

# **PESIT BANGALORE SOUTH CAMPUS**

(Formerly PES School of Engineering)

Hosur Road, 1KM before Electronic City, Bangalore-560 100

**Department of Computer Science & Engineering**

Academic Year 2014-2015 (Odd Semester)

SESSION: JULY 2014 – DECEMBER 2014

**VII SEMESTER**

# **WEB PROGRAMMING LAB MANUAL**

**SUBJECT CODE: 10CSL78**

**Faculty: Mr. Sajeevan K ,  
Mr. Jnanamurthy  
&**

**Mr. Hanumant Pujar**

**Note:** Student is required to solve one problem in the examination. The questions are allotted based on lots.

### Experiment 1:

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

**Objective:** - To learn how to create a simple web page using html along with the usage of style sheets, lists, creation or tables with borders, padding and colors.

### Procedure:-

1. Open a new file using Notepad.
2. Rename the file as pgm1.html (Extension for html programs is .html)
3. Create a file (external cascading style sheet) by name Lab1.css where the font, size, colors etc.. are specified. The style type should be set to “text/css” under the <head> tag.
4. The pgm1.css file should be included in pgm1.html file in the <link href ....> tag
5. In the <body> tag of the html document. Create an Ordered list using <ol> tag. Also create sublists under the main list as shown below:
  1. Dept of CSE
    - i. I Sem
    - ii. III Sem
    - iii. V Sem
    - iv. VII Sem
  2. Dept of ISE
    - i. I Sem
    - ii. III Sem
    - iii. V Sem
    - iv. VII Sem
6. Create a Table which shows the Room No of the class/section and the strength of the students in each class/section of the departments, CSE and ISE.
7. Fill the table using an appropriate colour and specify the thickness of the border for the table.
8. Make the website neat and attractive with relevant text and pictures
9. Use suitable tags wherever necessary.

/\*-----Basics.css-----\*/

```
body { background-color:white; }
table { border: 2px solid blue; padding: 0px; }
th { border: 1px solid blue; padding:8px; }
td { border: 1px solid blue; padding:8px; }
ol { font-size:80%; list-style:lower-roman; }
span { background-color:#grey; font-size:120%; font-style:italic; }
p { font-size:90%; }
```

/\* End of File \*/

<!--

-----XHTML CODE-----

-->

```
<?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="Basics.css" />
<title> CSS Basics </title>
</head>
```

```
<body>
<h1> This header is 36 pt </h1>
<h2> This header is black </h2>
<p> This paragraph has a left margin of 50 pixels </p>
```

```
<table>
<tr>
<th> Name </th>
<th> Email </th>
<th> Phone number </th>
</tr>
```

```
<tr>
<td> Xyz </td>
<td> xyz@abc.com </td>
<td> 12345 </td>
</tr>
```

```
<tr>
<td> Qwerty </td>
<td> qwerty@poi.com </td>
<td> 987654 </td>
</tr>
```

```
<tr>
<td> Zaqwsx </td>
<td> zaqwsx@mnbv.co.uk </td>
<td> 78563 </td>
</tr>
</table>
```

```
<hr />
This is an ordered list.
<ol>
  <li>ISE </li>
  <li>CSE </li>
  <li>ECE </li>
</ol>

<p>
  <span>This is some span text.</span>
  This is some random text.
  <span>This is some more span text.</span>
</p>
</body>
</html>
```



```
<!-- End of file -->
```

PES  
INSTITUTIONS  
CSE

## Experiment 2 :

Develop and demonstrate a XHTML file that includes Javascript script for the following problems:

- Input : A number n obtained using prompt  
Output : The first n Fibonacci numbers
- Input : A number n obtained using prompt  
Output : A table of numbers from 1 to n and their squares using alert

### Objective :-

To get acquainted with javascript and how to embed javascript in html code.

### Procedure :-

#### Question 2a :-

1. Declare the script tag as text/javascript in the beginning of the <body> of html program
2. Get the number of Fibonacci elements to be generated from the user using prompt()
3. Validate input given and alert the user for invalid input using alert()
4. Generate the Fibonacci numbers using the standard algorithm and print it to std out using document.write()

```

<!--
-----XHTML CODE-----
-->
<?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>
<title> Fibonacci Numbers </title>
</head>
<body>
<h1>Calculating the fibonacci numbers</h1>
<script type="text/javascript">

var n,a=0,b=1,i,c
n=prompt("Enter a number ","")
if(n<=0) alert("Invalid number")
else
{
if(n==1) document.write(a)
else document.write(a+"<br />" +b)
for(i=2;i<n;i++)
{
c=a+b
a=b
b=c
document.write("<br />" +c)
}
}
}
</script>

</body>
</html>

```

<!-- End of File -->

**Question 2b :-**

1. Declare the script tag as text/javascript in the beginning of the <body> of html program
2. Get the number till which the squares have to be generated using prompt()
3. Validate input given and alert the user for invalid input using alert()
4. For all the numbers from 1 to n, generate their squares and print them.

```

<!--
-----XHTML CODE-----
-->

<?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> <title>Squares of numbers</title> </head>

  <body>

    <h1>
      Printing numbers & calculating their squares
    </h1>
    <script type="text/javascript">

      var n,i;
      n=prompt("Enter a number","");
      if(n>0)
      {
        c="Number | Square"
        for(i=1;i<=n;i++) c=(c+"\n"+i+" ----> "+i*i);
        alert(c)
      }
      else alert("Enter a number greater than 1.")

    </script>
  </body>
</html>

```

<!--End of File -->

### Experiment 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

#### Objective :-

To get acquainted with javascript procedures and usage of regular expressions in javascript.

#### Procedure :-

#### Question 3a :-

1. Declare the script tag as text/javascript in the beginning of the <body> of html program
2. Get the string from the user using prompt()
3. Validate input string (should be only alphabets, a - z) using the regular expression “ /^[a-zA-Z]+\$/ “ and alert the user for invalid input using alert()
4. Convert the string to lowercase using toLowerCase()
5. Use indexOf(<vowel>) for each vowel to get the position of the vowel in the string.
6. Print the leftmost vowel i.e., print the lowest index got from indexOf()

```
<!--
-----XHTML CODE-----
-->
```

```
<?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> <title> Left most vowel </title> </head>
```

```
<body>
```

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

```
function disp(str)
{
    var reg=/^[a-zA-Z]+$/
    if(!str.value.match(reg))
    {
        alert("Enter alphabets only!")
        return false
    }
    var i,b
    b=str.value
    for(i=0;i<str.value.length;i++)
    {
        switch(b[i])
        {
            case "a":
```

```

        case "A":
        case "e":
        case "E":
        case "i":
        case "I":
        case "o":
        case "O":
        case "u":
        case "U":alert( "\"" +b[i]+"\" is the first vowel found in postion "+(i+1));
        exit(0);
        default: break;
    }
    }
    if(i>=str.value.length) alert("No vowels found. :)")
}
</script>
<form action="">
  <h2> Finding the left most Vowel </h2>
  <p>
    String: <input type="text" name="str" />
    <input type="button" value="Find" onclick="disp(str)" />
  </p>
</form>
</body>
</html>

<!-- End of File -->

```

**Question 3b :-**

1. Declare the script tag as text/javascript in the beginning of the <body> of html program
2. Get the number to be reversed from the user using prompt()
3. Validate input number (should be a positive number between 0 to 9) using the regular expression “ /^[0-9]+\$/ “ and alert the user for invalid input using alert()
4. Reverse the number using modulus operation.
5. Use math.floor(number/10) to get the floor of number after division (used for reversing)
6. Display the reversed string using alert()

```

<!--
-----XHTML CODE-----
-->
<?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> <title>Number reversal</title> </head>
<body>
  <h1>Printing digits in reverse order</h1>
  <script type="text/javascript">

    function disp(str)
    {

```



```
var n=0,r=0,res=0
n=str.value
if( n>0 || n==0 )
{
  while(n!=0)
  {
    r=n%10
    n=Math.floor(n/10)
    res=res*10+r
  }
  alert("The reverse of the number is: "+res)
}
else alert("Enter a valid number")
}

</script>
<form>
  <p>
    Number: <input type="text" name="str" />
    <input type="button" value="Find" onclick="disp(str)" />
  </p>
</form>
</body>
</html>
```

<!-- End of file -->

INSTITUTIONS

CSE

## Experiment 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)

### Objective :-

To write functions in html, validate using regular expressions.

### Procedure :-

#### Question 4a :-

1. Declare the script tag as text/javascript in the beginning of the <body> of html program
2. Get the USN from the user using prompt()
3. Validate USN input by the user using the regular expression : /^[1-4][A-Z][A-Z][0-9][0-9][A-Z][A-Z][0-9][0-9][0-9]\$/
4. If the USN format is correct, alert the user with a message, else alert the user with the format DAADDAADDD (for eg: 1PE09CS001)

```

<!--
-----XHTML CODE-----
-->

<?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> <title> USN Validation </title> </head>
<body>
  <script type="text/javascript">
function func(usn)
{
var pattern1=/^[1-4][A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{3}$/
if(!usn.value.match(pattern1)||usn.value.length==0)
{
    alert("Invalid USN!\nEnter a valid USN")
    return false
}
else alert("USN valid!")
}
</script>
<form action="">
  <p>
    USN: <input type="text" name="usn" /> <br/>
    <input type="button" value="Validate" onclick="func(usn)" />
  </p>
</form>
</body>
</html>

<!-- End of file -->
    
```

**Procedure :-**

**Question 4b :-**

1. Declare the script tag as text/javascript in the beginning of the <body> of html program
2. Get the USN and Semester from the user using prompt()
3. Validate USN input by the user using the regular expression : /^[1-4][A-Z][A-Z][0-9][0-9][A-Z][A-Z][0-9][0-9][0-9]\$/
4. If the USN format is correct, alert the user with a message, else alert the user with the format DAADDAADDD (for eg: 1PE09CS001)
5. Validate Semester by the user using the regular expression : /^[1-8]\$/
6. If the Semester is correct, alert the user with a message, else alert the user with the valid semester number (1 – 8).

```

<!--
-----XHTML CODE-----
-->

<?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> <title>USN and Semester Validation</title> </head>
<body>
<script type="text/javascript">
function disp(usn,sem)
{
var pattern1=/^[1-4][A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{3}$/
if(!usn.value.match(pattern1)||usn.value.length==0)
{
alert("Invalid USN!\nEnter a valid USN")
return false
}
else alert("USN valid!") var pattern2=/^[1-8]$/
if(!sem.value.match(pattern2)||sem.value.length==0)
{
alert("Invalid Semester!\nEnter a valid Semester")
return false
}
else alert("Semester valid!")
}
</script>
<form action="">
<p>
USN: <input type="text" name="usn" /> <br/>
Semester: <input type="text" name="sem" /> <br/>
<input type="button" value="Validate" onclick="disp(usn,sem)" />
</p>
</form>
</body>
</html>

<!-- End of file -->

```

### Experiment 5 :

a) Develop and demonstrate, using JavaScript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.

```

<!--
-----XHTML CODE-----
-->
<?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>
  <title>Stack</title>
  <script type="text/javascript">
    var top='p3'
    function toTop(newTop)
    {
      domTop=document.getElementById(top).style
      domNew=document.getElementById(newTop).style
      domTop.zIndex="0"
      domNew.zIndex="10"
      top=newTop
    }
  </script>
  <style type="text/css">
    .para1{position:absolute;top:10;left:120;z-index:0;
      border:solid;padding:80; width:300;background-color:aqua;}
    .para2{position:absolute;top:50;left:150;z-index:0;
      border:solid;padding:80; width:300;background-color:yellow; }
    .para3{position:absolute;top:100;left:180;z-index:0;
      border:solid;padding:80; width:300;background-color:red; }
  </style>
</head>

<body>
  <p class="para1" id="p1" onmouseover="toTop('p1')"> Frame One </p>
  <p class="para2" id="p2" onmouseover="toTop('p2')"> Frame Three </p>
  <p class="para3" id="p3" onmouseover="toTop('p3')"> Frame Two </p>
</body>
</html>
<!-- End of file -->

```

b) Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom

```

<!--
-----XHTML CODE-----
-->
<?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>
    <title>Stack</title>
    <script type="text/javascript">
      var top='p3'
      function toTop(newTop)
      { domTop=document.getElementById(top).style
        domNew=document.getElementById(newTop).style
        domTop.zIndex="0"
        domNew.zIndex="10"
          top=newTop
        }
    </script>
    <style type="text/css">

    .para1{position:absolute;top:10;left:120;z-index:0;
      border:solid;padding:80; width:300;background-color:aqua;}
    .para2{position:absolute;top:50;left:150;z-index:0;
      border:solid;padding:80; width:300;background-color:yellow; }
    .para3{position:absolute;top:100;left:180;z-index:0;
      border:solid;padding:80; width:300;background-color:red; }

    </style>
  </head>

  <body>
    <p class="para1" id="p1" onmouseover="toTop('p1')
      onmouseout="toTop('p2')"> Frame One </p>
    <p class="para3" id="p3" onmouseover="toTop('p3')
      onmouseout="toTop('p2')"> Frame Two </p>
    <p class="para2" id="p2" onmouseover="toTop('p2')
      onmouseout="toTop('p2')"> Frame Three </p>
  </body>
</html>
<!-- End of file -->

```

## Experiment 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, 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.

```

/* -----Info.css----- */

stud-info { display:block; color:blue; font-style:italic; font-size:200%; }
student { display:block; font-size:100%; }
stud1 { display:block; color:blue; }
stud2 { display:block; color:red; }
stud3 { display:block; color:black; } usn,name,nov,branch,you,did { display:block; }

/* End of File */

<!--
-----XML File-----
-->
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="Info.css" type="text/css"?>

<student>
<stud-info>Student Information</stud-info>
<stud1>
<usn>USN: 1PEo8CS009</usn>
<name>Name: Aashish S</name>
<noc>College: PES School of Engineering</noc>
<branch>Branch: Computer Science and Engineering</branch>
<yoy>Year: 2012</yoy>
<eid>Email: aashishh@me.com</eid>
</stud1>
<br/>
<stud2>
<usn>USN: 1PEo8CS001</usn>
<name>Name: Anish L R</name>
<noc>College: PES School of Engineering</noc>
<branch>Branch: Computer Science and Engineering</branch>
<yoy>Year: 2012</yoy>
<eid>Email: anishh@me.com</eid>
</stud2>
<br/>
<stud3>
<usn>USN: 1PEo8CS006</usn>
<name>Name: Amal Antony</name>
<noc>College: PES School of Engineering</noc>
<branch>Branch: Computer Science and Engineering</branch>
<yoy>Year: 2012</yoy>
<eid>Email: amal@me.com</eid>
</stud3> </student>

<!-- End of file -->
    
```

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

```
<!--  
-----XML File-----  
-->  
<?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl" href="6B - XSLT Student Info.xsl"?>  
  
<VTU>  
  <USN> 1PEo8CS020 </USN>  
  <name> John Doe </name>  
  <college> PES School of Engineering </college>  
  <branch> CSE </branch>  
  <YOJ> 2008 </YOJ>  
  <email> John@me.com </email>  
</VTU>  
  
<!-- End of file -->
```



## Experiment 7 :

a) Write a Perl program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.

```

<!--
-----XHTML File-----
-->

<?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>
<title>Server Information</title>
</head>
<body>
<form action="/cgi-bin/7A - Server Info.cgi" method="post">
<p>
Server Information Display Program
<input type="submit" value="Click here" />
</p> </form>
</body>
</html>

<!-- End of file -->
-----Perl Script-----

#!/usr/bin/perl

# Question 7A
# Write a Perl program to display various Server Information like Server Name, Server
# Software, Server protocol, CGI Revision etc.

use CGI:'standard';
print "content-type:text/html \n\n";
print "Server Name: $ENV{'SERVER_NAME'} <br/>";
print "Server Port: $ENV{'SERVER_PORT'} <br/>";
print "Server Software: $ENV{'SERVER_SOFTWARE'} <br/>"; print "Server Protocol:
$ENV{'SERVER_PROTOCOL'} <br/>"; print "CGI VERSION: $ENV{'GATEWAY_INTERFACE'} <br />";

# -----End of Perl Script-----
    
```



b) Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.

```

<!--
-----XHTML File-----
-->

<?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>
    <title>UNIX Command</title>
  </head>
  <body>
    <form action="/cgi-bin/7B - UNIX Commands.cgi" method="post">
      <p>
        Command: <input type="text" name="cmd" />
        <input type="submit" value="Execute" />
      </p> </form>
    </body>
  </html>
  <!-- End of file -->

```

```

-----Perl Script-----

#!/usr/bin/perl

# Question 7B
# Write a Perl program to accept UNIX command from a HTML form and to display the output
# of the command executed.

use CGI:standard;
print "content-type:text/html\n\n";
$c=param('cmd');
system($c);
exit(0);

# -----End of Perl Script-----

```

### Experiment 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.

```

<!--
-----XHTML File-----
-->

<?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> <title>Accept user name</title> </head>
<body>
<form action ="/cgi-bin/8A - Greetings.cgi" method="post">
<p> Username: <input type="text" name="username" />
<input type="submit" value="OK" /> </p>
</form>
</body>
</html>

<!-- End of file -->
-----Perl Script-----

#!/usr/bin/perl

# Question 8A
# Write a Perl program to accept the User Name and display a greeting message
# randomly chosen from a list of 4 greeting messages.

use CGI::standard;
print "content-type:text/html\n\n";
$input=param("username");
my @msgs=("Good morning","Welcome","How are you doing?","Hello!");
$i=int rand scalar @msgs;
print "Hi, $input.<br>Message: ", $msgs [$i];

# -----End of Perl Script-----

```

- 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.

```

<!--
-----XHTML File-----
-->

<?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> <title>Counting No. Of Visits to this Page</title> </head>
<body>
<form action="/cgi-bin/8B - View Count.cgi" method="post">
<p> Number of Visits to this page
<input type="submit" value="Click Here" /> </p>

```

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

```
<!-- End of file -->
```

-----Perl Script-----

```
#!/usr/bin/perl
```

```
# Question 8B
```

```
# 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.
```

```
use CGI 'standard';
```

```
print "content-type:text/html \n\n";
```

```
# Requires a file 'count.dat' to pre-exist with the content '0'
```

```
open FILE, "<count.dat";
```

```
my $count = <FILE>;
```

```
close(FILE);
```

```
$count++;
```

```
open Handler, ">count.dat";
```

```
print Handler $count ;
```

```
close Handler;
```

```
open FILE, "<count.dat";
```

```
my $count2 = <FILE>;
```

```
close(FILE);
```

```
print b("This page has been viewed $count times");
```

```
# -----End of Perl Script-----
```

INSTITUTIONS

CSE

### Experiment 9 :

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

```

<!--
-----XHTML File-----
-->

<?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>
<title>Digital Clock Display Program</title>
</head>

<body>
<form action="/cgi-bin/9 - Digital Clock.cgi" method="post">
<p>
Digital Clock <input type="submit" value="Click Here" />
</p>
</form>
</body>
</html>

<!-- End of file -->
-----Perl Script-----

#!/usr/bin/perl

# Question 9
# Implementing a Digital Clock

use CGI:'standard';
print "refresh:1 \n"; ($s,$m,$h)=localtime(time);
print "content-type:text/html \n\n";
print "The system time is $h:$m:$s", "<br>"; print "In words $h hours $m minutes $s seconds";

# -----End of Perl Script-----

```

## Experiment 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.

```

<!--
-----XHTML File-----
-->

<?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>
    <title>Perl Database Interfaces</title>
  </head>
  <body>
    <form action="/cgi-bin/10 - Perl DBI.cgi" method="post">
      <p>
        <b>Enter Your Information</b> <br />
        Name: <input type="text" name="name" /> <br />
        Age: <input type="text" name="age" /> <br />
        <input type="submit" value="Add" />
        <input type="reset" value="Clear" />
      </p> </form>
    </body>
  </html>

  <!-- End of HTML file -->
-----Perl Script-----

```

```

#!/usr/bin/perl

# Question 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.

use CGI:standard;
print "content-type:text/html\n\n";

#use lib '/Applications/XAMPP/xamppfiles/lib/perl5/site_perl/5.10.1/darwin-2level';
use DBI;

$dbh = DBI->connect("DBI:mysql:Temp","root");
$name=param("name");
$age=param("age");
$sql="insert into Students values ('$name','$age)";
$sth=$dbh->prepare("$sql");
$sth->execute;
$sql = "select * from Students";
$sth = $dbh->prepare($sql);
$sth->execute;

print "<table border size=1>

```

```
<tr>
<th>Name</th>
<th>Age</th>
</tr>
";

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

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

# -----End of Perl Script-----
```



**PES**  
**INSTITUTIONS**  
**CSE**

### Experiment 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.

```

<!--
-----XHTML File-----
-->

<?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> <title>Cookies</title> </head>
<body>
<form action="Cookies.php" method="post">
<p>
    The last visited time was <input type="submit" name="Display Now"/>
</p>
</form>
</body>
</html>

<!-- End of file -->
<!--

```

Output:

You've got some stale cookies!  
Your last visit was - 18:54 - 09/29/11

```

-->
<?php
/*
----- Cookies.php (PHP File) -----
*/

$inTwoMonths=60*60*24*60+time();
setcookie('lastVisit',date("G:i - m/d/y"),$inTwoMonths);
if(isset($_COOKIE['lastVisit']))
{
    $visit=$_COOKIE['lastVisit'];
    echo "Your last visit was - ".$visit;
}

else echo "You've got some stale cookies!";
?>

```

### Experiment 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.

```

<!--
-----XHTML File-----
-->

<?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> <title>SESSION PROGRAM </title> </head>
  <body>
    <form action="Sessions.php" method="post">
      <p>
        To see page views count in session <input type="submit" name="Click Here"/>
      </p>
    </form>
  </body>
</html>

  <!-- End of file -->

<?php
/*
-----Sessions.php(PHP File)-----
*/

session_start();
if(isset($_SESSION['views']))
    $_SESSION['views']=$_SESSION['views']+1;
else
    $_SESSION['views']=1;

echo "This page has been viewed " .$_SESSION['views']. " times.";

?>

  <!-- End of file -->

```



### Experiment 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.

```

<!--
-----XHTML File-----
-->

<?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">
<body>
<?php
$self = $_SERVER['PHP_SELF'];
$dbh = mysql_connect('localhost', 'root', 'root') or
die(mysql_error());
mysql_select_db('satish') or die(mysql_error());
if(isset($_POST['name']))
{
$name = $_POST['name'];
$ad1 = $_POST['add1'];
$ad2 = $_POST['add2'];
$eml = $_POST['email'];
if($name != "" && $ad1 != "")
{
$query = "INSERT INTO contact VALUES
('$name', '$ad1', '$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>

<!-- End of file -->

```

## Experiment 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.

```

<!--
-----XHTML File-----
-->

<?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> <title>Personal Details DB</title> </head>
<body>
<form action="Database_Store.php" method="post">
<p>
<b>Add Entry</b> <br/>
ACCESSION NO: <input type="text" name="accno" /> <br />
TITLE: <input type="text" name="title" /> <br />
AUTHOR: <input type="text" name="author" /> <br />
Edition: <input type="text" name="edn" /> <br />
Publisher: <input type="text" name="pub" /> <br />
<input type="submit" value="Add Details" />
</p>
</form>
<hr/>
<form action="Database_Retrieve.php" method="post">
<p>
<b>Search Book</b> <br/>
Title: <input type="text" name="title" />
<input type="submit" value="Search" />
</p>
</form>
</body>
</html>

<!-- End of file -->

/*
----- Database_Store.php -----
*/

<?php

$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("Temp", $con);
$sql="INSERT INTO Temp.Books
VALUES('$ _POST[accno]','$ _POST[title]','$ _POST[author]','$ _POST[edn]','$ _POST[pub]')";
if (!mysql_query($sql,$con))

```

```

{
    die('Error: ' . mysql_error());
}
echo "1 Record Added";
mysql_close($con);
?>

<!-- End of File -->

----- Database_Retrieve.php -----
/*
*/

<?php

$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("Temp", $con);
$sql="SELECT * FROM Temp.Books WHERE Title='$_POST[title]'";
$result=mysql_query($sql,$con);

echo "<table border='1'>
<tr>
<th>Access No</th>
<th>Title</th>
<th>Author</th>
<th>Edition</th>
<th>Publication</th>
</tr>";
while($row = mysql_fetch_array($result))
{
    echo "<tr><td>" . $row['AccessionNo'] . "</td>";
    echo "<td>" . $row['Title'] . "</td>";
    echo "<td>" . $row['Author'] . "</td>";
    echo "<td>" . $row['Edition'] . "</td>";
    echo "<td>" . $row['Publisher'] . "</td></tr>";
}
echo "</table>";
mysql_close($con);
?>

<!-- End of file -->

```