DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

WEB PROGRAMMING LAB MANUAL-10CSL78
VII SEMESTER
2016-2017

Prepared by: Suparna K,
Asst. Professor
Dept. of CSE,
GCEM

Reviewed by: N S Saradha Devi
Head of the Department
Dept. of CSE
GCEM

Approved by: Dr. A A Powly Thomas
Principal
GCEM

181/1, 182/1, Hoodi Village, Sonnenahalli, K.R. Puram, Bengaluru, Karnataka 560048
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>S. No</th>
<th>Title of Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>From</td>
</tr>
<tr>
<td>1</td>
<td>Syllabus</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Course Objective and Course Outcome</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Do’s And Don’ts</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>List of Experiments</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Programs and Sample output</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>Viva Questions</td>
<td>62</td>
</tr>
</tbody>
</table>
SYLLABUS

Web Programming Laboratory

Subject Code: 10CSL78
I.A. Marks : 25
Hours/Week : 03
Exam Hours: 03
Total Hours : 42
Exam Marks: 50

1. 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

2. 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)

3. 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.
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.

4. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include 100 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.
b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.

5. 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.

6. 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.

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

8. 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.

9. 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.

10. 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.

11. 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.

12. Build a Rails application 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.

Note: In the examination each student picks one question from the lot of all 12 questions.
COURSE OBJECTIVE

The course will enable students to

- Understand the importance of the web as a medium of communication.
- Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.
- Develop skills in analyzing the usability of a web site.
- Learn the language of the web: HTML and CSS.
- Be able to embed social media content into web pages.
- Implement and understand how to interpret basic web analytics.
- Use JavaScript to access and use web services for dynamic content.

COURSE OUTCOME

After studying this course, students will be able to:

- Use Javascript and XHTML to create web pages with advanced interactivity
- Program basic functions in Javascript and XHTML
- Use javascript to create functional forms
- Use Javascript to control browser frames and windows
- Use cascading style sheets to design web pages
DO'S AND DON’TS

Do’s

Do wear ID card and follow dress code.

- Do log off the computers when you finish.
- Do ask for assistance if you need help.
- Do keep your voice low when speaking to others in the LAB.
- Do ask for assistance in downloading any software.
- Do make suggestions as to how we can improve the LAB.
- In case of any hardware related problem, ask LAB in charge for solution.
- If you are the last one leaving the LAB, make sure that the staff in charge of the LAB is informed to close the LAB.
- Be on time to LAB sessions.
- Do keep the LAB as clean as possible.

Don’ts

- Do not use mobile phone inside the lab.
- Don’t do anything that can make the LAB dirty (like eating, throwing waste papers etc).
- Do not carry any external devices without permission.
- Don’t move the chairs of the LAB.
- Don’t interchange any part of one computer with another.
- Don’t leave the computers of the LAB turned on while leaving the LAB.
- Do not install or download any software or modify or delete any system files on any lab computers.
- Do not damage, remove, or disconnect any labels, parts, cables, or equipment.
- Don’t attempt to bypass the computer security system.
- Do not read or modify other user’s file.
- If you leave the lab, do not leave your personal belongings unattended. We are not responsible for any theft.
# LIST OF EXPERIMENTS

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Title</th>
<th>Page no.</th>
</tr>
</thead>
</table>
| 1.    | 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` | 10-15    |
| 2.    | 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) | 16-21    |
| 3.    | 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.  
|       | 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. | 22-26    |
| 4.    | a) 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, 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.  
<p>|       | b) Create an XSLT style sheet for one student element of the above document and | 27-31    |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 5. | 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. | 32-35 |
| 6. | 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. | 36-38 |
| 7. | Write a Perl program to display a digital clock which displays the current time of the server. | 39 |
| 8. | 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. | 40-42 |
| 9. | 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. | 43-44 |
| 10. | 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. | 45-46 |
| 11. | 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. | 47-52 |
| 12. | Build a Rails application 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. | 62 |
EXPERIMENT NO. 1(a)

AIM: To display Fibonacci series using Javascript

PROGRAM:

<!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>
<title>Fibonacci Series</title>
</head>
<body>
<script type="text/javascript">
var fib1=0,fib2=1,fib=0;
var num=prompt("Enter a number : \n", "");
if(num != null && num > 0 )
{
    document.write("<h1>The first "+num+" numbers in the fibonacci series </h1>" );
    if(num==1)
        document.write("<h2> " + fib1 + "</h2>" );
    else
    {
        document.write("<h2>" + fib1 + "</h2>" );
        document.write("<h2>" + fib2 + "</h2>" );
    }
    for(i=3;i<=num; i++)

}
```javascript
{  
    fib = fib1 + fib2;
    document.write("<h2> " + fib + "</h2>");
    fib1 = fib2;
    fib2 = fib;
}

else
    alert("Invalid Input");

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

SAMPLE OUTPUT

![Sample Output Image]

Enter a number: 5

OK  Cancel
The first 5 numbers in the Fibonacci series

0
1
1
2
3
EXPERIMENT NO. 1(b)

PROGRAM:
AIM: To display the square of a given numbers using Javascript

```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>
    <title>Number and its squares</title>
  </head>
  <body>
    <script type="text/javascript">
      var num = prompt("Enter a number : \n", ");
      var msgstr;

      if(num > 0 && num !== null){
        msgstr="Number and its Squares are \n";
        for(i=1;i <= num; i++)
        {
          msgstr = msgstr + i + " ^ 2 = " + i*i + 
"\n";
        }
        alert(msgstr);
      }
    else
```
alert("Invalid Input");

</script>

</body>

</html>

Note:
The \u00B2 character displays superscript 2 in the javascript boxes.

Code:

msgstr = msgstr + i + "\u00B2 = " + i*i + "n";

SAMPLE OUTPUT
Number and its Squares are:
1\times 2 = 1
2\times 2 = 4
3\times 2 = 9
4\times 2 = 16
5\times 2 = 25

[Prevent this page from creating additional dialogs] OK
EXPERIMENT NO. 2(a)

AIM: To validate the USN of the student and to print them

PROGRAM:

```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>
        <title> USN validator </title>
        <script type="text/javascript">
            function formValidator()
            {
                var usn = document.getElementById('usnFrm');

                usnExp=/[1-4][A-Z][A-Z]\d{2}[A-Z][A-Z]\d{3}$/

                if(usn.value.length==0)
                {
                    alert("USN is empty.");
                    usn.focus();
                    return false;
                }
                else if(!usn.value.match(usnExp))
                {
                    alert("USN should be in VTU USN format, eg., 1GD10CS001");
                }
            }
        </script>
    </head>
    <body>
        <form id="usnFrm">
            <input type="text" name="usn" placeholder="Enter USN">
            <button type="button" onclick="formValidator()">Validate</button>
        </form>
    </body>
</html>
```
usn.focus();
return false;
}

alert("USN: "+usn.value+" is in correct format");
return true;
}
</script>
</head>
<body>
<form onSubmit = "formValidator()">
Enter your VTU USN: <input type="text" id="usnFrm"/>
<br/>
<input type ="submit" value="SUBMIT"/>
</form>
</body>
</html>

SAMPLE OUTPUT
Form validation using javascript

Enter your VTU USN: ____________________________

Submit

USN is empty

OK

Form validation using javascript

Enter your VTU USN: 1M0

Submit

USN should be in VTU USN format, eg. 1M01325801

OK
EXPERIMENT No.2(b)

PROGRAM: To display the current semester

```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>
<title> USN validator </title>
<script type="text/javascript">
function formValidator()
{
    var usn = document.getElementById('usnFrm');
    var sem = document.getElementById('semFrm');

    usnExp=/^[1-4][A-Z][A-Z]\d{2}[A-Z][A-Z]\d{3}$/
    semExp=/^[1-8]$/

    if(usn.value.length==0)
    {
        alert("USN is empty.");
        usn.focus();
        return false;
    }
    else if(!usn.value.match(usnExp))
```
alert("USN should be in VTU USN format, eg., 1GD10CS001");
usn.focus();
return false;
}
else if(sem.value.length==0)
{
    alert("Semester field is empty.");
    sem.focus();
    return false;
}
else if(!sem.value.match(semExp))
{
    alert("Semester number should be from 1 to 8");
    sem.focus();
    return false;
}
alert("USN: "+usn.value+"\nSemester: "+sem.value);
return true;

</script>
</head>
<body>
<h1>Form validation using javascript</h1>
<form onSubmit = "formValidator()">
    <p>Enter your VTU USN : <input type="text" id="usnFrm"/></p>
    <p>Enter your current semester : <input type="text" id="semFrm"/></p>
    </form>
</body>
</html>

SAMPLE OUTPUT
Form validation using javascript

Enter your VTU USN : 1MJ12CS001
Enter your current semester :

Semester field is empty

OK

Form validation using javascript

Enter your VTU USN : 1MJ12CS001
Enter your current semester : 9

Semester number should be from 1 to 8

OK

Form validation using javascript

Enter your VTU USN : 1MJ12CS001
Enter your current semester : 1

USN 1MJ12CS001
Semester 1

OK
**EXPERIMENT No.3(a)**

**AIM:** To display the stacking of elements

```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>
    <title>Paragraph Stacking</title>
    <style type="text/css">
    .para {
        border: solid thin black;
        padding:1cm;
        position:absolute;
        width:300px;
    }
    #layer1 {
        background-color:yellow;
        top:200px; left:400px;
        z-index:1;
    }
    #layer2 {
        background-color:red;
        position:absolute;
        top:220px; left:420px;
        z-index:2;
    }
    #layer3 {
        background-color:blue;
        top:240px; left:440px;
        z-index:3;
    }
    </style>
    <script type="text/javascript">
    var topLayer = "layer3";
    function mover(toTop) {
        var oldTop = document.getElementById(topLayer).style;
        var newTop = document.getElementById(toTop).style;
        oldTop.zIndex = 0;
        newTop.zIndex = 5;
    }
    </script>
</head>
<body>
    <div class="para" id="layer3" onclick="mover('layer1')"></div>
    <div class="para" id="layer1" onclick="mover('layer2')"></div>
    <div class="para" id="layer2" onclick="mover('layer3')"></div>
</body>
</html>
```
Visibility of stacked paragraphs using Javascript

SAMPLE OUTPUT

Visibility of stacked paragraphs using Javascript

Visibility of stacked paragraphs using Javascript
EXPERIMENT No.3(b)

AIM: To display the stacking of elements

```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>
    <title>Paragraph Stacking</title>
    <style type="text/css">
      .para
      {
        border: solid thin black;
        padding: 1cm;
        position: absolute;
        width: 300px;
      }
      #layer1
      {
        background-color: yellow;
        top: 200px; left: 400px;
        z-index: 1;
      }
      #layer2
      {
        background-color: red;
        position: absolute;
        top: 220px; left: 420px;
        z-index: 2;
      }
      #layer3
      {
        background-color: blue;
        top: 240px; left: 440px;
        z-index: 3;
      }
    </style>
    <script type="text/javascript">
      var topLayer = "layer3";
      var origPos;
      function mover(toTop, pos) {
        var newTop = document.getElementById(toTop).style;
        newTop.zIndex = 5;
        topLayer = document.getElementById(toTop).id;
        origPos = pos;
      }
    </script>
  </head>
</html>
```
function moveBack()
{
    var layer = document.getElementById(topLayer).style;
    layer.zIndex=origPos;
}
</script>
</head>
<body>
<h3>Visibility of stacked paragraphs using Javascript</h3>
<div id="layer1" class="para" onMouseOver="mover('layer1','1')" onMouseOut="moveBack()">
    10CSL78
</div>
<div id="layer2" class="para" onMouseOver="mover('layer2','2')" onMouseOut="moveBack()">
    Web Programming Laboratory
</div>
<div id="layer3" class="para" onMouseOver="mover('layer3','3')" onMouseOut="moveBack()">
    Experiment 5a - Stacking of paragraphs
</div>
</body>
</html>

**SAMPLE OUTPUT**

Visibility of stacked paragraphs using Javascript

Visibility of stacked paragraphs using Javascript
AIM: To read student details using XML

1. studentDetails.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/css" href="student.css"?>
<VTU>
  <STUDENT>
    <USN>1GD11CS001</USN>
    <NAME>Arun Kumar</NAME>
    <COLLEGE>GOPALAN College of Engineering</COLLEGE>
    <BRANCH>Computer Science and Engineering</BRANCH>
    <YEAR>2011</YEAR>
    <EMAILID>arunk@gmail.com</EMAILID>
  </STUDENT>

  <STUDENT>
    <USN>1GD10ME012</USN>
    <NAME>Swaroop J</NAME>
    <COLLEGE>GOPALAN College of Engineering</COLLEGE>
    <BRANCH>Mechanical Engineering</BRANCH>
    <YEAR>2010</YEAR>
    <EMAILID>swaroopj@gmail.com</EMAILID>
  </STUDENT>

  <STUDENT>
    <USN>1GD12CS030</USN>
    <NAME>Pradeep L</NAME>
    <COLLEGE>GOPALAN College of Engineering</COLLEGE>
    <BRANCH>Computer Science and Engineering</BRANCH>
    <YEAR>2012</YEAR>
    <EMAILID>spradeepl@gmail.com</EMAILID>
  </STUDENT>
</VTU>
```

2. student.css

```css
VTU {
  background-color: #F7fff;
  width: 100%;
}
STUDENT {
  display: block;
  margin-bottom: 30pt;
}
SAMPLE OUTPUT

EXPERIMENT No.4(b)
AIM: To display the student details using XML

1. studentDetails.xml

```xml
<?xml version="1.0" encoding="utf-8"?><VTU>
  <STUDENT>
    <USN>1GD11CS001</USN>
    <NAME>Arun Kumar</NAME>
    <COLLEGE>GOPALAN College of Engineering</COLLEGE>
    <BRANCH>Computer Science and Engineering</BRANCH>
    <YEAR>2011</YEAR>
    <EMAILID>arunk@gmail.com</EMAILID>
  </STUDENT>
  <STUDENT>
    <USN>1GD10ME012</USN>
    <NAME>Swaroop J</NAME>
    <COLLEGE>GOPALAN College of Engineering</COLLEGE>
    <BRANCH>Mechanical Engineering</BRANCH>
    <YEAR>2010</YEAR>
    <EMAILID>swaroopj@gmail.com</EMAILID>
  </STUDENT>
  <STUDENT>
    <USN>1GD12CS030</USN>
    <NAME>Pradeep L</NAME>
    <COLLEGE>GOPALAN College of Engineering</COLLEGE>
    <BRANCH>Computer Science and Engineering</BRANCH>
    <YEAR>2012</YEAR>
    <EMAILID>spradeepl@gmail.com</EMAILID>
  </STUDENT>
</VTU>
```

2. student.xsl

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
      </head>
      <style>
        table, th, td {
          ...
<h2>VTU Student Details</h2>
<table>
  <tr bgcolor="#EEDD82">
    <th>USN</th>
    <th>Name</th>
    <th>College</th>
    <th>Branch</th>
    <th>Year</th>
    <th>Email ID</th>
  </tr>
  <xsl:for-each select="VTU/STUDENT">
    <tr>
      <td><xsl:value-of select="USN"/></td>
      <td><xsl:value-of select="NAME"/></td>
      <td><xsl:value-of select="COLLEGE"/></td>
      <td><xsl:value-of select="BRANCH"/></td>
      <td><xsl:value-of select="YEAR"/></td>
      <td><xsl:value-of select="EMAILID"/></td>
    </tr>
  </xsl:for-each>
</table>
## EXPERIMENT No.5(a)

### VTU Student Details

<table>
<thead>
<tr>
<th>USN</th>
<th>Name</th>
<th>College</th>
<th>Branch</th>
<th>Year</th>
<th>Email ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1M11CS801</td>
<td>Arun Kumar</td>
<td>MVJ College of Engineering</td>
<td>Computer Science and Engineering</td>
<td>2011</td>
<td><a href="mailto:arunkb@gmail.com">arunkb@gmail.com</a></td>
</tr>
<tr>
<td>1M10ME012</td>
<td>Swaroop J</td>
<td>MVJ College of Engineering</td>
<td>Mechanical Engineering</td>
<td>2010</td>
<td><a href="mailto:swaroopj@gmail.com">swaroopj@gmail.com</a></td>
</tr>
<tr>
<td>1M12AE039</td>
<td>Pradeep L</td>
<td>MVJ College of Engineering</td>
<td>Aeronautical Engineering</td>
<td>2012</td>
<td><a href="mailto:pradeepl@gmail.com">pradeepl@gmail.com</a></td>
</tr>
</tbody>
</table>
AIM : To display server information using Perl Programming

#!/usr/bin/perl -w

use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

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 "<p> This information is sent by web server to every CGI program. \\
"

print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"
print "<hr />\n"

exit(0);

SAMPLE OUTPUT

![About this server](attachment:image)

**About this server**

This information is sent by web server to every CGI program.

<table>
<thead>
<tr>
<th>Server name</th>
<th>localhost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running on port</td>
<td>80</td>
</tr>
<tr>
<td>Server Software</td>
<td>Apache/2.2.17 (Ubuntu)</td>
</tr>
<tr>
<td>CGI-Revision</td>
<td>CGI/1.1</td>
</tr>
<tr>
<td>Root Directory of Server</td>
<td>/var/www</td>
</tr>
<tr>
<td>Browser Type</td>
<td>Mozilla/5.0 (X11; Linux x86_64; rv:2.0) Gecko/20100101 Firefox/4.0</td>
</tr>
<tr>
<td>Full pathname of current CGI</td>
<td>/usr/lib/cgi-bin/server.pl</td>
</tr>
</tbody>
</table>
EXPERIMENT No.5(b)

AIM : To execute UNIX commands and to display the output

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

1. cmd.html
<html>
<head>
  <title>Execute UNIX Command</title>
</head>

<body>
<h3>Execute a UNIX Command</h3>
<form action="http://localhost/cgi-bin/cmd.pl" method="GET">
  Enter a UNIX command : <input type="text" name="com" />
  <input type="submit" value="Execute Command" />
</form>
</body>
</html>

2. cmd.pl (Store in /var/www/cgi-bin directory, change permissions using chmod 777 cmd.pl)
#!/usr/bin/perl -w

use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

print "content-type: text/html \n\n"
print "<html>\n"
print "<head> <title> Execute UNIX Command </title> </head>\n"

$c=param('com');
print "<body> <h1> Execute UNIX Command </h1>\n"
print "<h3>\n", $c, " <h3>\n"
print "<a href='$ENV{HTTP_REFERER}'>Back</a>\n"
print "<pre>\n"; system($c);
print "</pre>\n"; exit(0);

SAMPLE OUTPUT
EXPERIMENT No.6(a)
AIM: To display the messages randomly and to display the number of users visiting the pages using Perl Programming

1. greetings.pl

#!/usr/bin/perl -w

use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

@coins = ("Welcome to Web Programming Lab","Have a nice time in lab","Practice all the programs","Well done Good Day");
$range = scalar (@coins);
$random = int(rand($range));

print header();
print start_html(-title=>"Greetings",-bgcolor=>"#FFD800",-text=>"#800000");
print h1("Random Greetings");

if(param)
{
  $cmd=param("name");
  print b("Hello $cmd, $coins[$random]"),br();
  print start_form();
  print submit(-value=>"Back");
  print end_form();
}
else
{
  print h3("Enter your Name ");
  print start_form(),textfield(-name=>"name",-value=>""), submit(-name=>"submit",-value=>"Submit"), reset();
  print end_form();
}

print end_html();

SAMPLE OUTPUT
Random Greetings

Enter your Name

Arun

Submit  Reset

Random Greetings

Hello Arun, Have a nice time in lab

Back

Random Greetings

Hello Lisa, Well done Good Day

Back
EXPERIMENT No.6(a)

AIM: To display the messages randomly and to display the number of users visiting the pages using Perl Programming

2. count.pl
#!/usr/bin/perl

use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

print header();
print start_html(-title=>"Visitor Counter",-bgcolor=>"#FFD800", -text=>"#800000");
print h1("Welcome to Web Programming lab");
print h3("10CSL78");
print p("Experiment 6b : To display the count of visitors on a html page");

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

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

SAMPLE OUTPUT

Welcome to Web Programming lab

10CSL78

Experiment 6b : To display the count of visitors on a html page

This page has been viewed 20 times
EXPERIMENT No. 7

AIM: To display a digital clock using Perl programming

1. time.pl

#!/usr/bin/perl -w

use CGI qw(:standard);

use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

print "Refresh: 1\n";

print header();

print start_html(-title=>"Digital Clock",-bgcolor=>"indigo", -text=>"yellow");

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

print h4("The current system time is $h:$m:$s");

print hr();

print h5("In words $h hours $m minutes $s seconds");

print end_html;

SAMPLE OUTPUT

The current system time is 11:27:30

In words 11 hours 27 minutes 30 seconds
EXPERIMENT No.8

AIM: To display a name and age of user information by creating table using Perl programming

1. Create userDB database
   - Go to terminal, start mysql service
     
     service mysqld start
   
   - Execute the command:
     
     mysql
   
   - You will get the mysql> prompt
   - Create userDB database
     
     mysql> create database userDB;
   
   - Select database userDB
     
     mysql> use userDB;
   
   - Create user table with three attributes (fname, lname and age)
     
     mysql> create table user (fname varchar(50), lname varchar(50), age int);
   
   - To view the database
     
     mysql> show databases;
   
   - To check whether values are inserted in the table
     
     mysql > select * from user;

2. userForm.html

<html>
<head>
  <title>User Details Form</title>
</head>
<body bgcolor="#CCFFCC" text="green">
<h3>User Details</h3>
<form action="http://localhost/cgi-bin/display.pl">
<table>
  <tr>
    <td>First Name</td>
    <td><input type="text" name="fname" /></td>
  </tr>
  <tr>
    <td>Last Name</td>
    <td><input type="text" name="lname" /></td>
  </tr>
  <tr>
    <td>Age</td>
    <td><input type="text" name="age" /></td>
  </tr>
</table>
</form>
</body>
</html>
3.display.pl

#!/usr/bin/perl -w

use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

print header();
print start_html(title=>'Display User Details', -bgcolor=>'#FFDFFF', -text=>'800080');

use DBI;
$dbh=DBI->connect("DBI:mysql:userDB","root","");
(fname)=param("fname");
 lname=param("lname");
 age=param("age");
$qh=$dbh->prepare("insert into user values('$fname', '$lname', $age)");
$qh->execute();
$qh=$dbh->prepare("Select * from user");
$qh->execute();
print h3("User Details");

print "<a href='$ENV{HTTP_REFERER}'>Enter user details</a>";

print "<table width='500px' border='1' style='border-collapse:collapse'>";
print "<tr><th></th><th>FIRST NAME</th><th>LAST NAME</th>
 <th>AGE</th></tr>";

$cnt = 0;
while ( ($fname,lname,$age)=$qh->fetchrow())
{
  $cnt++;
  print "<tr><td>$cnt</td><td>$fname</td><td>$lname</td><td>$age</td></tr>";
}

print "</table>";
$qh->finish();
$dbh->disconnect();
print end_html;
SAMPLE OUTPUT

**User Details**

First Name: Sneha  
Last Name: L  
Age: 20  

**User Details**

Enter user details

<table>
<thead>
<tr>
<th></th>
<th>FIRST NAME</th>
<th>LAST NAME</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Arun</td>
<td>Kumar</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>Swetha</td>
<td>Bhaskar</td>
<td>19</td>
</tr>
</tbody>
</table>
AIM: To display the date and time of last visited page using cookie

1. visit.php

```php
<html>
<head>
<title>Last Visit using Cookies</title>
</head>
<body bgcolor="#C0FFC0" text="#003300">

<h1>Web Programming Lab</h1>
<p>Welcome to Web Programming Lab</p>
<hr />

<p style="color:blue; font-style: italic">
<?php
date_default_timezone_set('Asia/Calcutta');

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

// set expiry date to two months from now
$inTwoMonths = 60 * 60 * 24 * 60 + time();

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

if(isset($_COOKIE['lastVisit']))
{
    $visit = $_COOKIE['lastVisit'];
echo "Last Visited on : ".$visit;
}
else
    echo "You've got some old cookies!";
?>
</p>
</body>
</html>
```
Web Programming Lab

Welcome to Web Programming Lab

You've got some old cookies!

Web Programming Lab

Welcome to Web Programming Lab

Last Visited on: 11:53 - 06/11/14
EXPERIMENT No.10

AIM: To display the session count using PHP programming

1. views.php

```php
<html>
<head>
<title>Page Views </title>
</head>
<body bgcolor="#cCCFFCC" text="#003300">
<h1> Web Programming Lab</h1>
<p> Welcome to Web Programming Lab </p>
<hr />

<p style="color:blue; font-style: italic">
<?php
session_start();
session_register("count");

if (!isset($_SESSION["count"]))
{
   $_SESSION["count"] = 0;
   echo "Counter initialized... <br ";
}
else { $_SESSION["count"]++; }

echo "Number of Page Views : <b>$_SESSION[count]</b></p>";
?><p>Reload this page to increment</p>
</body>
</html>
```
SAMPLE OUTPUT

Web Programming Lab
Welcome to Web Programming Lab

Counter initialized...
Number of Page Views: 0
Reload this page to increment

Web Programming Lab
Welcome to Web Programming Lab

Number of Page Views: 7
Reload this page to increment
EXPERIMENT No.11

AIM: To display the student details using PHP programming by creating database

1. **Create contactDB database in mySQL**
   - Go to terminal, start mysql service
     service mysqld start
   - Execute the command:
     mysql
   - You will get the mysql> prompt
   - Create contactDB database
     mysql> create database contactDB;
   - Select database contactDB
     mysql> use contactDB;
   - Create contact table with 4 attributes (name, addr1, addr2, email)
     mysql> create table contact(name varchar(100), addr1 varchar(100), addr2 varchar(100), email varchar(100));
   - To view the database
     mysql> show databases;
   - To check whether values are inserted in the table
     mysql > select * from contact;

2. **menu.html**

```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>
<title> Menu </title>
</head>
<body bgcolor="#CCFFFF" text="#660099">
<h1> Menu </h1>
<ul>
<li><a href="/contact.php"> Add Contact </a> </li>
<li><a href="/search.php"> Search for Contacts </a> </li>
</ul>
</body>
</html>
```

3. **contact.php**

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" 
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```
<?php
$self = $_SERVER['PHP_SELF'];
$dbh=mysql_connect("localhost", "root", ")or die(mysql_error());
mysql_select_db('contactDB') or die(mysql_error());

if(isset($_POST['submit'])){ 
    $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());
        header("Location: /menu.html");
        die();
    } else
        echo "<p>One of the required fields is empty!";
}

<form action="<?=$self?>" method = "POST">
<h1>  Enter the contact Details </h1>
<p> Go to <a href="/menu.html">Menu</a></p>
<table>
<tr>
    <td> Name </td>
    <td><input type="text" name="name" /></td>
</tr>
<tr>
    <td> Address Line 1 </td>
    <td> <input type="text" name="add1" /></td>
</tr>
<tr>
    <td> Address Line 2 </td>
    <td> <input type="text" name="add2" value="" /></td>
</tr>
</table>
</form>
<tr>
<td>Email </td>
<td><input type="text" name="email" value=""/></td>
</tr>
<tr>
<td colspan="2" align = "center">
<input type="submit" value="SUBMIT" />
<input type="hidden" name="submit" value="yes" />
</td>
</tr>
</table>
<form action="<?=$self?>" method="GET">
Enter Name  : <input type="text" name="name"/>
<input type="hidden" name="search" />
<input type="submit" value = "Search" />
</form>
<?php
if(isset($_GET['search'])) {

</html>

4. search.php

<?php
$self =$_SERVER['PHP_SELF'];
?>
<form action="<?=$self?>" method="GET">
Enter Name  : <input type="text" name="name" />
<input type="hidden" name="search" />
<input type="submit" value = "Search" />
</form>
<?php
if(isset($_GET['search'])) {


$dbh = mysql_connect("localhost", "root", "") or die(mysql_error());
mysql_select_db('contactDB') or die(mysql_error());

$nme=$_GET['name'];
echo "<p>Searching for $nme...</p>";

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

if(mysql_num_rows($query) > 0) {
    ?
    <table border="1" style="border-collapse:collapse; color:#404040">
        <tr>
            <th>Name</th>
            <th>Address Line 1</th>
            <th>Address Line 2</th>
            <th>Email</th>
        </tr>
        <?php
        while ($row=mysql_fetch_array($query)) {
            echo "<tr>   <td>$row[0]</td>  <td>$row[1]</td>
        }
    } else
    echo "<p><b>No matches found...</b></p>";

    mysql_free_result($row);
    mysql_close($dbh);

    ?
    </table>

</body>
</html>
SAMPLE OUTPUT

Menu

- Add Contact
- Search for Contacts

Enter the contact Details

Go to Menu

Name: Rupesh Kumar
Address Line 1: #50, D.N. Layout
Address Line 2: Bangalore
Email: rupeshk@gmail.com

* Required Fields

One of the required fields is empty!

Enter the contact Details

Go to Menu

Name
Address Line 1
Address Line 2
Email

SUBMIT
Search for Contacts

Go to Menu

Enter Name: kumar

Searching for kumar...

<table>
<thead>
<tr>
<th>Name</th>
<th>Address Line 1</th>
<th>Address Line 2</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swathi Kumar</td>
<td>#20, Crimson Layout</td>
<td>Bangalore</td>
<td><a href="mailto:swathiK@gmail.com">swathiK@gmail.com</a></td>
</tr>
<tr>
<td>Rupesh Kumar</td>
<td>#50, D.N. Layout</td>
<td>Bangalore</td>
<td><a href="mailto:rupeshk@gmail.com">rupeshk@gmail.com</a></td>
</tr>
</tbody>
</table>

Search for Contacts

Go to Menu

Enter Name: jane

Searching for jane...

No matches found...

EXPERIMENT No. 12
AIM: To display the book information using Rails

**Software used:** xampp server, Instantrails2.0

Save instantrails2.0 in c:/
Start xampp
  - Goto C:\Instantrails2.0 there will be an icon I, double click on that.

Click on I (left side)- Rails applications → ruby console window

1. Create Database
Type this code at command prompt to login to mysql server as root and get the mysql prompt

> mysql -u root

mysql> create database bookApp_development;

mysql> create database bookApp_test;

mysql> create database bookApp_production;

mysql> use bookApp_development;

mysql> create table books

(ID INT NOT NULL AUTO_INCREMENT,
 acc_num int NOT NULL,
 title VARCHAR(150) NOT NULL,
 authors VARCHAR(150) NOT NULL,
 edition INT(2),
 publisher VARCHAR(150),
 PRIMARY KEY(ID)
 );

Quit MySQL by typing

mysql> quit;

2. Generate the ruby script

C:\InstantRails-2.0-win\rails_apps> rails -d mysql bookApp

C:\InstantRails-2.0-win\rails_apps> cd bookApp

C:\InstantRails-2.0-win\rails_apps\bookApp> ruby script/generate scaffold Book acc_num:int title:string authors:string, edition:int publisher:string

Start the application with mongrel

C:\InstantRails-2.0-win\rails_apps\bookApp> ruby script/server
3. Open the application in browser to insert data into table books

Click on the “New book” link to insert data into the table books
Click on Create

Click “Back” to see the listing
4. Create Views

Press control-c to stop the mongrel in command prompt and type

C:\InstantRails-2.0-win\rails_apps\bookApp> ruby script/generate controller main

This will create

main_controller.rb in C:\InstantRails-2.0-win\rails_apps\bookApp\app\controllers
main folder in C:\InstantRails-2.0-win\rails_apps\bookApp\app\views

main_controller.rb

class MainController < ApplicationController
  def welcome
    @num_books = Book.count
  end
  def result
@bookid = params[:sid]
@bookz = Book.find(:all,:conditions => "id = #{@bookid}")
end
end

5. Generate model

At the command prompt type,

C:\InstantRails-2.0-win\rails_apps\bookApp> ruby script/generate model book

This will create book.rb in C:\InstantRails-2.0-win\rails_apps\bookApp\app\models
directory.

6. Write rhtml pages to search for books.

Save the following two programs in \bookApp\app\views\main folder.

result.rhtml

<html>
<head>
<title> Welcome template for books </title>
</head>
<body bgcolor="#CCFFCC" text="#003800">
<h1>Welcome</h1>
<ul>
<li><a href="../books/new"> Add new book </a></li>
<li><a href="../books">View Book Listing</a></li>
</ul>
<h3> Search for books</h3>
<p> Total number of books : <%= @num_books %> </p>
<form action = "result" >

Enter title of book : <input type="text" name="sid" />

<input type=submit value="Search" />  
</form>

</body>
</html>

result.rhtml

<html>
<head>
<title> Welcome template for books </title>
<style>


table {
border-collapse: collapse;
}
th {
background-color: #003300;
color: #CCCCCC;
}
th, td { padding: 5px; }
</style>
</head>

<body bgcolor="#CCFFCC" text="#003800">
<h1> Search Results </h1>
<p> Search Results for book title containing <b> <%= @booktitle %>    </b></p>
</body>
</html>
<table border="1">
<tr>
<th>Accession Number</th>
<th>Title</th>
<th>Authors</th>
<th>Edition No.</th>
<th>Publisher</th>
</tr>
<% @bookz.each do |bk| %>
<tr>
<td><%= @acc_num %></td>
<td><%= @title %></td>
<td><%= @author %></td>
<td><%= @edition %></td>
<td><%= @publisher %></td>
</tr>
<% end %>
</table>
Start the application with mongrel
C:\InstantRails-2.0-win\rails_apps\bookApp> ruby script/server

7. Open the application in browser

Welcome
• Add new book
• View Book Listing

Search for books
Total number of books: 1
Enter title of book: [ ] Search
VIVA QUESTIONS

1. What is HTML?

2. What is a tag?

3. What is the simplest HTML page?

4. How do I create frames? What is a frameset?

5. How can I include comments in HTML?

6. What is a Hypertext link?

7. What is a DOCTYPE? Which one do I use?

8. Can I nest tables within tables?

9. How do you align a table to the right (or left)?

10. How can I use tables to structure forms?
11. What’s relationship between JavaScript and ECMAScript?
12. What are JavaScript types?
13. How do you convert numbers between different bases in JavaScript? –
14. What does isNaN function do? –
15. What is negative infinity? –
16. What boolean operators does JavaScript support?
17. What are Cascading Style Sheets?
18. What is class?
19. What are different selector forms
20. What is grouping?
21. What is ID selector?
22. What is contextual selector?
23. What does \ABCD (and \ABCDE) mean?
24. What are the advantages / disadvantages of various style methods?
25. What is property?
26. What is the CSS clear property?
27. What are the necessities of using HTML forms?
28. What are the sequences of steps for each HTTP request from a client to the server?
29. What is XML?
30. What are the advantages of XML?
31. What does "1"+2+4 evaluate to? –
32. How about 2+5+"8"?
33. What looping structures are there in JavaScript?
34. How do you create a new object in JavaScript?

35. How do you assign object properties?

36. What’s a way to append a value to an array?

37. What is this keyword?

38. What is an Empty HTML Tag?

39. How do I open a link into a new window?

40. How do I let people download a file from my page?