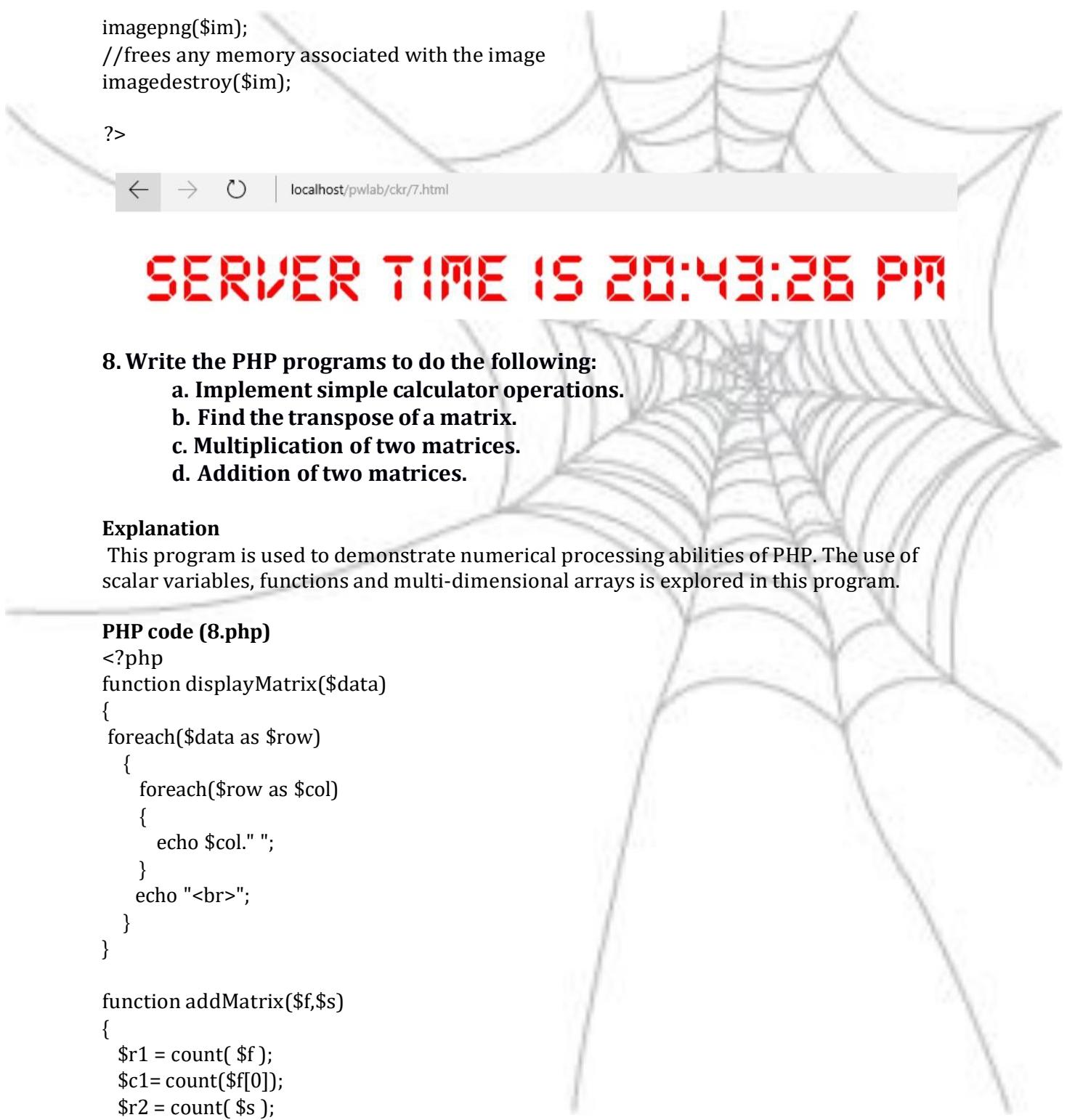
```
<?php
date_default_timezone_set('Asia/Calcutta');
header("Refresh: 1");
header("Content-Type: image/png");
$im = @imagecreate(1600, 800);
$background_color = imagecolorallocate($im, 255, 255, 255);
$str="Server time is ". date('H:i:s a');
$font="C:/WINDOWS/Fonts/DS-DIGIB.TTF";
```

```
$red = imagecolorallocate($im, 255, 0, 0);

imagettftext($im, 70, 0, 20, 100, $red,$font, $str);

imagepng($im);
//frees any memory associated with the image
imagedestroy($im);
```

?>

localhost/pwlab/ckr/7.html

SERVER TIME IS 20:43:26 PM

8. Write the PHP programs to do the following:

- a. Implement simple calculator operations.
- b. Find the transpose of a matrix.
- c. Multiplication of two matrices.
- d. Addition of two matrices.

Explanation

This program is used to demonstrate numerical processing abilities of PHP. The use of scalar variables, functions and multi-dimensional arrays is explored in this program.

PHP code (8.php)

```
<?php
function displayMatrix($data)
{
foreach($data as $row)
{
    foreach($row as $col)
    {
        echo $col." ";
    }
    echo "<br>";
}
}

function addMatrix($f,$s)
{
$r1 = count( $f );
$c1= count($f[0]);
$r2 = count( $s );
$c2= count($s[0]);
$summat=array();
if($r1==$r2 && $c1==$c2)
```

```

{
    for($i=0;$i<$r1;$i++)
    {
        for($j=0;$j<$c1;$j++)
        {
            $summat[$i][$j]=$f[$i][$j]+$s[$i][$j];
        }
    }
    print "Matrix A:<br>";
    displayMatrix($f);
    print "Matrix B:<br>";
    displayMatrix($s);
    print "Sum Matrix:<br>";

    displayMatrix($summat);
}
else
echo "Matrix Addition not possible";
}

function mulMatrix($f,$s)
{
    $r1 = count( $f );
    $c1=count($f[0]);
    $r2 = count( $s );
    $c2= count($s[0]);
    $prod=array();
    //echo $c1." ".$r2;
    if($c1==$r2 )
    {
        for($i=0;$i<$r1;$i++)
        {
            for($j=0;$j<$c2;$j++)
            {
                $prod[$i][$j]=0;
                for($k=0;$k<$c1;$k++)
                {
                    $prod[$i][$j]=$prod[$i][$j]+($f[$i][$k]*$s[$k][$j]);
                }
            }
        }
        print "Matrix A:<br>";
        displayMatrix($f);
        print "Matrix B:<br>";
        displayMatrix($s);
        print "Prouct Matrix:<br>";
        displayMatrix($prod);
    }
}

```

```

else
echo "Matrix Multiplication not possible";

}

function calc($num1,$num2,$opr)
{
switch($opr)
{
    case '+': echo "Sum of $num1 and $num2 is ".($num1+$num2)."  
";break;
    case '-': echo "Difference of $num1 and $num2 is ".($num1-$num2)."  
";break;
    case '*': echo "Product of $num1 and $num2 is ".$num1*$num2."  
";break;
    case '/': echo "Quotient of $num1 by $num2 is ".$num1/$num2."  
";break;
}
}

function transpose($mat)
{
$transpose = array();
$r1 = count( $mat );
$c1=count($mat[0]);

for($i=0;$i<$r1;$i++)
{
    for($j=0;$j<$c1;$j++)
    {
        $transpose[$j][$i] = $mat[$i][$j];

    }
}
print "Given Matrix:<br>";
displayMatrix($mat);
print "Transpose of Matrix:<br>";
displayMatrix($transpose);
}

echo "Demonstration of Calculator...";
$num1=15;
$num2=5;
calc($num1,$num2,'+');
calc($num1,$num2,'-');
calc($num1,$num2,'*');
calc($num1,$num2,'/');
echo "<hr>";
$mat=array(
    array(1,2),
    array(3,4)
);

```

```

transpose($mat);
echo "<hr>";
$fir=array(
    array(1,2),
    array(3,4)
);
$sec=array(
    array(4,3),
    array(5,6)
);
mulMatrix($fir,$sec);
echo "<hr>";
addMatrix($fir,$sec);
?>

```

Functions used:

The `array()` function is used to create an array.

In PHP, there are three types of arrays:

Indexed arrays - Arrays with numeric index

Associative arrays - Arrays with named keys

Multidimensional arrays - Arrays containing one or more arrays

Syntax

Syntax for indexed arrays:

`array(value1,value2,value3,etc.);`

Syntax for associative arrays:

`array(key=>value,key=>value,key=>value,etc.);`

Parameter	Description
<code>key</code>	Specifies the key (numeric or string)
<code>value</code>	Specifies the value

A multidimensional array is an array containing one or more arrays.

PHP understands multidimensional arrays that are two, three, four, five, or more levels deep. However, arrays more than three levels deep are hard to manage for most people.

The dimension of an array indicates the number of indices you need to select an element.

For a two-dimensional array you need two indices to select an element

For a three-dimensional array you need three indices to select an element

count() - The `count()` function returns the number of elements in an array.

Syntax

`count(array,mode);`

Parameter	Description
<code>array</code>	Required. Specifies the array

Optional. Specifies the mode. Possible values:

- mode*
- 0 - Default. Does not count all elements of multidimensional arrays
 - 1 - Counts the array recursively (counts all the elements of multidimensional arrays)
-

Output:

← → ⌂ | localhost/pwlab/ckr/8easy.php

Demonstration of Calculator...Sum of 15 and 5 is 20

Difference of 15 and 5 is 10

Product of 15 and 5 is 75

Quotient of 15 by 5 is 3

Given Matrix:

1 2

3 4

Transpose of Matrix:

1 3

2 4

Matrix A:

1 2

3 4

Matrix B:

4 3

5 6

Product Matrix:

14 15

32 33

Matrix A:

1 2

3 4

Matrix B:

4 3

5 6

Sum Matrix:

5 5

8 10

Enhancement

This program can be enhanced to include PHP interface for the 4 options:

HTML file: (8.html)

```
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="UTF-8">
<title>Demonstrating PHP</title>
</head>

<body>
<section>
<h1>Simple Calculator</h1>
<form action=8.php method=POST>
Number 1 <input type=text name=t1 size=4> <br>
Number 2 <input type=text name=t2 size=4> <br>
Operation <select name=s1>
<option>+</option> <option>-</option> <option>*</option> <option>/</option>
</select> <br>
<input type=submit name=calc value=Calculate> </form>
</section>
<hr>
<section>
<form action=8.php method=POST>

<h1>Transpose of a Matrix</h1>
Matrix: <textarea name=t3 rows=5 cols=40></textarea><br>
<input type=submit name=trans value=Transpose>
</form>

</section>
<hr>
<section>
<form action=8.php method=POST>
<h1>Multiplication of Matrices</h1>
First Matrix: <textarea name=t4 rows=3 cols=20></textarea><br>
Second Matrix: <textarea name=t5 rows=3 cols=20></textarea><br>

<input type=submit name=mul value=Multiply>
</form>

</section>
<hr>
<section>
<form action=8.php method=POST>

<h1>Addition of Matrices</h1>
First Matrix: <textarea name=t6 rows=3 cols=20></textarea><br>
Second Matrix: <textarea name=t7 rows=3 cols=20></textarea><br>

<input type=submit name=add value=Add>
</form>
</section>
```

```
</body>
```

```
</html>
```

Modified PHP code: (8.php)

```
<?php
function displayMatrix($data)
{
foreach($data as $row)
{
    foreach($row as $col)
    {
        echo $col." ";
    }
    echo "<br>";
}
}

function getMatrix($data)
{
$segments = preg_split('/\n/', $data);
$mat = array();
$r = 0;
foreach ($segments as $arr)
{
$c = 0;
$row = preg_split('/\s+/', $arr);
foreach ($row as $cell)
{
$c
$mat[$r][$c] = $cell;
++$c;
}
++$r;
}
return $mat;
}

function addMatrix($f,$s)
{
$r1 = count( $f );
$c1= count($f[0])-1;
$r2 = count( $s );
$c2= count($s[0])-1;
$summat=array();
if($r1==$r2 && $c1==$c2)
{
for($i=0;$i<$r1;$i++)
{
for($j=0;$j<$c1;$j++)
{
$summat[$i][$j] = $f[$i][$j] + $s[$i][$j];
}
}
return $summat;
}
}
```

```

    {
        for($j=0;$j<$c1;$j++)
        {
            $summat[$i][$j]=$f[$i][$j]+$s[$i][$j];
        }
    }
    displayMatrix($summat);
}
else
echo "Matrix Addition not possible";
}

function mulMatrix($f,$s)
{
    $r1 = count( $f );
    $c1=count($f[0])-1;
    $r2 = count( $s );
    $c2= count($s[0])-1;
    $prod=array();
    //echo $c1." ".$r2;
    if($c1==$r2 )
    {
        for($i=0;$i<$r1;$i++)
        {
            for($j=0;$j<$c2;$j++)
            {
                $prod[$i][$j]=0;
                for($k=0;$k<$c1;$k++)
                {
                    $prod[$i][$j]=$prod[$i][$j]+($f[$i][$k]*$s[$k][$j]);
                }
            }
        }
        displayMatrix($prod);
    }
    else
    echo "Matrix Multiplication not possible";
}

if (isset($_POST['calc']))
{
    $num1=$_POST['t1'];
    $num2=$_POST['t2'];
    $opr=$_POST['s1'];
    switch($opr)
    {

```

```

case '+': echo "Sum of $num1 and $num2 is ".($num1+$num2);break;
case '-': echo "Difference of $num1 and $num2 is ".($num1-$num2);break;
case '*': echo "Product of $num1 and $num2 is ".$num1*$num2);break;
case '/': echo "Quotient of $num1 by $num2 is ".$num1/$num2);break;
}
exit(0);
}

if (isset($_POST['trans']))
{
$mat=$_POST['t3'];

$segments = preg_split('/\n/', $mat);
$transpose = array();
$r = 0;
foreach ($segments as $arr)
{
$c = 0;
$row = preg_split('/\s/', $arr);
foreach ($row as $cell)
{
$transpose[$c][$r] = $cell;
++$c;
}
++$r;
}
print "Given Matrix:<br>";
displayMatrix(getMatrix($mat));
print "Transpose of Matrix:<br>";
displayMatrix($transpose);
}

if (isset($_POST['add']))
{
$first=$_POST['t6'];
$second=$_POST['t7'];
$first_mat=getMatrix($first);
$second_mat=getMatrix($second);
print "First Matrix:<br>";
displayMatrix($first_mat);
print "Second Matrix:<br>";
displayMatrix($second_mat);
print "Sum Matrix:<br>";
addMatrix($first_mat,$second_mat);

}

if (isset($_POST['mul']))
{

```

```
$first=$_POST['t4'];
$second=$_POST['t5'];
$first_mat=getMatrix($first);
$second_mat=getMatrix($second);
print "First Matrix:<br>";
displayMatrix($first_mat);
print "Second Matrix:<br>";
displayMatrix($second_mat);
print "Product Matrix:<br>";
mulMatrix($first_mat,$second_mat);
}
?>
```

Output

← → ⌂ | localhost/pwlab/ckr/8.html

Simple Calculator

Number 1
Number 2
Operation

Transpose of a Matrix

Matrix:

Multiplication of Matrices

First Matrix:
Second Matrix:

Addition of Matrices

First Matrix:
Second Matrix:

Calculator Demonstration:

The image shows a 2x2 grid of browser windows. Each window displays a 'Simple Calculator' application with two input fields ('Number 1' and 'Number 2') containing '10' and '9' respectively, and an 'Operation' dropdown set to '+'. The top-left window shows the 'Calculate' button, while the other three show the result of the addition: 'Sum of 10 and 9 is 19'. The bottom-left window shows the 'Calculate' button, while the other two show the result of subtraction: 'Difference of 10 and 9 is 1'. The bottom-left window shows the 'Calculate' button, while the other two show the result of multiplication: 'Product of 10 and 9 is 90'. The bottom-left window shows the 'Calculate' button, while the other two show the result of division: 'Quotient of 10 by 9 is 1.111111111111111'.

Transpose of a Matrix:

Transpose of a Matrix

Matrix:

1	2	3
4	5	2



Given Matrix:

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 2 \end{matrix}$$

Transpose of Matrix:

$$\begin{matrix} 1 & 4 \\ 2 & 5 \\ 3 & 2 \end{matrix}$$
Multiplication of Matrices:**Multiplication of Matrices**

First Matrix:

1	2
3	4

Second Matrix:

5	1
7	3



First Matrix:

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

Second Matrix:

$$\begin{matrix} 5 & 1 \\ 7 & 3 \end{matrix}$$

Product Matrix:

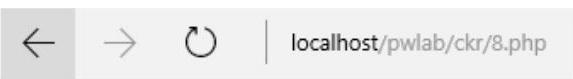
$$\begin{matrix} 19 & 7 \\ 43 & 15 \end{matrix}$$
Addition of Matrices:**Addition of Matrices**

First Matrix:

1	2	3
2	3	1

Second Matrix:

1	1	2
0	-3	4



First Matrix:

1 2 3

2 3 1

Second Matrix:

1 1 2

0 -3 4

Sum Matrix:

2 3 5

2 0 5

9. Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:

- a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.**
- b. Search for a word in states that begins with k and ends in s. Perform a caseinsensitive comparison. [Note: Passing re.Ias a second parameter to method compile performs a case-insensitive comparison.] Store this word in element1 of statesList.**
- c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.**
- d. Search for a word in states that ends in a. Store this word in element 3 of the list.**

Explanation:

This program illustrates the regular expression capabilities in PHP. Writing patterns and matching against a given string is provided. Case sensitive and anchor patterns are explored.

PHP Code: (9.php)

```
<?php
$states="Mississippi Alabama Texas Massachusetts Kansas";
$list=array();
$keywords=preg_split("/\s+/", $states);
foreach ($keywords as $word)
{
if(preg_match('/xas$/i', $word))
{
echo "Word $word ends in xas <br>";
$list[0]=$word;
}
}

foreach ($keywords as $word)
{
```

```

if(preg_match('/^k.*s$/i', $word))
{
    echo "Word $word begins with k and ends s
<br>"; $list[1]=$word;
}

foreach ($keywords as $word)
{
if(preg_match('/^M.*s$/i', $word))
{
    echo "Word $word begins with M and ends s
<br>"; $list[2]=$word;
}

foreach ($keywords as $word)
{
if(preg_match('/a$/i', $word))
{
    echo "Word $word ends with a <br>";
    $list[3]=$word;
}

print_r($list);

?>

```

Functions used:

preg_split() – The preg_split() function operates exactly like split(), except that regular expressions are accepted as input parameters for pattern.

If the optional input parameter limit is specified, then only limit number of substrings are returned.

flags can be any combination of the following flags –

PREG_SPLIT_NO_EMPTY – If this flag is set, only non-empty pieces will be returned by preg_split().

PREG_SPLIT_DELIM_CAPTURE – If this flag is set, parenthesized expression in the delimiter pattern will be captured and returned as well.

PREG_SPLIT_OFFSET_CAPTURE – If this flag is set, for every occurring match the appendant string offset will also be returned.

preg_match() – The preg_match() function searches string for pattern, returning true if pattern exists, and false otherwise.

If the optional input parameter pattern_array is provided, then pattern_array will contain various sections of the subpatterns contained in the search pattern, if applicable.

If this flag is passed as PREG_OFFSET_CAPTURE, for every occurring match the appendant string offset will also be returned

Normally, the search starts from the beginning of the subject string. The optional parameter offset can be used to specify the alternate place from which to start the search.

print_r() - The print_r() function is used to print human-readable information about a variable. If the variable is an integer or a float or a string the function returns value of the variable. If the variable is an array the function returns keys and elements, a similarly notation is used for the object. Setting TRUE to return_output parameter the function returns a string

Output:

```
← → ⏪ | localhost/pwlab/csr/9.php
Word Texas ends in xas
Word Kansas begins with k and ends s
Word Massachusetts begins with M and ends s
Word Alabama ends with a
Array ( [0] => Texas [1] => Kansas [2] => Massachusetts [3] => Alabama )
```

Enhancement:

The program can be configured with a front end HTML:

HTML page: (9.html)

```
<html>
<body>
<h1>Demonstration of PHP Regular Expressions </h1>
<form action=9n.php method=POST>
Enter String <input type=text name=t1
size=70><br> <input type=submit name=sub
value=Find> </form>

</body>
</html>
```

PHP Code: (9n.php)

```
<?php
$states=$_POST['t1'];
echo "<h3>Given sentence is $states </h3><br>";
$list=array();
$keywords=preg_split("/\s+/", $states);
foreach ($keywords as $word)
{
if(preg_match('/xas$/ ', $word))
{
echo "Word $word ends in xas <br>";
$list[0]=$word;
}
}
```

```

}

foreach ($keywords as $word)
{
if(preg_match('/^k.*s$/i', $word))
{
    echo "Word $word begins with k and ends s
<br>"; $list[1]=$word;
}

}

foreach ($keywords as $word)
{
if(preg_match('/^M.*s$/i', $word))
{
    echo "Word $word begins with M and ends s
<br>"; $list[2]=$word;
}

}

foreach ($keywords as $word)
{
if(preg_match('/a$/i', $word))
{
    echo "Word $word ends with a <br>";
    $list[3]=$word;
}
}

echo "<h3>List contents: ". implode(',',$list);
?>

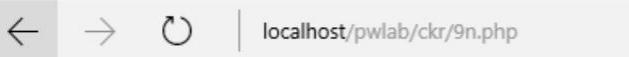
```

Output:

| localhost/pwlab/ckr/9.html

Demonstration of PHP Regular Expressions

Enter String



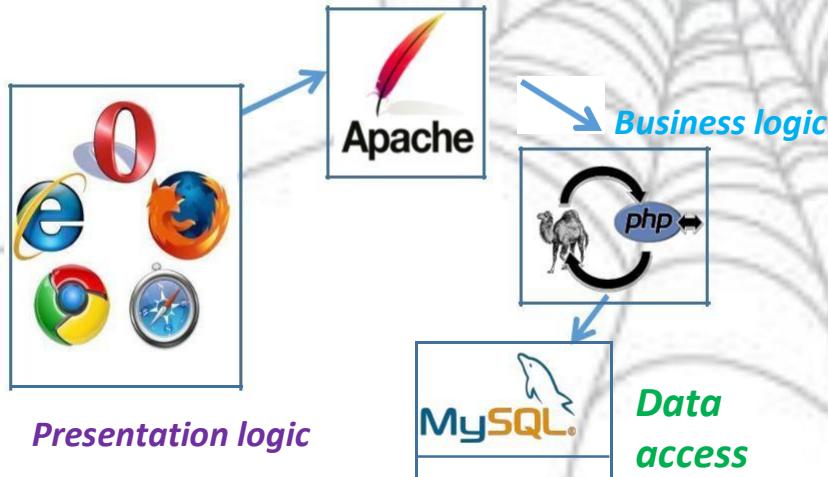
Given sentence is Mississippi Alabama Texas Massachusetts Kansas

Word Texas ends in xas
 Word Kansas begins with k and ends s
 Word Massachusetts begins with M and ends s
 Word Alabama ends with a

List contents: Texas, Kansas, Massachusetts, Alabama

10. Write a PHP program to sort the student records which are stored in the database using selection sort.

This program brings out the use of PHP in developing dynamic web pages. The web-application required follows 3-tier architecture as shown in the figure below. Three tiers include Presentation logic (HTML with CSS), Business logic (PHP) and Data access (MySQL). The choice of MySQL is because its open source, light weight and platform independent.



PHP Program:

```
<?php

function selection_sort($data,$pos)
{
for($i=0; $i<count($data)-1; $i++) {
    $min = $i;
    for($j=$i+1; $j<count($data); $j++) {
        if ($data[$j][$pos]<$data[$min][$pos]) {
            $min = $j;
        }
    }
    $data = swap_positions($data, $i, $min);
}
}
```

```

return $data;
}

function swap_positions($data1, $left, $right) {
    for($j=0;$j<3;$j++)
    {
        $backup_old_data_right_value = $data1[$right][$j];
        $data1[$right][$j] = $data1[$left][$j];
        $data1[$left][$j] = $backup_old_data_right_value;

    }
    return $data1;
}

echo <<<END
<form action=10.php method=POST>
Student USN: <input type=text name=t1 size=15 maxlen=10><br>
Student Name:<input type=text name=t2 size=25><br>
Student CGPA::<input type=text name=t3 size=5><br>
<input type=submit name=ins value=Insert>
&nbsp;&nbsp;
<input type=submit name=sname
value=SortByName> &nbsp;&nbsp;
<input type=submit name=smarks value=SortByMarks>
</form>
END;
if (isset($_POST['ins']))
{
try{

$con="mysql:host=localhost;dbname=studentdb";
$pdo=new PDO($con,"root","");
$usn=$_POST['t1'];
$name=$_POST['t2'];
$marks=$_POST['t3'];
$sql="INSERT into student values('$usn','$name','$marks')";
$result=$pdo->query($sql);
echo "inserted...";

}
catch(Exception $e)
{
    echo "DB Error." . $e;
}
}

```

```
if (isset($_POST['sname']))
{
try{
$student_data=array();
$con="mysql:host=localhost;dbname=studentdb";
$pdo=new PDO($con,"root","");
$sql="SELECT *
from student"; $result=$pdo->query($sql); $cnt=0;

while(($row=$result->fetch()))
{
    $student_data[$cnt][0]=$row['usn'];
    $student_data[$cnt][1]=$row['name'];
    $student_data[$cnt][2]=$row['cgpa'];
    ++$cnt;
}
$student_data=selection_sort($student_data,1);
echo "<table border><tr><th>USN</th>Name<th>CGPA</th></tr>";
for($j=0;$j<$cnt;$j++)
{
    $usn=$student_data[$j][0];
    $name=$student_data[$j][1];
    $marks=$student_data[$j][2];
    echo "<tr><td>$usn<td>$name<td>$marks";
}
}
catch(Exception $e)
{
    echo "DB Error";
}

}

if (isset($_POST['smarks']))
{
try{
$student_data=array();
$con="mysql:host=localhost;dbname=studentdb";
$pdo=new PDO($con,"root","");
$sql="SELECT *
from student"; $result=$pdo->query($sql); $cnt=0;

while(($row=$result->fetch()))
{
    $student_data[$cnt][0]=$row['usn'];
    $student_data[$cnt][1]=$row['name'];
    $student_data[$cnt][2]=$row['cgpa'];
}
```

```

++$cnt;
}
$student_data=selection_sort($student_data,2);
echo "<table border><tr><th>USN<th>Name<th>CGPA</tr>";
for($j=0;$j<$cnt;$j++)
{
    $usn=$student_data[$j][0];
    $name=$student_data[$j][1];
    $marks=$student_data[$j][2];
    echo "<tr><td>$usn<td>$name<td>$marks";
}
}
catch(Exception $e)
{
    echo "DB Error";
}
?
?>

```

Functions/Libraries used:

The PHP Data Objects (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP. Each database driver that implements the PDO interface can expose database-specific features as regular extension functions. Note that you cannot perform any database functions using the PDO extension by itself; you must use a database-specific PDO driver to access a database server.

PDO provides a *data-access* abstraction layer, which means that, regardless of which database you're using, you use the same functions to issue queries and fetch data. PDO does *not* provide a *database* abstraction; it doesn't rewrite SQL or emulate missing features. You should use a full-blown abstraction layer if you need that facility. PDO ships with PHP 5.1, and is available as a PECL extension for PHP 5.0; PDO requires the new OO features in the core of PHP 5, and so will not run with earlier versions of PHP.

Connections are established by creating instances of the PDO base class. It doesn't matter which driver you want to use; you always use the PDO class name. The constructor accepts parameters for specifying the database source (known as the DSN) and optionally for the username and password (if any). If there are any connection errors, a *PDOException* object will be thrown. You may catch the exception if you want to handle the error condition, or you may opt to leave it for an application global exception handler that you set up via **set_exception_handler()**

new PDO(..) – establish a new connection with specified credentials
query() – Execute specified SQL query
fetch() – fetch result of SQL query record at a time

Output

Preparing Back End(Database):

Creating table in phpmyadmin:

Database:

← Server: mysql wampserver » Database: studentdb

Table	Action	Rows	Type	Collation
student	Browse Structure Search Insert Empty Drop	~4	InnoDB	latin1_swedish_ci
1 table	Sum			

Table:

← Server: mysql wampserver » Database: studentdb » Table: student

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	usn	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary
2	name	text	latin1_swedish_ci		No	None		Change Drop Primary
3	cgpa	float			No	None		Change Drop Primary

Front-end interface

← → ⌂ localhost/pwlab/ckr/10.php

Student USN:

Student Name:

Student CGPA:

[Insert](#) [SortByName](#) [SortByMarks](#)

Result after insertion:

← → ⌂ localhost/pwlab/ckr/10.php

Student USN:

Student Name:

Student CGPA:

[Insert](#) [SortByName](#) [SortByMarks](#)

← → ⌂ localhost/pwlab/ckr/10.php

Student USN:

Student Name:

Student CGPA:

[Insert](#) [SortByName](#) [SortByMarks](#)

inserted...

Backend result:

```
SELECT * FROM `student`
```

Number of rows: 25

Sort by key: None

+ Options

	usn	name	cgpa
<input type="checkbox"/>	Edit Copy Delete	4JN15CS009 Kelvin	7.5
<input type="checkbox"/>	Edit Copy Delete	4JN15CS101 Mou	7.2
<input type="checkbox"/>	Edit Copy Delete	4JN15CS102 amu	9.2
<input type="checkbox"/>	Edit Copy Delete	4JN15CS103 Jamu	4.5
<input type="checkbox"/>	Edit Copy Delete	4JN15CS104 Keerti	9.6

**Selection sort by Name:**

← → ⚡ | localhost/pwlab/ckr/10.php

Student USN:
 Student Name:
 Student CGPA:

USN	Name	CGPA
4JN15CS103	Jamu	4.5
4JN15CS104	Keerti	9.6
4JN15CS009	Kelvin	7.5
4JN15CS101	Mou	7.2
4JN15CS102	amu	9.2

Selection sort by CGPA:

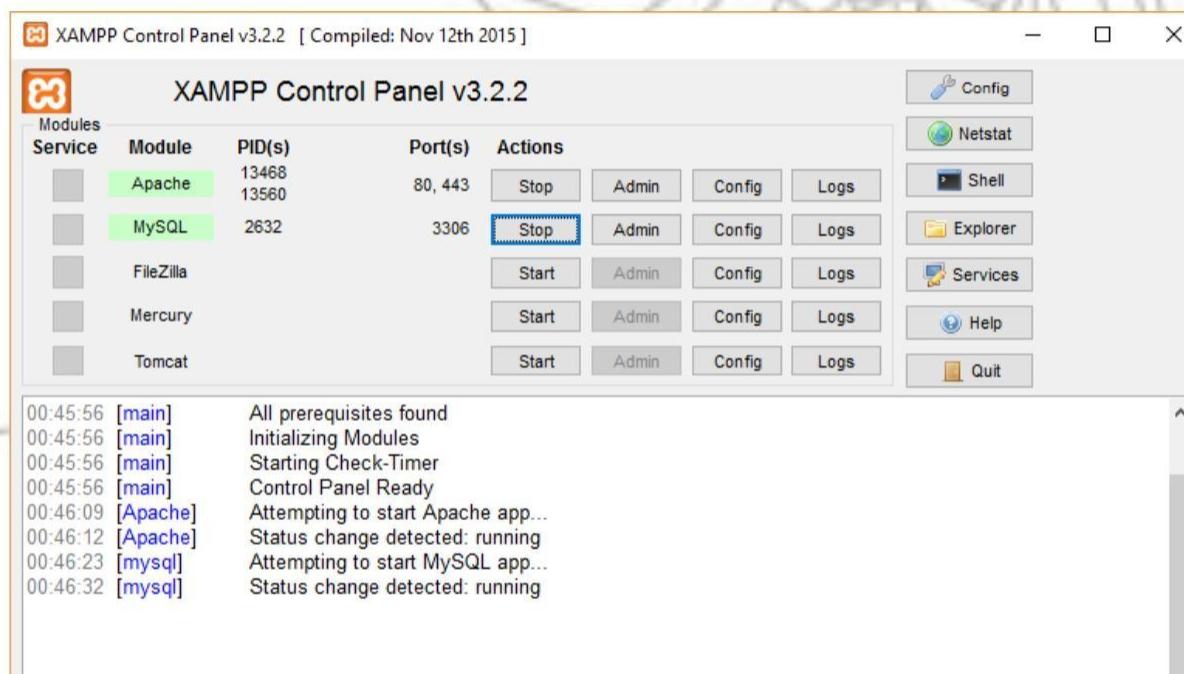
← → ⚡ | localhost/pwlab/ckr/10.php

Student USN:
 Student Name:
 Student CGPA:

USN	Name	CGPA
4JN15CS103	Jamu	4.5
4JN15CS101	Mou	7.2
4JN15CS009	Kelvin	7.5
4JN15CS102	amu	9.2
4JN15CS104	Keerti	9.6

About XAMP

XAMPP (/ˈzæmp/ or /ˈɛks.æmp/[2]) is a free and open source cross-platform web server solution stack package developed by Apache Friends,[2] consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.[3][4] XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.



Lab Working Environment:

The Web Programming Lab is implemented as Client/Server environment. XAMP software is installed only in a server with an IP Address connected by a LAN with many client machines. For each student a login is created and a folder is created to save the programs and scripts of the students. A login screen is presented, when the student enters URL of remote machine:

localhost/pwlab/login1.htm

Web Programming Lab

Login

Name	<input type="text" value="ckr"/>
Password	<input type="password"/>
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

Once login is successful, following editor screen is presented:

localhost/pwlab/checkloginnew1.php

Upload File	<input type="file" value="Choose File"/> No file chosen
File Name	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Open"/> <input type="button" value="Upload"/> <input type="button" value="Compile-Perl"/> <input type="button" value="Compile-PHP"/> <input type="button" value="View"/> <input type="button" value="List"/> <input type="button" value="CheckWellFormedNess"/> <input type="button" value="Validate"/> <input type="button" value="LineNumbers"/>	

Open - Used to open the file specified in the textbox beside Filename
Save - Save the code typed in the textarea into a file name specified

Upload – used to upload a file. One should have selected file in the Browse button beside the label Upload File

Compile-Perl – used to compile Perl program

Compile-PHP – used to compile PHP program

View – to view the output of XHTML/Perl/PHP/XML in browser

List – to display list of files belonging to the user

Checkwellformedness – to check whether XML file is well-formed

Validate – to check whether XML confirms to specific DTD

LineNumbers – to display line numbers in the editor textarea

Viva Questions and Answers:

1.	What are tags? Content is placed in between HTML tags in order to properly format it. It makes use of the less than symbol (<) and the greater than symbol (>).
2.	Do all HTML tags come in pair? No, there are single HTML tags that does not need a closing tag. Examples are the < img> tag and < br> tags.
3.	What is the difference between HTML elements and tags? HTML elements communicate to the browser how to render text. When surrounded by angular brackets < > they form HTML tags. For the most part, tags come in pairs and surround text.
4.	What is “Semantic HTML”? <ul style="list-style-type: none"> • Semantic HTML is a coding style where the tags embody what the text is meant to convey. • In Semantic HTML, tags like < b> < /b> for bold, and < i> < /i> for italic should not be used, reason being they just represent formatting, and provide no indication of meaning or structure. • The semantically correct thing to do is use < strong> < /strong> and < em> < /em> . • These tags will have the same bold and italic effects, while demonstrating meaning and structure (emphasis in this case).
5.	How do you insert a copyright symbol on a browser page? To insert the copyright symbol, you need to type © or & #169; in an HTML file.
6.	Is there any way to keep list elements straight in an html file? <ul style="list-style-type: none"> • By using indents, you can keep the list elements straight. • If you indent each subnested list in further than the parent list that contains it, you can at a glance determine the various lists and the elements that it contains.
7.	Does a hyperlink apply to text only? No, hyperlinks can be used on text as well as images. That means you can convert an image into a link that will allow user to link to another page when clicked. Just surround the image within the < a href=" "> ...< /a> tag combinations.
8.	What are style sheets? Style sheets enable you to build consistent, transportable, and well-defined style templates. These templates can be linked to several different web pages, making it easy to maintain and change the look and feel of all the web pages within a site.
9.	What bullet types are available? With ordered lists, you can select to use a number of different list types including alphabetical and Roman numerals. The type attribute for unordered lists can be set to disc, square, or circle.
10.	How do you make comments without text being picked up by the browser? Comments are used to explain and clarify code or to prevent code from being recognized by the browser. Comments start with “*<!--” and end with “-->”.

	<!-- Insert comment here. -->
11.	What is span in HTML? The SPAN having notation as < SPAN > < /SPAN > is used for highlighting text of any color desired, for adding colored text, for adding background image to text. SPAN doesn't cause a line break. It delimits text and it allows styles to be applied to a 'elemental' region, or for the 'elemental' region to be identified, without causing a break in the text flow.
12.	Is CSS case sensitive? <ul style="list-style-type: none">• Cascading Style Sheets (CSS) is not case sensitive.• However, font families, URLs to images, and other direct references with the style sheet may be.• The trick is that if you write a document using an XML declaration and an XHTML doctype, then the CSS class names will be case sensitive for some browsers.
13.	What are the limitations of CSS ? Limitations are: <ul style="list-style-type: none">• Ascending by selectors is not possible• Limitations of vertical control• No expressions• No column declaration• Pseudo-class not controlled by dynamic behavior• Rules, styles, targeting specific text not possible
14.	What is the difference between an ID selector and CLASS? An ID selector identifies and sets style to only one occurrence of an element, while CLASS can be attached to any number of element
15.	What is Contextual Selector? <ul style="list-style-type: none">• Contextual selector addresses specific occurrence of an element.• It is a string of individual selectors separated by white space (search pattern), where only the last element in the pattern is addressed providing it matches the specified context
16.	What are Pseudo Classes? Pseudo classes allow you to identify HTML elements on characteristics (as opposed to their name or attributes). The classes are specified using a colon to separate the element name and pseudo class. A good example is the :link and :visited pseudo classes for the HTML A element. Another good example is first-child, which finds an element's first child element.
	What does z-index do? The z-index property specifies the stack order of an element. An element with greater stack order is always in front of an element with a lower stack order. Here is the Example, where using z-index property you can display in front of image.
17.	What is parent-child selector? Parent-child selector is a selector representing the direct descendent of a parent element. Parent-child selectors are created by listing two or more tilde (~) separated selectors. <code>BODY ~ P {background: red; color: white}</code> The P element will be declared the specified style only if it directly descends

	from the BODY element:
18.	Enumerate the differences between Java and JavaScript? Java is a complete programming language. In contrast, JavaScript is a coded program that can be introduced to HTML pages. These two languages are not at all inter-dependent and are designed for the different intent. Java is an object-oriented programming (OOPS) or structured programming language like C++ or C whereas JavaScript is a client-side scripting language.
19.	What is the use of isNaN function? isNaN function returns true if the argument is not a number otherwise it is false.
20.	What is negative infinity? Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.
21.	What are global variables? How are these variable declared and what are the problems associated with using them? Global variables are those that are available throughout the length of the code, that is, these have no scope. The var keyword is used to declare a local variable or object. If the var keyword is omitted, a global variable is declared.
22.	What is a prompt box? A prompt box is a box which allows the user to enter input by providing a text box. Label and box will be provided to enter the text or number.
23.	What is === operator? === is called as strict equality operator which returns true when the two operands are having the same value without any type conversion.
24.	What is called Variable typing in Javascript? Variable typing is used to assign a number to a variable and the same variable can be assigned to a string.
25.	What do mean by NULL in Javascript? The NULL value is used to represent no value or no object. It implies no object or null string, no valid boolean value, no number and no array object.
26.	What is the difference between an alert box and a confirmation box? An alert box displays only one button which is the OK button. But a Confirmation box displays two buttons namely OK and cancel.
27.	Explain what is pop() method in JavaScript? The pop() method is similar as the shift() method but the difference is that the Shift method works at the start of the array. Also the pop() method take the last element off of the given array and returns it. The array on which is called is then altered. What are the two basic groups of datatypes in JavaScript? They are as – <ul style="list-style-type: none"> • Primitive • Reference types
28.	What is the use of type of operator? 'Typeof' is an operator which is used to return a string description of the type of a variable.
29.	What is the use of blur function? Blur function is used to remove the focus from the specified object.
30.	Explain window.onload and onDocumentReady? The onload function is not run until all the information on the page is loaded.

	This leads to a substantial delay before any code is executed. <code>onDocumentReady</code> loads the code just after the DOM is loaded. This allows early manipulation of the code.
31.	Define event bubbling? JavaScript allows DOM elements to be nested inside each other. In such a case, if the handler of the child is clicked, the handler of parent will also work as if it were clicked too.
32.	How are event handlers utilized in JavaScript? Events are the actions that result from activities, such as clicking a link or filling a form, by the user. An event handler is required to manage proper execution of all these events. Event handlers are an extra attribute of the object. This attribute includes event's name and the action taken if the event takes place.
33.	What is the correct and the most two common way to start and finish a PHP block of code? The two most common ways to start and finish a PHP script are: <code><?php [--- PHP code----] ?></code> and <code><? [--- PHP code ---] ?></code>
34.	How can we display the output directly to the browser? To be able to display the output directly to the browser, we have to use the special tags <code><?=</code> and <code>?></code> .
35.	How can PHP and HTML interact? It is possible to generate HTML through PHP scripts, and it is possible to pass pieces of information from HTML to PHP.
36.	What type of operation is needed when passing values through a form or an URL? If we would like to pass values through a form or an URL, then we need to encode and to decode them using <code>htmlspecialchars()</code> and <code>urlencode()</code> .
37.	How can PHP and Javascript interact? PHP and Javascript cannot directly interact since PHP is a server side language and Javascript is a client-side language. However, we can exchange variables since PHP can generate Javascript code to be executed by the browser and it is possible to pass specific variables back to PHP via the URL.
38.	What is the main difference between require() and require_once()? <code>require()</code> , and <code>require_once()</code> perform the same task except that the second function checks if the PHP script is already included or not before executing it.
39.	What does the PHP error 'Parse error in PHP - unexpected T_variable at line x' means? This is a PHP syntax error expressing that a mistake at the line x stops parsing and executing the program.
40.	What is the function file_get_contents() useful for? <code>file_get_contents()</code> lets reading a file and storing it in a string variable.
41.	How is it possible to know the number of rows returned in the result set? The function <code>mysqli_num_rows()</code> returns the number of rows in a result set.
42.	Which function gives us the number of affected entries by a query? <code>mysqli_affected_rows()</code> return the number of entries affected by an SQL query.
43.	What is the difference between mysqli_fetch_object() and mysqli_fetch_array()? The <code>mysqli_fetch_object()</code> function collects the first single matching record where <code>mysqli_fetch_array()</code> collects all matching records from the table in an array.

44.	<p>How can we access the data sent through the URL with the POST method?</p> <p>To access the data sent this way, you use the <code>\$_POST</code> array. Imagine you have a form field called 'var' on the form when the user clicks submit to the post form, you can then access the value like this:</p> <pre><code>\$_POST["var"];</code></pre>
45.	<p>How do I escape data before storing it in the database?</p> <p>The addslashes function enables us to escape data before storage into the database.</p>
46.	<p>What is the static variable in function useful for?</p> <p>A static variable is defined within a function only the first time, and its value can be modified during function calls as follows:</p>
47.	<p>How to initiate a session in PHP?</p> <p>A session is a logical object enabling us to preserve temporary data across multiple PHP pages. The use of the function <code>session_start()</code> lets us activating a session.</p>
48.	<p>How can you propagate a session id?</p> <p>You can propagate a session id via cookies or URL parameters.</p>
49.	<p>What does <code>\$_SERVER</code> mean?</p> <p><code>\$_SERVER</code> is an array including information created by the web server such as paths, headers, and script locations.</p>
50.	<p>What does <code>\$_FILES</code> means?</p> <p><code>\$_FILES</code> is an associative array composed of items sent to the current script via the HTTP POST method.</p>
51.	<p>How can we pass the variable through the navigation between the pages?</p> <p>It is possible to pass the variables between the PHP pages using sessions, cookies or hidden form fields.</p>