Program 1:
Develop and demonstrate a XHTML document that illustrates the use external style sheet, ordered list, table, borders, padding, color, and the <span> tag.

Basics:

- Every XHTML document must begin with:
  <?xml version = ″1.0″?>
  <!DOCTYPE html PUBLIC ″-//w3c//DTD XHTML 1.1//EN″
  http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>
- <html>, <head>, <title>, and <body> are required in every document
- The whole document must have <html> as its root and html must have the xmlns attribute:
  <html xmlns = ″http://www.w3.org/1999/xhtml″
- A document consists of a head and a body
- The <title> tag is used to give the document a title, which is normally displayed in the browser's window title bar (at the top of the display)

Lists:

- Unordered lists
  - The list is the content of the <ul> tag
  - List elements are the content of the <li> tag
- *Ordered lists*
  - The list is the content of the <ol> tag
  - Each item in the display is preceded by a sequence value

Table:

- A table is a matrix of cells, each possibly having content
- The cells can include almost any element
- Some cells have row or column labels and some have data
- A table is specified as the content of a <table> tag
- A border attribute in the <table> tag specifies a border between the cells
- If border is set to "border", the browser's default width border is used
- The border attribute can be set to a number, which will be the border width
- Without the border attribute, the table will have no lines!
- Each row of a table is specified as the content of a <tr> tag
- The row headings are specified as the content of a <th> tag
- The contents of a data cell is specified as the content of a <td> tag

Span:

- The <span> tag is similar to other HTML tags, they can be nested and they have id and class attributes.

Solution:

// style.css

```
p,table,li,
{
font-family: "lucida calligraphy", arial, 'sans serif';
margin-left: 10pt;
}

p { word-spacing: 5px; }

body { background-color:rgb(200,255,205); }

p,li,td { font-size: 75%;}

td { padding: 0.5cm; }

th {
     text-align:center;
     font-size: 85%;
   }

h1, h2, h3, hr {color:#483d8b;}

table
{
border-style: outset;
background-color: rgb(100,255,105);
}

li {list-style-type: lower-roman;}

span
{
color:blue;
background-color:pink;
font-size: 29pt;
font-style: italic;
font-weight: bold;
}
```

<!-- **lab1.html** -->

```
<?xml version = "1.0" encoding = "utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<link rel="stylesheet" type="text/css" href="style.css" />
<title> Lab program1 </title>
</head>
<body>
  <h1>This header is 36 pt</h1>
  <h2>This header is blue</h2>
  <p>This paragraph has a left margin of 50 pixels</p>

<table border="4" width="5%">   <!-- table with name & email -->
 <tr>
          <th width="204">Name </th>
          <th>Email</th>
 </tr>
 <tr>
          <td width="204">Dr. HNS</td>
          <td>abc@pes.edu</td>
 </tr>
 <tr>
          <td width="204">Dr. MKV</td>
          <td>xyz@rediffmail.com</td>
 </tr>
 <tr>
          <td width="204">Dr. GTR</td>
          <td>aaa@yahoo.co.in</td>
 </tr>
 <tr>
          <td width="204">Dr. MVS</td>
          <td>bbb@hotmail.com</td>
 </tr>
</table>
<hr>      <!-- horizontal line -->
<ol> <!-- ordered list -->
 <li> Gowtham</li>
 <li> Gowrav </li>
 <li> Gopalakrishna </li>
</ol>

<p>
<span>This is a text.</span> This is a text. This is a text. This is a text. This is a text. This is a text. This is a text.
This is a text. This is a text. <span>This is a text.</span>
</p>
</body>
</html>
```

Program 2:
Develop and demonstrate a XHTML file that includes Javascript script for the following problems:
(a) Input: A number n obtained using prompt
Output: The first n Fibonacci numbers
(b) Input: A number n obtained using prompt
Output: A table of numbers from 1 to n and their squares using alert

Basics:

XHTML:

- An XHTML document is a mix of content and controls
- Controls are tags and their attributes
- Tags often delimit content and specify something about how the content should be arranged in the document
- Attributes provide additional information about the content of a tag

JAVASCRIPT:

- A client-side HTML-embedded scripting language
- Only related to Java through syntax
- Dynamically typed and not object-oriented
- Provides a way to access elements of HTML documents and dynamically change them

JavaScript in XHTML:

- Directly embedded
```
<script type="text/javascript">
        <!--
                …Javascript here…
        -->
</script>
```
- Indirect reference
```
<script type="text/javascript" src="tst_number.js"/>
```

Solution:

```
<!-- lab2a.html  -->

<html>
<head>
<title>Program 2a</title>
</head>

<script type="text/javascript">
<!--

function fib()
{
var n = prompt("Enter N: ","");

fib1=0;
fib2=1;
fib=0;
document.write("<h2>" +fib1 +"\n</h2>");
document.write("<h2>" +fib2 +"\n</h2>");

for(i=0;i<n;i++)
{
 fib = fib1+fib2;
 fib1=fib2;
 fib2=fib;
 document.write("<h2>" +fib +"\n</h2>");
}
}
-->
</script>
<body onload="fib()">
</body>

<!-- lab2b.html-->

<html>
<head>
<title>Program 2b</title>
</head>

<script type="text/javascript">
```

```
<!--

function sqr()
{
var n = prompt("Enter N: ","");
msgstr ="The square of numbers from 1 to "+n +" is: \n";

for(i=1;i<=n;i++)
  {
 msgstr = msgstr +i +"------->" +i*i +"\n";
  }
alert(msgstr);

}
-->
</script>
<body onload="sqr()">
</body>
```

Program 3:

Develop and demonstrate a XHTML 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.

Solutions:

<!-- lab3a.html -->

```
<html>
<head>
<title>Program 3a</title>
</head>

<script type="text/javascript">
<!--

function vovel()
{
var n = prompt("Enter a string: ","");
flag=0;
for(i=0;i<n.length && flag!=1 ;i++)
{
        switch(n[i])
        {
                case 'a':
                case 'e':
                case 'i':
                case 'o':
                case 'u': alert("The left most vowel is present in the position: " +(i+1));
                          flag = 1;
                          break;
                default : break;
        }
}
if(!flag)
                alert("No Vowels found.");
}
```

```
-->
</script>
<body onload="vovel()">
</body>
```

<!-- lab3b.html -->

```
        <html>
        <head>
        <title>Program 3b</title>
        </head>

        <script type="text/javascript">
        <!--

        function rev()
        {
        var n = prompt("Enter a number: ","");
        var str=0;
        for(i=n.length-1;i>=0;i--)
                str = str*10 + Number(n[i]);

        alert(str);
        }

        -->
        </script>
        <body onload="rev()">
        </body>
```

Program 4:

(a) Develop and demonstrate, using Javascript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.

(b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8)

Basics:

Events and Event Handling:

- Event-driven programming is a style of programming in which pieces of code, event handlers, are written to be activated when certain events occur
- Events represent activity in the environment including, especially, user actions such as moving the mouse or typing on the keyboard
- An event handler is a program segment designed to execute when a certain event occurs
- Events are represented by JavaScript objects

Setting a Handler:

- Using a an attribute, a JavaScript command can be specified:
  ```
  <input type=“button” name=“myButton”
  onclick=
  “alert(‘You clicked the button!’)”/>
  ```
- A function call can be used if the handler is longer than a single statement
  ```
  <input type=“button” name=“myButton”
  onclick=“myHandler()”/>
  ```

Solutions:

```
<!-- lab4a.html -->

        <html>
        <head>
        <title>Program 4a</title>
        </head>

        <script type="text/javascript">
        <!--
```

```
        function validate()
        {

        var pattern=/[1-4][A-Z][A-Z][0-9][0-9][A-z][A-Z][0-9][0-9][0-9]$/;
        usn = document.getElementById('usn');

        if( !usn.value.match(pattern))
                alert("Invalid USN");
        else
                alert("Valid USN");
        }

        -->
        </script>
        <body>
        <form>
        Enter USN: <input type="text" name="usn" id="usn" />
        <input type="submit" value="Check" onclick="validate()"/>
        </form>
</body>

<!-- lab4b.html -->

<html>
<head>
<title>Program 4b</title>
</head>

<script type="text/javascript">
<!--

function validate()
{

var pattern1=/[1-4][A-Z][A-Z][0-9][0-9][A-z][A-Z][0-9][0-9][0-9]$/;
var pattern2=/[1-8]/;
usn = document.getElementById('usn');
sem = document.getElementById('sem');
if( !usn.value.match(pattern1))
        alert("Invalid USN");
else
        alert("Valid USN");
if( !sem.value.match(pattern2))
        alert("Invalid SEM");
else
        alert("Valid SEM");


}

-->
</script>
<body>
<form>
Enter USN: <input type="text" name="usn" id="usn" />
Enter SEM: <input type="text" name="sem" id="sem" />
<input type="submit" value="Check" onclick="validate()"/>
</form>
</body>
```

Program 5:
(a) Develop and demonstrate using a JavaScript, a XHTML document that contains 3 shot paragraphs of text, stacked on top of one another, with only enough of each showing so that the mouse cursor can be placed over some part of them. When cursor is placed over the exposed part of any paragraph it should rise to the top to become completely visible.
(b) Modify the above document so that when a paragraph is moved from top stacking position, it returns to its original position rather to the bottom.

<!—**lab5a.js: This is for 5a program** -->

```
// JavaScript Document
var top="a1";
function toTop(newTop)
{
        domTop = document.getElementById(top).style;
        domNew = document.getElementById(newTop).style;
        domTop.zIndex="0";
        domNew.zIndex="10";
        top=newTop;
}
```

<!-- **lab5a.html** -->

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Stack</title>
<script type="text/javascript" src="lab5a.js">
</script>
<style type="text/css">
.box1 { position:absolute; top:0; left:0; z-index:0; background-color:#FF0000 }
.box2 { position:absolute; top:50px; left:100px; z-index:0; background-color:#0000FF}
.box3 { position:absolute; top:100px; left:200px; z-index:0; background-color:#FFFF00}
</style>
</head>

<body>
<p>
<p class="box1" id="a1" onmouseover="toTop('a1')">
111111111111111111111111111111111111111111111<br />
111111111111111111111111111111111111111111111<br />
111111111111111111111111111111111111111111111<br />
```

```
1111111111111111111111111111111111111111111111<br />
1111111111111111111111111111111111111111111111<br />
1111111111111111111111111111111111111111111111<br />
</p>
<p class="box2" id="a2" onmouseover="toTop('a2')">
2222222222222222222222222222222222222222222222<br />
2222222222222222222222222222222222222222222222<br />
2222222222222222222222222222222222222222222222<br />
2222222222222222222222222222222222222222222222<br />
2222222222222222222222222222222222222222222222<br />
2222222222222222222222222222222222222222222222<br />
</p>
<p class="box3" id="a3" onmouseover="toTop('a3')" >
3333333333333333333333333333333333333333333333<br />
3333333333333333333333333333333333333333333333<br />
3333333333333333333333333333333333333333333333<br />
3333333333333333333333333333333333333333333333<br />
3333333333333333333333333333333333333333333333<br />
3333333333333333333333333333333333333333333333<br />
</p>
</p>
</body>
</html>
```

<!—**lab5b.js: This is for 5b program** -->

```
        var top="a1";
        function toTop(newTop)
        {
                domTop = document.getElementById(top).style;
                domNew = document.getElementById(newTop).style;
                domTop.zIndex="0";
                domNew.zIndex="10";
                top=newTop;
        }

        function orignal()
        {
                dom1 = document.getElementById('a1').style;
                dom2 = document.getElementById('a2').style;
                dom3 = document.getElementById('a3').style;
                dom1.zIndex="0";
                dom2.zIndex="1";
                dom3.zIndex="2";
```

```
}

<!-- lab5b.html -->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Stack</title>
<script type="text/javascript" src="lab5b.js">
</script>
<style type="text/css">
.box1 { position:absolute; top:0; left:0; z-index:0; background-color:#FF0000 }
.box2 { position:absolute; top:50px; left:100px; z-index:1; background-color:#0000FF}
.box3 { position:absolute; top:100px; left:200px; z-index:2; background-color:#FFFF00}
</style>
</head>

<body>
<p>
<p class="box1" id="a1" onmouseover="toTop('a1')" onmouseout="orignal()">
11111111111111111111111111111111111111111111111<br />
11111111111111111111111111111111111111111111111<br />
11111111111111111111111111111111111111111111111<br />
11111111111111111111111111111111111111111111111<br />
11111111111111111111111111111111111111111111111<br />
11111111111111111111111111111111111111111111111<br />
</p>
<p class="box2" id="a2" onmouseover="toTop('a2')" onmouseout="orignal()">
22222222222222222222222222222222222222222222222<br />
22222222222222222222222222222222222222222222222<br />
22222222222222222222222222222222222222222222222<br />
22222222222222222222222222222222222222222222222<br />
22222222222222222222222222222222222222222222222<br />
22222222222222222222222222222222222222222222222<br />
</p>
<p class="box3" id="a3" onmouseover="toTop('a3')" onmouseout="orignal()">
33333333333333333333333333333333333333333333333<br />
33333333333333333333333333333333333333333333333<br />
33333333333333333333333333333333333333333333333<br />
33333333333333333333333333333333333333333333333<br />
33333333333333333333333333333333333333333333333<br />
33333333333333333333333333333333333333333333333<br />
</p>
```

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

Program 6:

(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, Brach, 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.
(b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.

Basics:

XML:

- EXtensible Markup Language (XML) is a way to apply structure to a web page. XML provides a standard open format and mechanisms for structuring a document so that it can be exchanged and manipulated.
- Like HTML, XML uses tags to "mark-up" data content. Unlike HTML, in XML you define your own tags that meet the exact needs of your document. The custom tags make your data easier to organize and search.
- When creating an XML document, you must begin with the XML declaration statement. This statement alerts the browser or other processing tools that this document contains XML tags. The declaration looks like this:
  <?xml version="1.0"?>

XML Style Sheets:

- Extensible Stylesheet Language (XSL) is a language for expressing style sheets specifically for use with
- XML. It consists of two parts:
  - Extensible Stylesheet Language Transformer (XSLT)—a language for transforming XML documents. XSLT engines are typically used to convert XML into a strict form of HTML known as XHTML.
  - Formatting Objects (FO)—an XML vocabulary for specifying formatting semantics. Formatting Objects are not widely implemented at present (IE5.0 does not implement them).
- These two parts form the basis of the XSL style sheet that describes how the structured content of the XML data file should be displayed.

Solutions:

```
<!-- 6a.xml -->

<?xml-stylesheet type = "text/css" href = "6a.css" ?>
<students>
  <VTU>
    <USN> 1PI06ISIS001 </USN>
    <name> Amar </name>
    <college> PESIT </college>
    <branch> ISE</branch>
    <YOJ> 2006 </YOJ>
    <email> amar@gmail.com </email>
  </VTU>
  <VTU>
    <USN> 1PI06ISIS002</USN>
    <name> asha</name>
    <college> PESIT </college>
    <branch> ISE </branch>
    <YOJ> 2006 </YOJ>
    <email> asha@PESIT.in </email>
  </VTU>
  <VTU>
    <USN> 1PI06ISIS003 </USN>
    <name> Bhavya </name>
    <college> PESIT </college>
    <branch> ISE </branch>
    <YOJ> 2006</YOJ>
    <email> bhavya@yahoo.com </email>
  </VTU>
</students>

<!-- 6a.css -->


VTU {
font-family: "lucida calligraphy", arial, 'sans serif';
margin-left: 10pt;
background-color:rgb(200,255,205);
}

USN {
font-size: 85%;
background-color:rgb(100,255,105);
}
```

```
name {
color:blue;
font-size: 39pt;
font-style: italic;
font-weight: bold;
}

college {
color:pink;
font-size: 25pt;
font-style: italic;
font-weight: bold;
}

branch {
color:red;
font-size: 32pt;
font-style: italic;
}

YOJ
{
padding: 0.5cm;
}

email
{
color:green;
font-size: 25pt;
font-style: italic;
font-weight: bold;
}
```

<!-- 6b.xml -->

```
<?xml version = "1.0"?>
<?xml:stylesheet type = "text/xsl" href = "6b.xsl" ?>
<VTU>
   <USN> 1PI06ISE </USN>
   <name> Amar </name>
   <college> PESIT </college>
   <branch> ISE </branch>
   <YOJ> 2006 </YOJ>
   <email> amar@gmail.com </email>
</VTU>
```

<!-- **6b.xsl** -->

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

<xsl:template match = "VTU">
 <html><head><title> Style sheet for 6b.xml </title>
 </head><body>
 <h2> VTU Student Description </h2>
 <span style = "font-style: italic; color: blue;"> USN:
 </span>
 <xsl:value-of select = "USN" /> <br />
 <span style = "font-style: italic; color: blue;"> Name:
 </span>
 <xsl:value-of select = "name" /> <br />
 <span style = "font-style: italic; color: blue;"> College:
 </span>
 <xsl:value-of select = "college" /> <br />
 <span style = "font-style: italic; color: blue;"> Branch:
 </span>
 <xsl:value-of select = "branch" /> <br />
 <span style = "font-style: italic; color: blue;"> Year of Join:
 </span>
 <xsl:value-of select = "YOJ" /> <br />
 <span style = "font-style: italic; color: blue;"> E-Mail:
 </span>
 <xsl:value-of select = "email" /> <br />
 </body></html>
</xsl:template>
</xsl:stylesheet>
```

Program 7:
(a) Write a Perl program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.
(b) Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.

Basics:

- Perl: Practical Extraction and Report Language
- Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, games, bioinformatics, and GUI development.
- Symbols and their meanings:

$	a scalar
@	an array
%	a hash
none	a file handle
&	a subroutine (the & is optional in some contexts)
*	a typeglob

- Perl is widely favored for database applications. Its text-handling facilities are useful for generating SQL queries; arrays, hashes, and automatic memory management make it easy to collect and process the returned data.

Solutions:

# **7a.pl**

```
#!/usr/bin/perl
use CGI':standard';
print "content-type:text/html","\n\n";
print "<html>\n";
print "<head> <title> About this server </title> </head>\n";
print "<body><h1> About this server </h1>","\n";
print "<hr>";
print "Server name :",$ENV{'SERVER_NAME'},"<br>";
print "Running on port :",$ENV{'SERVER_PORT'},"<br>";
print "Server Software :",$ENV{'SERVER_SOFTWARE'},"<br>";
print "CGI-Revision :",$ENV{'GATEWAY_INTERFACE'},"<br>";
print "<hr>\n";
print "</body></html>\n";
```

```
exit(0);
```

```
<!-- 7b.html -->

<html>
<body>
<form action="http://localhost/6b.pl">
<input type="text" name="com">
<input type="submit" value="Submit">
</form>
</html>
```

```
# 7b.pl

#!/usr/bin/perl
use CGI':standard';
print "content type: text/html \n\n";
$c=param('com');
system($c);
```

Program 8:
(a) Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.
(b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

Solutions:

#**8a.pl**

```
#!/usr/bin/perl
use CGI ':standard';
use CGI::Carp qw(warningsToBrowser);
@coins = ("Welcome to Web Programming Lab","Have a nice time in lab", "Practice all the programs", "well
done Good Day");
$range = 4;
$random_number = int(rand($range));
if(param)
{
print header();
print start_html(-title=>"User Name",-bgcolor=>"Pink",-text=>"blue");
$cmd=param("name");
print b("Hello $cmd, $coins[$random_number]"),br();
print start_form();
print submit(-value=>"Back");
print end_form();
print end_html();
}
else
{
print header();
print start_html(-title=>"Enter user name",-bgcolor=>"yellow",text=>"blue");
print start_form(),textfield(-name=>"name",-value=>" "), submit(-name=>"submit",-value=>"Submit"),reset();
print end_form();
print end_html();
}
```

#**8b.pl**

```
#!/usr/bin/perl
use CGI ':standard';
use CGI::Carp qw(warningsToBrowser);

print header();
print start_html(-title=>"WebPage Counter",          -bgcolor=>"Pink",-text=>"blue");
```

```
open(FILE,'<count.txt');
$count=<FILE>;
close(FILE);

$count++;
open(FILE,'>count.txt');
print FILE "$count";
print b("This page has been viewed $count times");
close(FILE);
print end_html();
```

Program 9:
Write a Perl program to display a digital clock which displays the current time of the server.

Solutions:

```
#!/usr/bin/perl

use CGI ':standard';
print "Refresh: 1\n";
print "Content-Type: text/html\n\n";

print start_html(-title=>"Program 8",-bgcolor=>"Black",-text=>"white");

($s,$m,$h)=localtime(time);

print br,br,"The current system time is $h:$m:$s";
print br,br,hr,"In words $h hours $m minutes $s seconds";
print end_html;
```

Program 10:
Write a Perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.

Basics:

MySQL is a relational database management system (RDBMS that has more than 6 million installations. MySQL stands for "My Structured Query Language". The program runs as a server providing multi-user access to a number of databases. MySQL code uses C and C++. The SQL parser uses yacc and a home-brewed lexer, sql_lex.cc.

Solutions:

```
#! /usr/bin/perl

print "Content-type: text/html\n\n";
print "<HTML><TITLE>Result of the insert operation </TITLE>";
use CGI ':standard';
use DBI;
$dbh=DBI->connect("DBI:mysql:satish","root","ghalige");
$name=param("name");
$age=param("age");
$qh=$dbh->prepare("insert into stud values('$name','$age')");
$qh->execute();
$qh=$dbh->prepare("Select * from stud");
$qh->execute();
print "<table border size=1><tr><th>Name</th><th>Age</th></tr>";

while ( ($name,$age)=$qh->fetchrow())
{
    print "<tr><td>$name</td><td>$age</td></tr>";
}

print "</table>";
$qh->finish();
$dbh->disconnect();
print"</HTML>";

<html>
<body>
<form action="http://localhost/9.pl">
  Name : <input type="text" name="name"> <br>
  Age :<input type="text" name="age"> <br>
<input type="submit" value="Submit">
</form>
</html>
```

Program 11:
Write a PHP program to store current date-time in a COOKIE and display the ‘Last visited on’ date-time on the web page upon reopening of the same page.

Basics:
PHP, or PHP: Hypertext Preprocessor, is a widely used, general-purpose scripting language that was originally designed for web development, to produce dynamic web pages. It can be embedded into HTML and generally runs on a web server, which needs to be configured to process PHP code and create web page content from it. It can be deployed on most web servers and on almost every operating system and platform.
PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP parser compiles input to produce bytecode for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Solutions:

```
<?php
date_default_timezone_set('Asia/Calcutta');

//Calculate 60 days in the future
//seconds * minutes * hours * days + current time

$inTwoMonths = 60 * 60 * 24 * 60 + time();

setcookie('lastVisit', date("G:i - m/d/y"), $inTwoMonths);

if(isset($_COOKIE['lastVisit']))
        {
            $visit = $_COOKIE['lastVisit'];
            echo "Your last visit was - ". $visit;
        }
else
        echo "You've got some stale cookies!";
?>
```

Program 12:
Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

Solutions:

```
<?php
session_start();
session_register("count");

if (!isset($_SESSION))
{
        $_SESSION["count"] = 0;
        echo "<p>Counter initialized</p>\n";
}
else { $_SESSION["count"]++; }

echo "<p>The counter is now <b>$_SESSION[count]</b></p>".
        "<p>reload this page to increment</p>";
?>
```

Program 13:
Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.

Solutions:

```
<-- 13a.php -->

<html>
<body>
<?php
   $self = $_SERVER['PHP_SELF'];

   $dbh = mysql_connect('localhost', 'root', 'satish1') or
                                  die(mysql_error());
   mysql_select_db('satish') or die(mysql_error());

   if(isset($_POST['name']))
     {
      $nme = $_POST['name'];
      $ad1 = $_POST['add1'];
      $ad2 = $_POST['add2'];
           $eml = $_POST['email'];
      if($nme != "" && $ad1 != "")
      {
      $query = "INSERT INTO contact VALUES
                        ('$nme', '$ad1', '$ad2', '$eml')";
      $result = mysql_query($query) or die(mysql_error());
      }
      else
       echo "one of the field is empty";
     }
    mysql_close($dbh);

?>
<FORM ACTION="<?=$self?>" METHOD="POST">
  <P>
   Name: <INPUT TYPE=text NAME="name" value=""> <BR>
   Address 1:<INPUT TYPE=text NAME="add1" value=""><BR>
   Address 2:<INPUT TYPE=text NAME="add2" value=""><BR>
   email:<INPUT TYPE=text NAME="email" value=""><BR>
   <INPUT TYPE=submit>
</FORM>
</body>
</html>
```

```
<!-- 13b.html -->


<html>
<head><title>Program 13</title></head>
<body>
  <form action="13b.php" method="post">
     Enter Name of the person <input type="text" name="name">
     <input type=submit>
  </form>
</body>
</html>


<!-- 13b.php -->

<html>
<head><title>Search Result </title></head>
<body>
<h3>Search Result </h3>
<hr>
<?php
 $link=mysql_connect("localhost","root","satish1");
 mysql_select_db("satish");

 $n=$_POST["name"];
 print "Entered Name is $n \n";

 $var=mysql_query("SELECT * FROM contact WHERE name like '%$n%'");

 echo"<table border size=1>";
 echo"<tr><th>Name</th> <th>Address 1</th> <th>Address 2</th>
         <th>E-mail</th></tr>";
 while ($arr=mysql_fetch_row($var))
  {
   echo "<tr><td>$arr[0]</td> <td>$arr[1]</td> <td>$arr[2]</td>
        <td>$arr[3]</td> </tr>";
  }
  echo"</table>";
  mysql_free_result($var);
  mysql_close($link);
?>
<hr>
<form action="12b.html">
<input type="submit" value="Back">
</form>
</body>
```

```
</html>
```

Program 14:

Using PHP and MYSQL , develop a program to accept book Information viz, Accession number, title,authors,edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

```
<!-- 14a.html -->

<html>
<head><title>ENTER THE INFORMATION INTO DATABASE</title></head>
<body>
<form action=”http://localhost/14a.php” method=”post”>
<h1>ENTER THE BOOK DETAILS</h1>
ACCNO: <input type="text"name="Acc_no"> <br>
TITLE: <input type="text"name="Title"> <br>
AUTHOR:<input type="text"name="Author"> <br>
EDITION:<input type="text"name="Edition"> <br>
PUBLICATION:<input type="text"name="Publisher"> <br>
<input type="submit"value="SUBMIT">
<input type="reset"value="CLEAR">
</form>
</body>
</html>


<!-- 14a.php -->

<?php
header('content-type:text/plain');
$db=mysql_connect('localhost','root','') or die("Connection Failed");
mysql_select_db('librarys',$db);
mysql_query("insert into book
values('$_POST[Acc_no]','$_POST[Title]','$_POST[Author]','$_POST[Publisher]','$_POST[Edition]')") or
die("Insertion Failed");
echo "Record Insertion Successful";
?>

<!-- 14b.html -->

<html>
<head><title>Enter the Title of the book to be searched</title></head>
<body>
<form action =" http://localhost/14b.php" method =' POST'>
Enter the Title of the book to be searched<input type ="text" name="Title"><br>
<input type="submit" value="SUBMIT">
</form>
```

```
</body>
</html>

<!-- 14b.php -->

<?php
header("Content-type:text/plain");
$db=mysql_connect('localhost','root','') or die("connection failed");
mysql_select_db('student',$db);
$result=mysql_query("select * from library where title like '%".$_POST['Title']."%'",$db);
if(!$result)
{
	echo "Query failed";
}
$row=mysql_fetch_row($result);
if(empty($row))
{
	echo "No matching results";
	exit;
}
echo "Access No.\tTitle\t\tAuthor\t\tPublisher\tEdition\n";
do
{
	echo $row[0]."\t\t".$row[1]."\t".$row[2]."\t".$row[3]."\t".$row[4]."\n";
}while($row=mysql_fetch_row($result));

?>
```

WEB PROGRAMMING LAB VIVA

1. Expand HTML.
2. Explain Web Engineering?
3. What is the difference between width="100" and width="100%"?
4. What are meta tags and why it is used?

   Ans: Metadata is information about data.The tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.Meta elements are typically used to specify page description, keywords, author etc.
5. What is web 2.0?

   Ans: Web 2.0 is all about "Design Pattern" and "Business Model" for the next generation of the software
6. What is the difference between HTML and XHTML?
7. How do you make decision when to use use HTML & XHTML ?

   Ans: HTML and XHTML are very similar, but XHTML follows a stricter set of rules, making it easier to validate data and design pages ...
8. What is the format of document structure in HTML?
9. What is DIV in HTML?
10. What is Empty Elements in HTML?

    Ans: Empty Elements in HTML depicts that there is no text contained and EMPTY attributes are not permitted to have an end-tag. They are used simply as markers. ...
11. What is SPAN in HTML?

    Ans: The tag is used to group inline elements in aa document to format them with styles.p>This is another paragraph ...
12. What are HTML elements?

    Ans: HTML elements are nothing but HTML tags. eg: html, head, title, meta, body, table, h1, h2, h3, font, p, marquee etc ...
13. What is HEAD in HTML document?

    Ans: The head tag is placed above the body element in the html document. The head element contains general information, also called meta-information, about a document.
14. What is Document Type Definition?

    Ans: Document Type Defination (DTD) specifies the syntax of a web page,It is used to specify rules that apply to the markup of the document of a particular type,including a set of element and entity declarations.

15. What are differences between DIV and SPAN?

Ans: DIV is used to select a block of text so that one can apply styles to it. SPAN is used to select inline text and let users to apply styles to it.

16. What are tags used for the following and give details on the same?

- Ordered list
- Unordered list
- Table
- Span
- Image
- Paragraph
- Scrolling text
- Headings
- Frames

17. Give examples for nested tags.
18. Explain the various tags used with the <table> tag.
19. Explain the various tags with <form> tag.
20. How do you specify the size of the text?
21. What is meant by CSS?
22. How do you add an external CSS?
23. How do you define a CSS in the same HTML code?
24. What are the various attributes defined to a text in CSS?
25. How do you define a class in CSS?
26. What is meant by Javascript?
27. Which tag do we use insert a JavaScript in HTML page?
28. When is Javascript used?
29. What is the function of prompt?
30. How can you declare an array and string in Javascript?
31. Why do we use document.write function?
32. What is the significance of document?
33. What does alert function do?
34. What do you mean by regular expression?
35. Why is "$" appended at the end of the pattern?
36. What does indexof() do?

37. What is focus();
38. How do you compare the regular expression with the given string?
39. Expand XML.
40. Expand XSL.
41. Can CSS be used in place of XSL? Why?
42. What do you mean by XML?
43. Explain the structure of XML document.
44. Explain the function of Z-index.
45. What is Perl?
46. What does the $ENV command do?
47. What is the first line of any Perl program? What is its significance?
48. Why is “USE CGI ‘:standard’ used?
49. How are the parameters extracted from an HTML page by Perl program?
50. Why do we use “content type”?
51. Why do we use two “\n” in the content type? Isn’t one sufficient?
52. What do the symbols $ @ and % mean when prefixing a variable?
53. What are the two ways of sending data from HTML page to the server?
54. Differentiate between get and post method?
55. What is the pre requisite to use execute() in Perl?
56. Explain file operations in Perl.
57. Explain the process from the press of submit button to retrial of data from server using perl.
58. What is the difference between for & foreach, exec & system?
59. What is meant by php?
60. What is meant by a cookie?
61. What is a session?
62. How do you make a session secure?
63. Explain various SQL commands.
64. What is meant by a Query?
65. What does a special set of tags <?= and ?> do in PHP?
    Ans: Display content directly on browser.
66. How do you define a constant a constant in php?
    Ans: Via define() directive, like define ("MYCONSTANT", 100);

67. What is CGI?
68. What is php?
69. Name some sever side and client side scripting languages
70. What is a Web server?
71. What is a application server?
72. Give examples for web & application servers.
73. What is DBI? Explain
74. Explain ‘ print header’
75. Explain ‘ Refresh ‘ statement in html
76. Explain pre tag, frames tag and a href tab
77. How web works?
78. compare programming language and scripting language

****

ALL THE BEST