WEB TECHNOLOGY LABORATORY WITH MINI PROJECT LAB MANUAL

VII Semester
Course Code: 15CSL77

As per the Choice Based Credit System Scheme
Scheme: 2015

Version 1.1

With effect from: August 2019

Editorial Committee

WEB Lab Faculty, Department of CSE

Approved by

H.O.D, Department of CSE

Global Academy of Technology

Department Of Computer Science and Engineering
Rajarajeshwari Nagar, Bangalore – 560 098

DOCUMENT LOG

Name of the document	Web Technology Lab with Mini Project Manual
Current version number	1.1
Date of Updation	7-7-2019
Subject code	15CSL77
Authored by	Mr Shyam Sundar Bushan, Ms. Sushmitha S Ms. Snigdha Sen
Verified by	Dr Kavitha K S
Approved by	HoD, Dept of CSE

Table of Contents

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision of the Department

To achieve academic excellence and strengthen the skills to meet emerging challenges of Computer Science and Engineering.

Mission of the Department

M1: To impart strong theoretical foundations in the field of Computer Science and Engineering accompanied with extensive practical skills.

M2: To inculcate research and innovation spirit through interaction with industry and carry out projects that address societal needs.

M3: Instill professional ethics and values with concern for environment.

Program Educational Objectives (PEOs) of Department

After the course completion, CSE graduates will be able to:

- Succeed in engineering/management positions with professional ethics.
- Engage in improving professional knowledge through certificate/post-graduate programs in engineering or management.
- Establish themselves as entrepreneurs and contribute to the Society.

Program Specific Outcomes (PSOs)

PSO1: Design, implement and test System Software and Application Software to meet the desired needs.

PSO2: Develop solutions in the area of Communication Networks, Database Systems and Computing Systems.

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Details

Course Name **:** Web Technology Laboratory with Mini Project

Course Code **:** 15CSL77

Course prerequisite **:** Basic of programming Languages

Course Objectives

1. Design and develop static and dynamic web pages
2. Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages.
3. Learn Database Connectivity to web applications.

Syllabus

Subject Code : 15CSL77	**IA Marks : 20**
No. of Practical Hrs/ Week : 01I + 02P	**Exam Hours : 03**
Total No. of Practical Hrs : 40	**Exam Marks : 80**

Part A
Laboratory Experiments

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.
2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
3. Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT-SHRINKING" in BLUE color. Then the font size decreases to 5pt.
4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
    a. Parameter: A string
    b. Output: The position in the string of the left-most vowel
    c. Parameter: A number
    d. Output: The number with its digits in the reverse order
5. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
6. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
7. Write a PHP program to display a digital clock which displays the current time of the server.
8. Write the PHP programs to do the following:
    a. Implement simple calculator operations.
    b. Find the transpose of a matrix.
    c. Multiplication of two matrices.
    d. Addition of two matrices.

9. Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:
    a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.
    b. Search for a word in states that begins with k and ends in s. Perform a case insensitive comparison. [Note: Passing re.Ias a second parameter to method compile performs a case-insensitive comparison.] Store this word in element1 of statesList.
    c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.
    d. Search for a word in states that ends in a. Store this word in element 3 of the list.
10. Write a PHP program to sort the student records which are stored in the database using selection sort.

Part B
Study Experiment / Project

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

Note:

1. In the examination each student picks one question from part A.

2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.

3. The team must submit a brief project report (15-20 pages) that must include the following

a. Introduction

b. Requirement Analysis

c. Software Requirement Specification

d. Analysis and Design

e. Implementation

f. Testing

Course Outcomes

Upon successful completion of this course, students are able to:

COs	COURSE OUTCOMES
CO1	**Design** and **Develop** dynamic web pages with good aesthetic sense of designing and latest technologies.
CO2	**Make use of** good Web Application Terminologies, Internet Tools other web services for building a mini project.
CO3	**Identify** how to link and publish web sites.

Conduction of Practical Examination:

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 30 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
    a. Part A: Procedure + Conduction + Viva:10 + 35 +5 =50 Marks
    b. Part B: Demonstration + Report + Viva voce = 15+10+05 = 30 Marks

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

CO-PO-PSO MAPPING

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
15CSL77.1	3	2	2	-	1	-	-	1	1	1	-	1	2	-
15CSL77.2	3	3	3	2	1	1	1	1	1	2	1	1	3	-
15CSL77.3	3	3	3	2	1	1	1	1	1	2	1	1	3	-
AVG	**3**	**3**	**3**	**2**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**1**	**3**	**-**

LAB EVALUATION PROCESS

Rubrics for WEB Lab (Part A)

Sl. No	Criteria	Marks
1	Record +observation	4+4
2	Write Up	10
3	Execution	25
4	Viva	5
TOTAL		**40**

Rubrics for WEB Mini Project Evaluation (Part B)

Sl. No.	Criteria	Marks
1	Requirement Collection	**1**
2	Design Document	**1**
3	Back End Design	**1**
4	Front End Design	**1**
5	Project Demo(Demonstration & Questionnaire Session)	**2**
6	Report	**2**
TOTAL		**8**

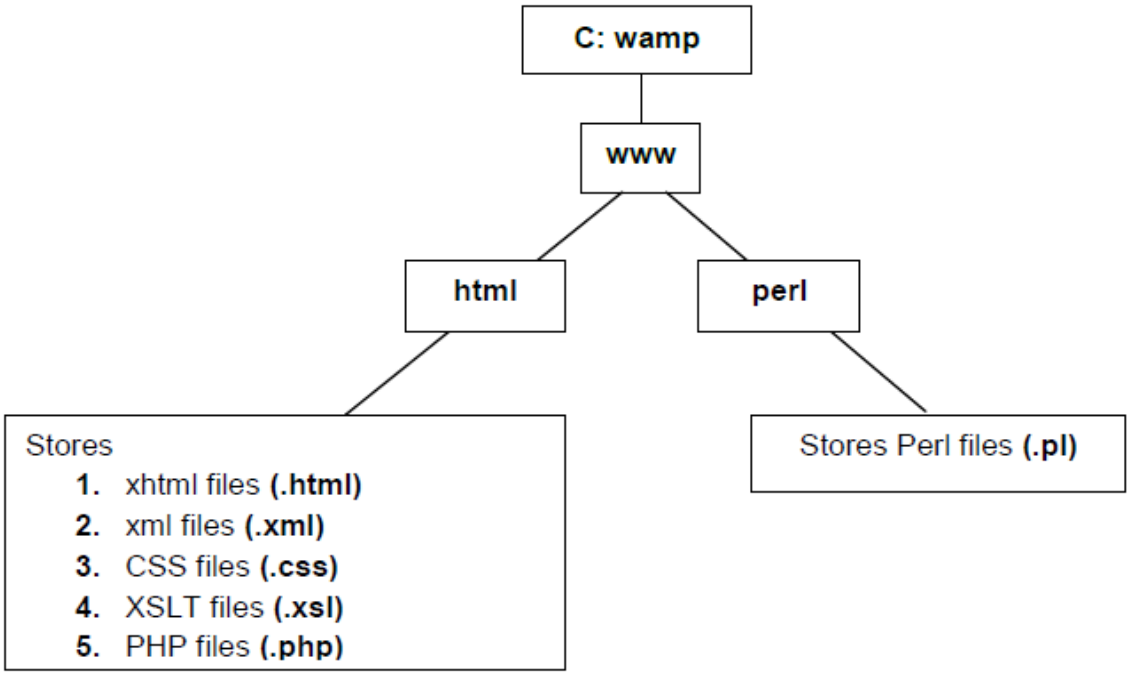
External Assessment Evaluation (End of Semester)		
Sl. No	**Activity**	**Marks**
1	**Part A:** Procedure + Conduction + Viva	**10+35+5=50**
2	**Part B:** Demonstration + Report + Viva voce	**15+10+5=30**
TOTAL		

Final Internal Assessment Calculation		
Sl.No	**Activity**	**Marks**
1	Average of Weekly Entries	**8**
2	Internal Assessment Reduced To	**8+4**
TOTAL		**20**

CHAPTER 1

Lab Instructions

Folders and files used in the Web programming



Execution of programs

1. HTML, XML, PHP files

In the browser window, type localhost/html/filename.extension

Example :localhost/html/1a.html

2. Programs with MySQL

1. Creating a table in MySQL

In the browser, type localhost/phpmyadmin (MySQL Homepage)

Click on the database **test**

Click on SQL menu and write SQL query statement

2. Execution

Run at browser: **localhost/html/filename.ext**

CHAPTER 2

Introduction to web

In 1989, a small group of people led by Tim Berners-Lee (Father of web) proposed a new protocol for the Internet. They named it as World Wide Web that was to allow scientists around the world to use Internet to exchange documents describing their work.

Web is a vast collection of documents, some of which are connected by links. These documents are accessed by web browsers and are provided by web servers. In 1989, a small group of people led by Tim Berners-Lee proposed a new protocol for the Internet. They named it as World Wide Web, that was to allow scientists around the world to use Internet to exchange documents describing their work. For the documents, the system used hypertext (text with embedded links to text in other documents).Web is a vast collection of documents, some of which are connected by links. These documents are accessed by web browsers and are provided by web servers.

2.1. Web Browser

It is a program used to view HTML documents. A web browser runs on a client machine which communicates with the servers.

Common browsers: IE, Firefox, Google chrome, Suite, Opera, Safari, Microsoft edge.

The name browser because they allow the user to browse the resources available on servers. A browser is a client on the web because it initiates the communication with a server, waits for a request from a client before doing anything. The server locates the document and sends it to the browser, which displays it for the user. Web supports many protocols, one of which is HTTP. HTTP provides a standard form of communication between browsers and web servers.

2.2. Web Server

Web server is a program that provides documents to requesting browsers. Web servers are the slave programs. They act only when requests are made to them by browsers running on other computers on the Internet. Most commonly used web servers are Apache, Microsoft‟s Internet Information Server (IIS) which runs on Windows OS.

CHAPTER 3

Introduction to HTML

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

3.1. Basic HTML Tags

1. Paragraph
2. Line breaks
3. Preserving whitespace
4. Headings
5. Font styles and sizes
6. Horizontal rules
7. Images
8. Hypertext links
9. Lists
10. Table
11. Form
12. Frame

Tag Description	XHTML Tag	Attributes
Paragraph		align = "left/right/center"
Line Breaks	 	
Preserving white space	<pre></pre>	
Headings	 	align = "left/right/center"
Font styles	bold italic <u></u>underline bold italic superscript subscript	
Horizontal Rules	<hr />	align = "left/right/center" size="value in pixels" width="value is 5 or pixels" color="color"
Images	<img />	src = "path of the image" height="value in pixels" width="value in pixels"
Hypertext links	<a>	href = "path to the linking file"
Lists	unordered list ordered list List Item	
Table	<table></table>Table defining	bgcolor="color"
	<tr></tr> Table row	bgcolor="color"
	<th></th> Table heading	bgcolor="color" align="left/right/center"
	<td></td> Table data	bgcolor="color" align="left/right/center"
Form	<form></form>	action="path to linking file" method="GET/POST"
	<input />	type="text/password/checkbox /radio/submit/reset"
Frame	<frameset></frameset>	Rows="value" Cols="value"
	<frame></frame>	src="path to linking file"

3.2. Sample programs on HTML

Program 1 - HTML code with line break, bold font

```
<html>
<head><title> sample document </title></head>
<body>
<strong>Student Details : <br /></strong>
USN : 1GA12CS0160 <br /> NAME : Amit<br />
COLLEGE : Global Academy of Technology <br />
</body>
</html>
```

Activity

1. Change the title as Web programming
2. Use 2 line spacing for each line
3. Make USN as bold

Program 2 - HTML code with header tags

```
<html>
<head><title> sample document </title></head>
<body bgcolor=”yellow”>
<p align=”center”><strong>Header tags : </strong></p>
<p>All header tags are bold but have different font size.<br /> There are 6
header tags.<br />
<strong>Break tag is automatically encountered</strong> by the header
tags.</p>
<h1>Global Academy of Technology</h1>
<h2>Growing Ahead of Time</h2>
<h3> Ideal Homes Township</h3>
<h4>RR Nagar</h4>
<h5>Bengaluru</h5>
<h6>560098</h6> </body>
</html>
```

Activity

1. Change the title as Header tags
2. Change the background color
3. Give double line spacing after each header
4. Keep the first paragraph left aligned and second paragraph center aligned

Program 3 - HTML code with list

```
<html>
<head><title> sample document </title></head>
<body>
      <h2>Global Academy of Technology</h2>
      <h3>Various Departments</h3>
      <ol>
          <li>CSE
          <li>ISE
          <li>ECE
```

```
            <li>EEE
            <li>Mech
            <li>Civil
        </ol>
        <hr size=”5”>
        <h3>Computer Science and Engineering</h3>
        <ul>
            <li>Academics
            <li>Cultural Events
            <li>Sports
        </ul>
</body>
</html>
```

Activity

1. Add “cyan” as background color to the web page
2. Change the horizontal ruler size and add a color to it
3. Try to nest <ol> and

Program 4 - HTML code for drawing tables

```
<html>
<head><title> Tables </title></head>
<body> <table border="1">
<caption align="center">
<strong> Differences between Internet and Web </strong></caption>
<tr>
    <th> Internet </th>
    <th>Web</th>
</tr>
<tr>
    <td>Connecting computers</td>
    <td>Connecting people</td>
</tr>
<tr>
    <td> Way of transporting content </td>
    <td> Software that lets you use that content </td>
</tr>
<tr>
    <td> Independent of the WWW </td>
    <td> Depends on Internet to work </td>
</tr>
</body>
</html>
```

Activity

1. Change the table border
2. Add background color to the whole table.
3. Add different background colors to each row.

CHAPTER 4

Introduction to CSS

CSS (Cascading Style Sheets) are used for describing the look and formatting of a document written in a markup language. It is a simple text file or text embedded in the head of an HTML document. CSS is a style language that defines styles (e.g. fonts, colors, spacing) and layout of web pages written in HTML/XHTML and XML. The difference between CSS and HTML is that HTML/XHTML is used to structure content and CSS is used for formatting structured content

4.1. Levels of CSS

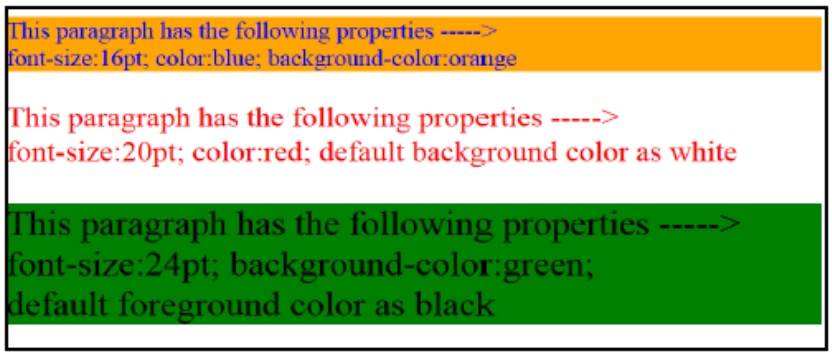
Inline style sheet: using the style attribute in HTML element.

Document level style sheet: using the <style> element in the <head> section.

External style sheet: using the external CSS file.

Example :

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> testing </title>
<style type="text/css">
      p.bg1 { font-size:16pt; color:blue; background-color:orange;}
      p.bg2 { font-size:20pt; color:red;}
      p.bg3 { font-size:24pt; background-color:green;}
</style>
</head>
<body>
<p class="bg1"> This paragraph has the following properties -----><br />font-
size:16pt; color:blue; background-color:orange</p>
<p class="bg2"> This paragraph has the following properties -----><br />
font-size:20pt; color:red; default background color as white</p>
<p class="bg3"> This paragraph has the following properties -----><br />font-
size:24pt; background-color:green; <br />default foreground color as black
</p>
</body>
</html>
```



CHAPTER 5

Introduction to JavaScript

JavaScript is a scripting language (lightweight programming language – not too many constraints).It was designed to add interactivity to HTML pages. It is usually embedded directly into HTML pages. Everyone can use JavaScript without purchasing a license. JavaScript is often used as client-side scripting language.

5.1. JavaScript Syntactic Characteristics

All JavaScript scripts are embedded in XHTML documents.

- Scripts can appear directly as the content of a <script> tag. Type attribute of <script> must be set to "text/javascript"

  Scripts can be indirectly embedded in XHTML document using src attribute of <script> whose value is the name of a file that contains the script. <script type="text/javascript" src="t1.js"></script>
- In JavaScript, identifiers are similar to those of other common programming languages. They must begin with a letter, underscore or $. Subsequent characters may be letters, underscores, $s or digits.
- No length limitation of identifiers.
- Identifiers are case-sensitive.

5.2. Screen Output and Keyboard Input

Normal output screen for Java Script is the same as the screen in which the content of the host XHTML is displayed. Browser models the XHTML document with Window Object. JavaScript models the XHTML document with the document Object

5.2.1 Window Object:

JavaScript model for the browser window. Window includes 3 methods that create dialog boxes for 3 specific kinds of user interactions.

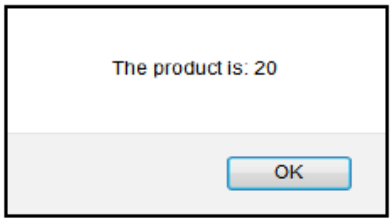
1. Alert method
2. Confirm method
3. Prompt method

Alert method:

It opens a dialog window and displays its parameter in that window. It also displays an OK button. The parameter string to alert is NOT XHTML code, it is the plain text. Hence the string parameter to alert may include \n but never

Example :

```
<html>
<body>
<script type="text/javascript">
var a=10,b=2,c; c=a*b;
alert("The product is: " +c+ "\n");
</script>
</body>
</html>
```

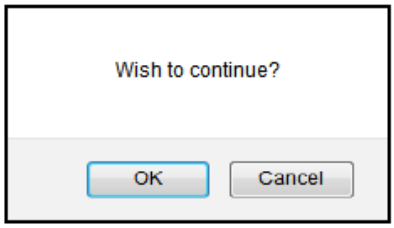


Confirm method:

It opens a dialog window and displays its string parameter along with two buttons, OK and Cancel. Confirm returns a Boolean value that indicates the user's input: true for OK and false for Cancel. This method is often used to offer the user, the choice of continuing some process.

Example :

```
 <html>
<body>
<script type="text/javascript">
var a=confirm("Wish to continue?");
</script>
</body>
</html>
```

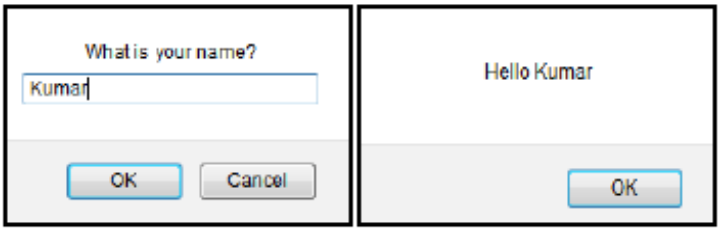


Prompt method:

It creates a dialog window that contains a text box. The text box is used to collect a **string** of input from the user, which prompt returns as its value. The window includes two buttons OK and Cancel. Prompt takes two parameters - The string that prompts the user for input and the default string, in case the user does not type a string before pressing a button.

Example :

```
<html>
<body>
<script type="text/javascript">
var name=prompt("What is your name?", " ");
alert("Hello " + name + "\n");
</script>
</body>
</html>
```



5.2.2 Document Object :

The document object has several properties and methods.

Examples:

document.write("The result is:", result, "
");

document.write("<hr />Hello World Wide Web<hr />");

CHAPTER 6

Introduction to XML

XML is the shorthand name for **E**xtensible **M**arkup **L**anguage. XML is a markup language much like HTML and was designed to describe data. It is developed to retain the flexibility but reduce the complexity of HTML. XML tags are not predefined. You must define your own tags according to your needs. XML was designed to transport and store data. There are two types of XML documents. They are

- Well formed document - XML with correct syntax is "well formed" XML.
- Valid document – XML validated against a DTD is "valid" XML.

6.1 Well-formed document

A "well formed" XML documents has correct XML syntax. The Syntax rules are as follows:

- XML documents must have a root element.
- XML elements must have a closing tag.
- XML tags are case sensitive
- XML elements must be properly nested
- XML attribute values must be quoted.

Example for well-formed document.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<note>
      <to> Principal</to>
      <from> Manjunath</from>
      <heading>Request<</heading>
      <body>Pls allow me to participate in the training workshop
      </body>
</note>
```

The difference between well-formed and valid documents is that a valid document requires a document type (!DOCTYPE) declaration**,** which includes a document type definition(DTD). A well-formed document does not. A DTD defines the elements**,** attributes**,** entities and rules for creating one or more documents in a markup language such as SGML**,**HTML or XML. A valid xml document is a well formed xml document, which also conforms to the rules of a document type definition – DTD.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
      <to> Principal</to>
```

```
      <from> Manjunath</from>
      <heading>Request<</heading>
      <body> Pls allow me to participate in the training workshop
      </body>
</note>
```

The DOCTYPE declaration in the example above , is a reference to an external DTD file. The content of the file is shown in the paragraph below.

<?xml version="1.0"?>

<!ELEMENT note (to,from,heading,body)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT heading (#PCDATA)>

<!ELEMENT body (#PCDATA)>

6.2 XML Structure

XML documents parts:

- Prolog
- Document element

6.2.1 The Prologue

It is equivalent to the header in HTML. It may include the following:

1. An XML declaration. It is optional. It declaration states

1. Version of XML
2. Type of character coding
3. Markup declarations requirement of XML document.

XML declaration is as follows**:**

<? Xml version="1.0" encoding="UTF-8" standalone="yes" ?>

The current and only version of xml is 1.0.

Character encoding formats are

US-ASCII

UTF-8 (It is the default value)

Standalone declaration indicates the presence of external **"**markup declaration**"**.

Standalone declaration with a value of **"**yes**"** indicates that there is no external markup declaration. Standalone declaration with a value of **"**no**"** indicates external markup declaration may exist.

2. DTD or reference to one. It is also optional.

<!DOCTYPE LANGLIST SYSTEM "langlist.dtd">

3. A Processing instruction

<?xml-stylesheet type="text/css" href="xmlstyle.css"?>

6.2.2.Document element

- XML documents must contain **at least one element**.
  Example: <title>Tootsie</title>
- XML documents must contain a **unique opening and closing tag** that contains the whole document, forming what is called a root element.

Example:

<videocollection>

<title>Tootsie</title>

<title>Jurassic Park</title>

<title>Mission Impossible</title>

</videocollection>>

- All other tags must be nested properly
- Tags in XML are case sensitive
- Attribute values must always be quoted (as opposed to HTML).

CHAPTER 7

Introduction to PHP

PHP was developed by Rasmus Lerdorf, a member of Apache Group in 1994.The Initial purpose was to provide a tool to help Lerford track visitors to his personal web page. Original name of PHP was Personal Home Page. Later it was changed to Hyper Text Preprocessor. It is used as server-side scripting language. So PHP is used for form handling and database access.

Example:

```
<html>
<body>
<?php echo "My first PHP script!"; ?>
</body>
</html>
```

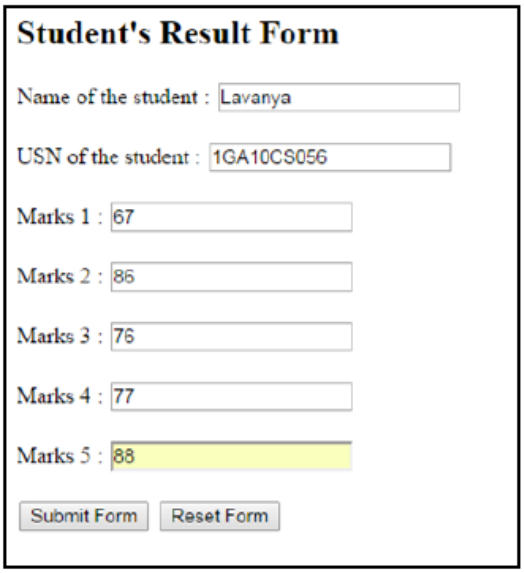
7.1 PHP Form Handling

The PHP super globals $_GET and $_POST are used to collect form-data.

Example:

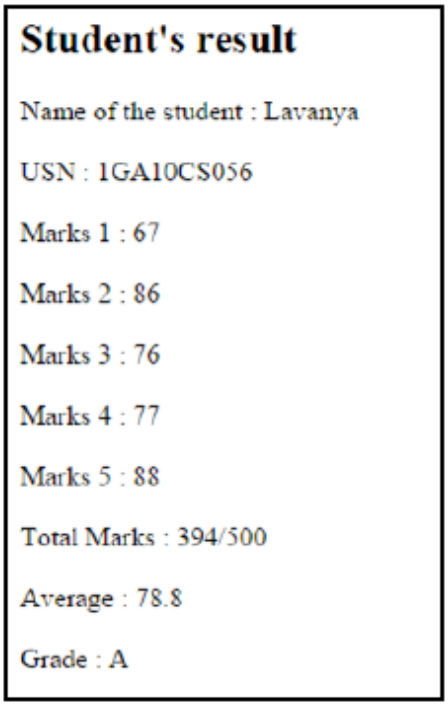
result.html

```
<html>
<body>
<form action="result.php" method="POST">
<h2>Student's Result Form </h2>
Name of the student :
<input type="text" name="name"><br /><br />
USN of the student :
<input type="text" name="usn"><br /><br />
Marks 1 :
<input type="text" name="m1"><br /><br />
Marks 2 :
<input type="text" name="m2"><br /><br />
Marks 3 :
<input type="text" name="m3"><br /><br />
Marks 4 :
<input type="text" name="m4"><br /><br />
Marks 5 :
<input type="text" name="m5"><br /><br />
<input type="submit" value="Submit Form">
<input type="reset" value="Reset Form">
</body>
</html>
```



result.php

```
<html>
<body>
<?php
$name=$_POST["name"];
$usn=$_POST["usn"];
$m1=$_POST["m1"];
$m2=$_POST["m2"];
$m3=$_POST["m3"];
$m4=$_POST["m4"];
$m5=$_POST["m5"];
$total=$m1+$m2+$m3+$m4+$m5;
$avg=$total/5;
if($avg>=75)
$grade="A";
else if($avg>=60)
$grade="B";
else if($avg>=50)
$grade="C";
else if($avg>=35)
$grade="D";
else
$grade="E";
?>
<h2>Student's result </h2>
Name of the student :
<?php echo $name; ?><br /><br />
USN : <?php echo $usn; ?><br /><br />
Marks 1 : <?php echo $m1; ?><br /><br />
Marks 2 : <?php echo $m2; ?><br /><br />
Marks 3 : <?php echo $m3; ?><br /><br />
Marks 4 : <?php echo $m4; ?><br /><br />
Marks 5 : <?php echo $m5; ?><br /><br />
Total Marks :
<?php echo $total."/500"; ?><br /><br />
Average : <?php echo $avg ?><br /><br />
Grade : <?php echo $grade ?><br /><br />
</body>
</html>
```



CHAPTER 8

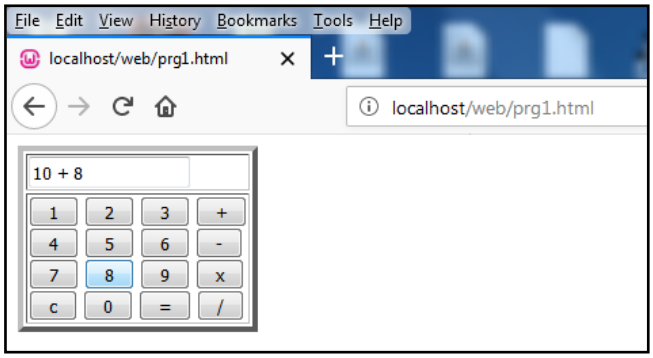
Syllabus Programs

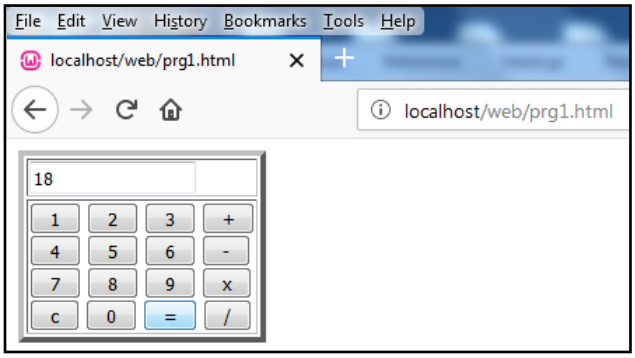
1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.

```
<DOCTYPE html>
<html>
        <body>
<FORM NAME="Calc">
        <TABLE BORDER=4>
        <TR>
        <TD>
        <INPUT TYPE="text"   NAME="Input" Size="16">
        <br>
        </TD>
        </TR>
        <TR>
        <TD>
        <INPUT TYPE="button" NAME="one"   VALUE="  1  " OnClick="Calc.Input.value += '1'">
        <INPUT TYPE="button" NAME="two"   VALUE="  2  " OnCLick="Calc.Input.value += '2'">
        <INPUT TYPE="button" NAME="three" VALUE="  3  " OnClick="Calc.Input.value += '3'">
        <INPUT TYPE="button" NAME="plus"  VALUE="  +  " OnClick="Calc.Input.value += ' + '">
        <br>
        <INPUT TYPE="button" NAME="four"  VALUE="  4  " OnClick="Calc.Input.value += '4'">
        <INPUT TYPE="button" NAME="five"  VALUE="  5  " OnCLick="Calc.Input.value += '5'">
        <INPUT TYPE="button" NAME="six"   VALUE="  6  " OnClick="Calc.Input.value += '6'">
        <INPUT TYPE="button" NAME="minus" VALUE="  -  " OnClick="Calc.Input.value += ' - '">
        <br>
        <INPUT TYPE="button" NAME="seven" VALUE="  7  " OnClick="Calc.Input.value += '7'">
        <INPUT TYPE="button" NAME="eight" VALUE="  8  " OnCLick="Calc.Input.value += '8'">
        <INPUT TYPE="button" NAME="nine"  VALUE="  9  " OnClick="Calc.Input.value += '9'">
        <INPUT TYPE="button" NAME="times" VALUE="  x  " OnClick="Calc.Input.value += ' * '">
        <br>
        <INPUT TYPE="button" NAME="clear" VALUE="  c  " OnClick="Calc.Input.value = ''">
        <INPUT TYPE="button" NAME="zero"  VALUE="  0  " OnClick="Calc.Input.value += '0'">
        <INPUT TYPE="button" NAME="DoIt"  VALUE="  =  " OnClick="Calc.Input.value =
          eval(Calc.Input.value)">
        <INPUT TYPE="button" NAME="div"   VALUE="  /  " OnClick="Calc.Input.value += ' / '">
        <br>
        </TD>
        </TR>
        </TABLE>
</FORM>
</body>
</html>
```

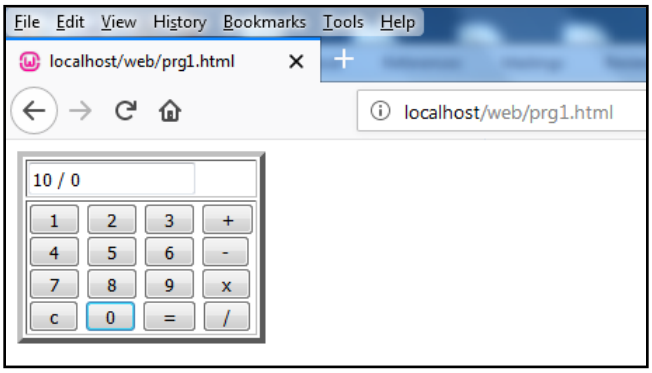
Output:

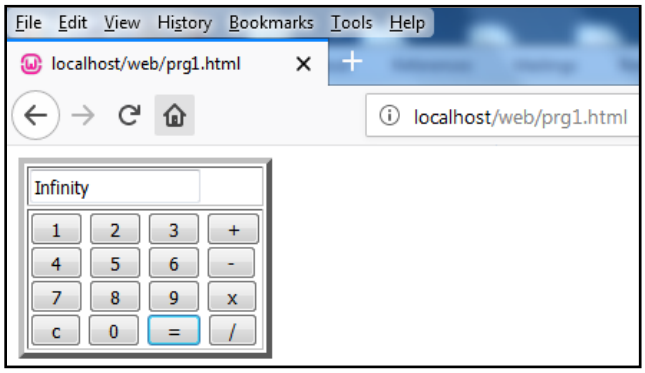
Run 1:





Run 2:

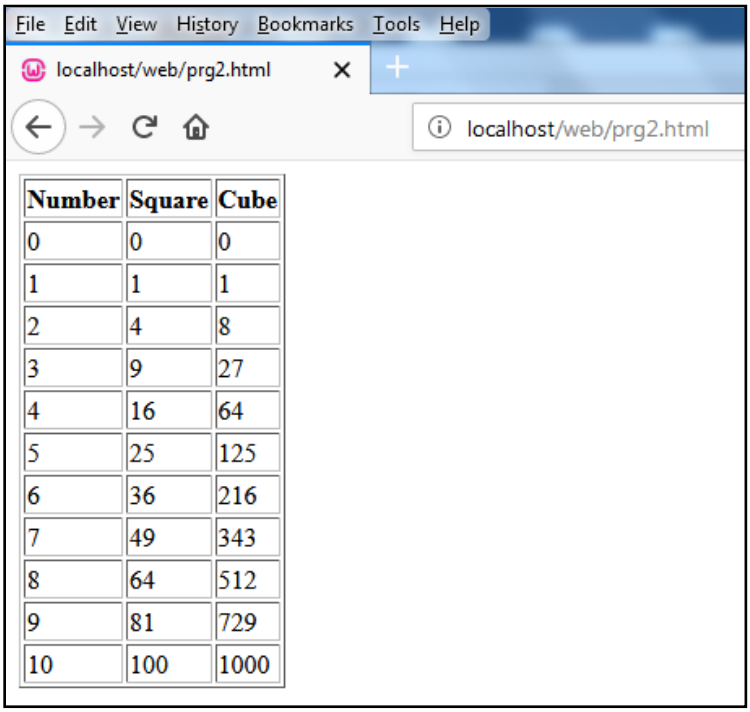




2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.

```
<html>
<head>
<script>
document.write( "<table border> <tr> <th>Number</th> <th>Square</th> <th>Cube</th> </tr>" )
for(var n=0; n<=10; n++)
{
document.write( "<tr><td>" + n + "</td><td>" + n*n + "</td><td>" + n*n*n + "</td></tr>" )
}
document.write( "</table>" )
</script>
</head>
</html>
```

Output:



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

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

```
<!DOCTYPE HTML>
<html>
<head>
<style>
p
{
                position: absolute; top: 50%;
                left: 50%;
                transform: translate(-50%, -50%);
}
</style>
</head>
<body>
<p id="demo"></p>
<script>
                var var1 = setInterval(inTimer, 1000);
                var fs = 5;
                var ids = document.getElementById("demo");
                function inTimer() {
                    ids.innerHTML = 'TEXT GROWING';
                    ids.setAttribute('style', "font-size: " + fs + "px; color: red");
                    fs += 5;
                    if(fs >= 50 ){
                            clearInterval(var1);
                            var2 = setInterval(deTimer, 1000);
                    }
                }
                function deTimer() {
                    fs -= 5;
                ids.innerHTML = 'TEXT SHRINKING';
                ids.setAttribute('style', "font-size: " + fs + "px; color: blue");
                if(fs === 5 ){
                    clearInterval(var2);
                   }
                }
</script>
</body>
</html>
```

Output:

TEXT-GROWING

TEXT SHRINKING

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

a. Parameter: A string

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

c. Parameter: A number

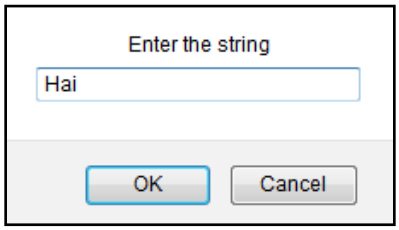
d. Output: The number with its digits in the reverse order

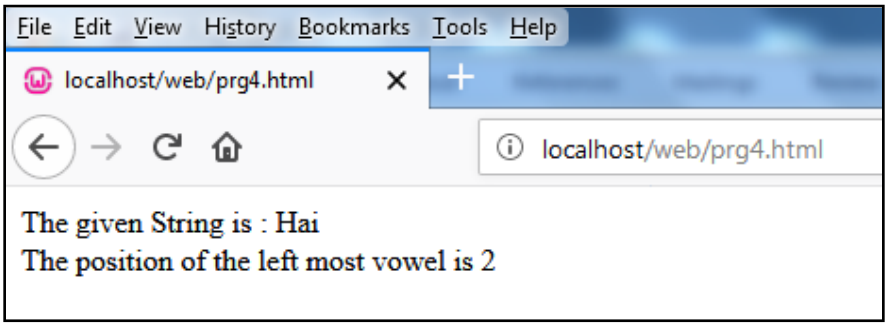
```
<!DOCTYPE HTML>
<?xml version="1.0" encoding="UTF-8"?>
<html>
<head> </head>
</body>
<script type ="text/javascript">
var str=prompt("Enter the string\n", " ");
  if(isNaN(str))
     vowel_pos();
   else
     rev_num();

function vowel_pos()
{
 str1 = str.toLowerCase();
 for(var i = 0; i<str1.length; i++)
 {
      ch = str1.charAt(i);
      if(ch=='a' || ch=='e'||ch=='o'||ch=='u')
      {
        break;
      }
 }

 if(i<str1.length)
    document.write("The given String is : " +str+ " <br/>  The position of the left most vowel is " +
(i+1) + "<br />");
 else
     document.write("No vowel is found in the entered string");

}

function rev_num()
{

var n= str;
var rev = 0, rem;
while (n>0)
```

```
{
rem = n % 10;
rev = rev * 10 + rem ;
n = Math.floor(n/10);
}
document.write("The given number is : " +str+ " <br/> The reversed number is : " +rev+ "\n");
}

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

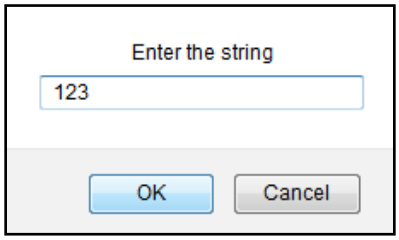
Output:

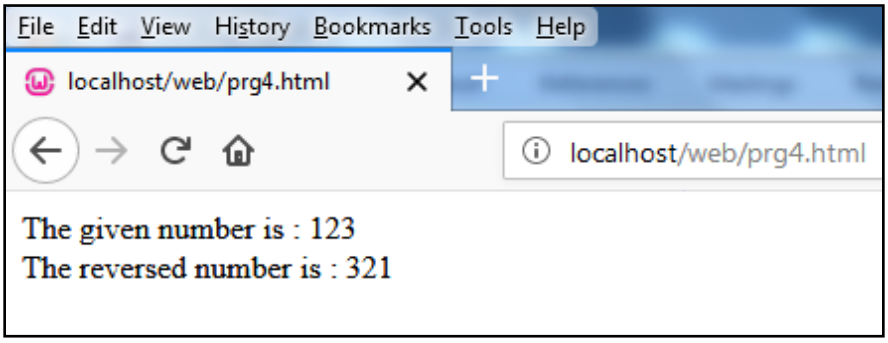
Run 1:





Run 2:





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

5.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/css" href = "5.css" ?>
<student>
<usn> 1GA12CS001 </usn>
<name> Abhishek Ejam</name>
<college> GAT </college>
<branch> CSE</branch>
<yoj> 2012 </yoj>
<email> abhishek@gmail.com </email>

<usn> 1GA12CS055</usn>
<name>Nagashree</name>
<college> GAT </college>
<branch> CSE </branch>
<yoj> 2012 </yoj>
<email> nagashree@gat.ac.in </email>

<usn> 1GA12CS044 </usn>
<name>Madhushree</name>
<college> GAT </college>
<branch> CSE </branch>
<yoj> 2012</yoj>
<email> madhushree@yahoo.com </email>
</student>
```

5.css

```
usn
{
    font-weight:bold;
    color:purple;
    display:block;
}
name
{
    font-style:italic;
    color:blue;
    display:block;
}
college
{
    color:red;
```

```
    display:block;
}
branch
{
    color:green;
    display:block;
}
yoj
{
    color:gold;
    display:block;
}
email
{
    color:olive;
    display:block;
}
```

Output:

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

```
<html>
<head>
<title>Web Page Counter</title>
</head>
<body bgcolor="pink" text="blue">
<?php
$my_file = "count.txt";
$handle = fopen($my_file, "r");
$count = fgets($handle);
$count=$count+1;
$handle = fopen($my_file, 'w') or die('Cannot open file:  '.$my_file);
fwrite($handle, $count);
echo "This page has been viewed $count times";
?>
</body>
</html>
```

Output:

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

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="refresh" content="1"/>
<style>
p {
        color:white;
        font-size:90px;
        position: absolute;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -50%);
}
body{background-color:black;}
</style>
<p> <?php echo date(" h: i : s A");?> </p>
</head>
```

Output:

8. Write the PHP programs to do the following:

a. Implement simple calculator operations.

b. Find the transpose of a matrix.

c. Multiplication of two matrices.

d. Addition of two matrices

8a.html

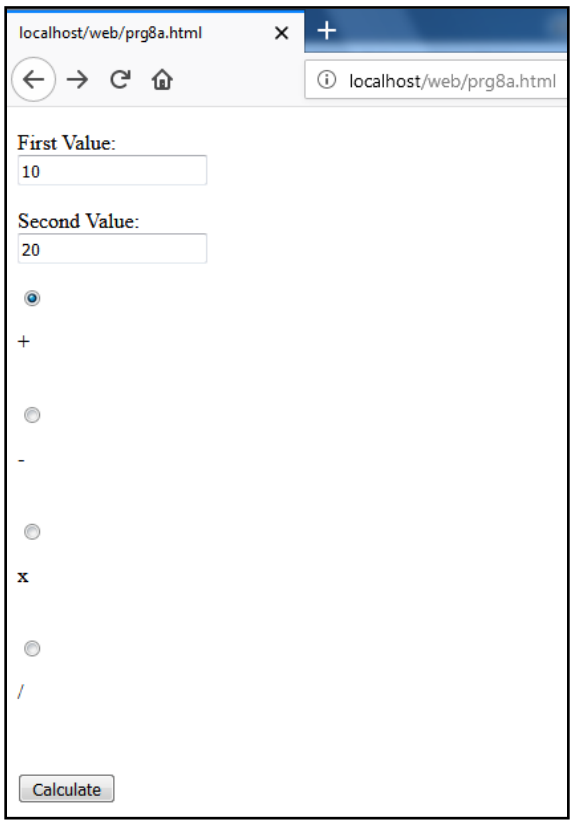
```
<html>
<body>
<form method="post" attribute="post" action="disp_form.php">
<p>First Value:<br/>
<input type="text" id="first" name="first"></p>
<p>Second Value:<br/>
<input type="text" id="second" name="second"></p>
<input type="radio" name="group1" id="add" value="add" checked="true"><p>+</p><br/>
<input type="radio" name="group1" id="subtract" value="subtract"><p>-</p><br/>
<input type="radio" name="group1" id="times" value="times"><p>x</p><br/>
<input type="radio" name="group1" id="divide" value="divide"><p>/</p><br/>
<p></p>
<button type="submit" name="answer" id="answer" value="answer">Calculate</button>
</form>
</body>
</html>
```

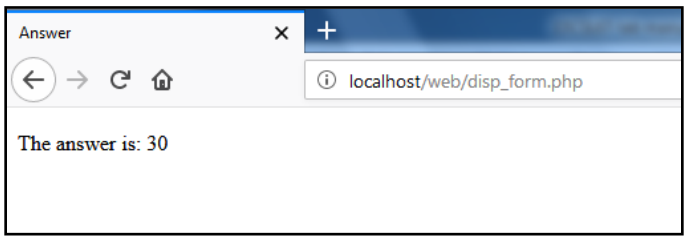
disp_form.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Answer</title>
</head>
<body>
<p>The answer is:
<?php
if($_POST['group1'] == 'add')
{
 echo $_POST['first'] + $_POST['second'];
}
else if($_POST['group1'] == 'subtract')
{
echo $_POST['first'] - $_POST['second'];
}
else if($_POST['group1'] == 'times')
{
echo $_POST['first'] * $_POST['second'];
}
else if($_POST['group1'] == 'divide')
```

```
{
echo $_POST['first'] / $_POST['second'];
}
?>
</p>
</body>
</html>
```

Output:





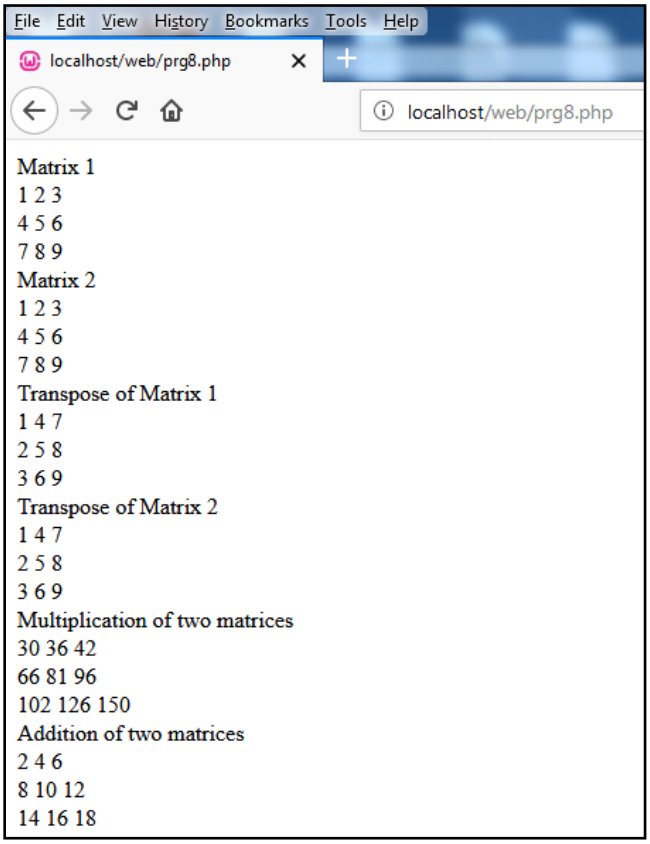
prg8.php

```
<?php
function transpose($data)
{
  $transData = array();
    foreach ($data as $row => $arr) {
      foreach ($arr as $col => $val) {
          $transData[$col][$row] = $val;
      }
    }
  display($transData);
}
function mul($a,$b)
{
```

```
$c = array();

for($i=0;$i<count($a);$i++)
{
   for($j=0;$j<count($b);$j++)
   {
       $c[$i][$j]=0;

       for($k=0;$k<count($a);$k++)
       {

          $c[$i][$j]=$c[$i][$j]+($a[$i][$k]*$b[$k][$j]);
       }
   }
}
  display($c);
}
function add($a,$b)
{
  $c = array();
  for($i=0; $i<count($a); $i++)
  {
    for($j=0; $j<count($b); $j++)
    {
       $c[$i][$j] = $a[$i][$j] + $b[$i][$j];
    }
  }
  display($c);
}
function display($data)
{
 foreach ($data as $value)
   {
      for($i=0;$i<count($value);$i++)
      {
        echo $value[$i]."\t";
      }
      echo "<br>";
   }
}
$a = array(
  array(1,2,3),
  array(4,5,6),
  array(7,8,9),
  );
$b = array(
  array(1,2,3),
  array(4,5,6),
  array(7,8,9),
```

```
 );
echo "Matrix 1 <br>";
display($a);
echo "Matrix 2 <br>";
display($b);
echo "Transpose of Matrix 1<br>";
transpose($a);
echo "Transpose of Matrix 2<br>";
transpose($b);
echo "Multiplication of two matrices<br>";
mul($a,$b);
echo "Addition of two matrices<br>";
add($a,$b);
?>
```

Output:



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

a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.

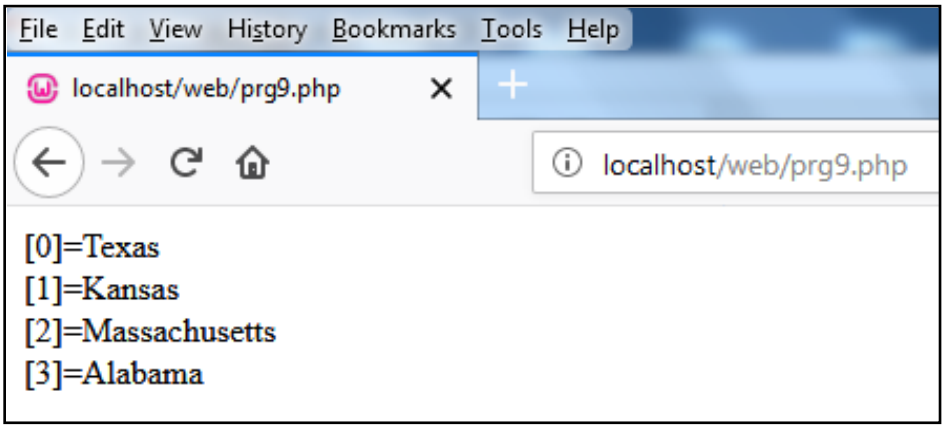
b. Search for a word in states that begins with k and ends in s. Perform a case insensitive comparison. [Note: Passing re.Ias a second parameter to method compile performs a case - insensitive comparison.] Store this word in element1 of statesList.

c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.

d. Search for a word in states that ends in a. Store this word in element 3 of the list.

```
<?php

 $states = "Mississippi Alabama Texas Massachusetts Kansas";
 $my_array  = preg_split("/ /", $states);

 for($i=0;$i<count($my_array);$i++)
 {
  if (preg_match("/xas/", $my_array[$i]))
  {
     $statesList[0] = $my_array[$i];
  }

  if (preg_match("/^k.*s$/i", $my_array[$i]))
  {
     $statesList[1] = $my_array[$i];
  }

  if (preg_match("/^M.*s$/", $my_array[$i]))
  {
     $statesList[2] = $my_array[$i];
  }

  if (preg_match("/a$/", $my_array[$i]))
  {
     $statesList[3] = $my_array[$i];
  }
 }

 for($i=0;$i<count($statesList);$i++)
 {
  echo "[".$i."]=" .$statesList[$i]."<br>";

 }
?>
```

Output:

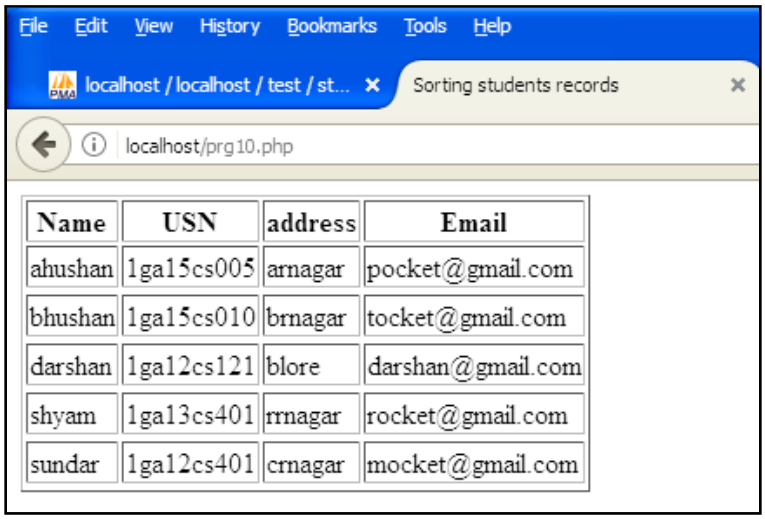

File Edit View History Bookmarks Tools Help
localhost/web/prg9.php
localhost/web/prg9.php
[0]=Texas
[1]=Kansas
[2]=Massachusetts
[3]=Alabama

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

```
<html>
<head>
  <title> Sorting students records </title>
</head>
<body>
<?php
function selectionSort($array) {
        $length = count($array);
        for ($j = 0; $j < $length-1; $j++) {
                $iMin = $j;
                for ($i = $j+1; $i < $length; $i++) {
                        if ($array[$i] < $array[$iMin]) {
                                $iMin = $i;
                        }
                }
                if ($iMin != $j) {
                        // swap
                        $temp = $array[$j];
                        $array[$j] = $array[$iMin];
                        $array[$iMin] = $temp;
                }
        }
        return $array;
}

$con=@mysql_connect('127.0.0.1','root','');
mysql_select_db('test');
$sql="select * from student";
$result=mysql_query($sql) or die("No such user: " . mysql_error());
if (mysql_num_rows($result) == 0)
{
echo "No matching records!!!!";
}
$i=0;
while($row=mysql_fetch_row($result))
{

  $n[$i] = $row[0];
  $usn[$i] = $row[1];
  $add[$i] = $row[2];
  $email[$i] = $row[3];
  $i = $i+1;

}
$a = selectionSort($n);
echo "<table border>
```

```
      <th> Name </th>
      <th> USN </th>
      <th> address </th>
      <th> Email </th>";

for ($j = 0; $j < count($a); $j++)
{
  for ($k = 0; $k < count($a); $k++)
  {

    if($a[$j] == $n[$k])
     {
      echo "<tr>
            <td>$n[$k]</td>
            <td>$usn[$k]</td>
            <td>$add[$k]</td>
            <td>$email[$k]</td>
          <tr>";
    }
  }
}
echo"</table>";
?>
</body>
</html>
```

Output:



Name	USN	address	Email
ahushan	1ga15cs005	arnagar	pocket@gmail.com
bhushan	1ga15cs010	brnagar	tocket@gmail.com
darshan	1ga12cs121	blore	darshan@gmail.com
shyam	1ga13cs401	rrnagar	rocket@gmail.com
sundar	1ga12cs401	crnagar	mocket@gmail.com

CHAPTER 9

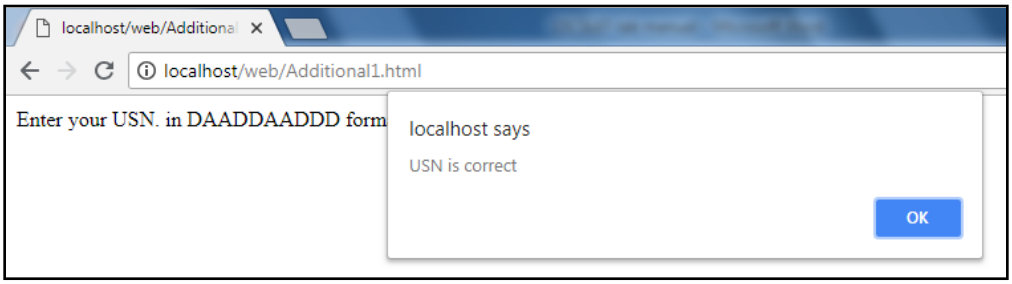
Additional Programs

1. 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.

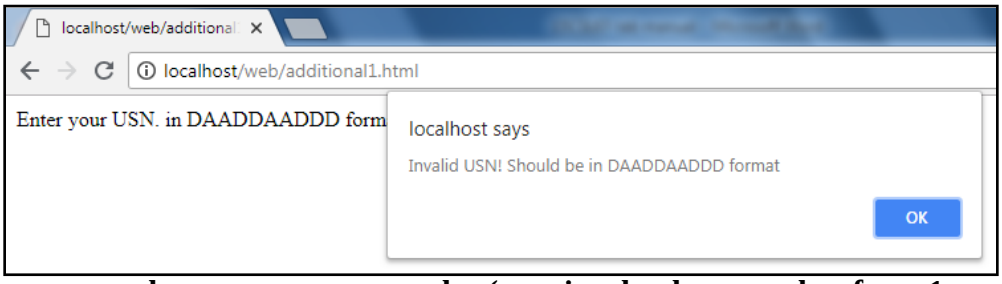
```
<?xml version = "1.0" encoding = "utf-8" ?>
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<body>
<script type="text/javascript">
function validator()
{
var usn = document.getElementById("req1");
var exp1=/^[1-4][A-Z][A-Z][0-9][0-9][A-Z][A-Z][0-9][0-9][0-9]$/i;
if(usn.value.length == 0)
{
 alert("USN is empty!");
return false;
}
else if(!usn.value.match(exp1))
{
alert("Invalid USN! Should be in DAADDAADDD format");
return false;
}
alert("USN is correct");
return true;
}
</script>
 <form action=" ">
 Enter your USN. in DAADDAADDD format :
<input type="text" id="req1" size="10" maxlength="10" />
<input type="button" value="Submit" onclick="validator()" />
</form>
</body>
</html>
```

Output:

Run 1:



Run 2:

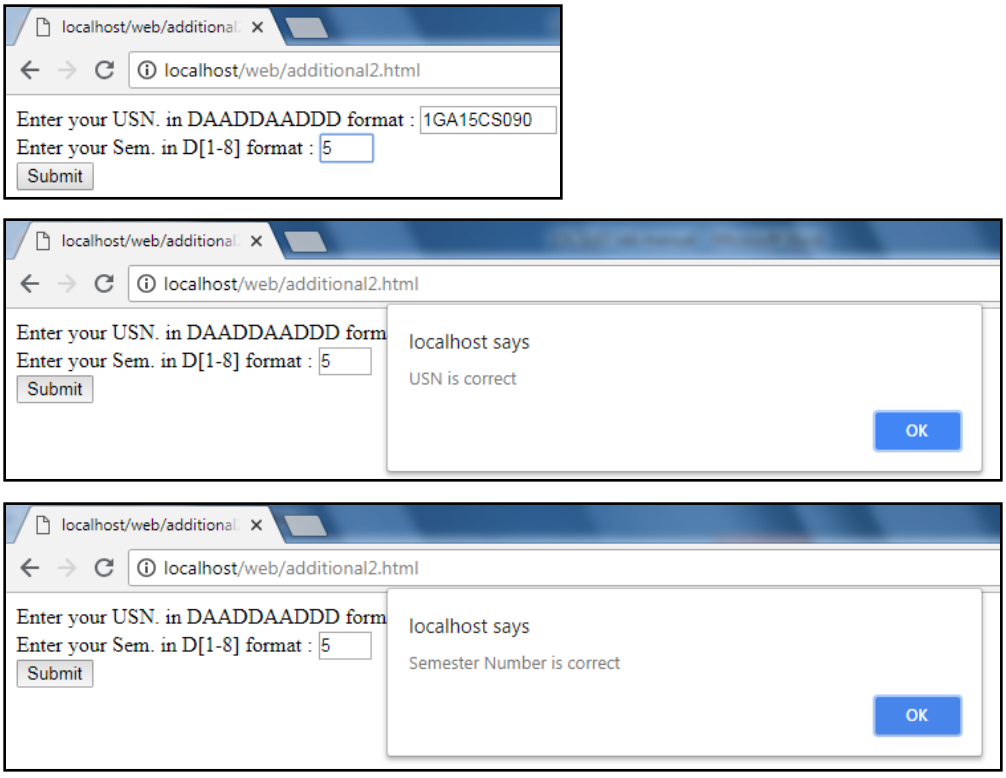


2. Modify the above program to get the current semester also (restricted to be a number from 1 to 8)

```
<?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>
<script type="text/javascript">
 function validator()
{ var usn = document.getElementById("req1");
var sem = document.getElementById("req2");
var exp1 = /^[1-4][A-Z][A-Z][0-9][0-9][A-Z][A-Z][0-9][0-9][0-9]$/i ;
if(usn.value.length == 0 )
{
alert("USN is empty!");
}
else if(!usn.value.match(exp1))
{
alert("Invalid USN! Should be in DAADDAADDD format");
```

```
}
else alert("USN is correct");
var exp2 = /^[1-8]$/;
if(sem.value.length == 0)
{
alert("Semester Number is empty!");
return false;
}
else if(!sem.value.match(exp2))
{
alert("Invalid Semester Number! Should be between 1-8");
return false;
}
alert("Semester Number is correct");
return true;
 }
</script>
<form action=" ">
Enter your USN. in DAADDAADDD format :
<input type="text" id="req1" size="10" maxlength="10" /><br />
Enter your Sem. in D[1-8] format :
<input type="text" id="req2" size="1" maxlength="1" /><br />
<input type="button" value="Submit" onclick="validator()" />
</form>
</body>
</html>
```

Output:

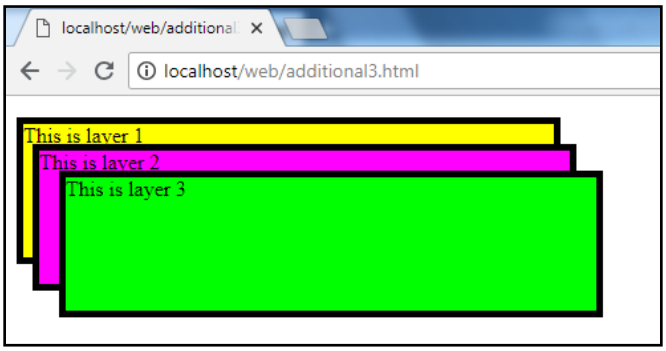


2. Develop and demonstrate, using Javascript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph,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

```
<?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>
<style type="text/css">
.layer1
{
    position: absolute;
    top:0px;
    left:0px
    z-index:0;
    width:400px;
    height:100px;
    overflow:hidden;
    border:solid thick black;
    background-color:yellow;
 }
.layer2
{
     position: absolute;
    top:20px;
     left:20px;
    z-index:1;
    width:400px;
    height:100px;
    overflow:hidden;
    border:solid thick black;
    background-color:magenta;
}
.layer3
{
    position: absolute;
     top:40px;
    left:40px;
    z-index:2;
    width:400px;
    height:100px;
    overflow:hidden;
    border:solid thick black;
    background-color:lime;
 }
</style>
```

```
<script type="text/javascript">
 function totop(newtop)
{
  newdoc=document.getElementById(newtop);
  one=document.getElementById("layer1").style.zIndex;
  two=document.getElementById("layer2").style.zIndex;
  three=document.getElementById("layer3").style.zIndex;
  newmax=Math.max(one,two,three);
   newdoc.style.zIndex=newmax+1;
}
</script>
<p class ="layer1" id="layer1" onmouseover=totop("layer1")> This is layer 1</p>
<p class ="layer2" id="layer2" onmouseover=totop("layer2")> This is layer 2 </p>
<p class ="layer3" id="layer3" onmouseover=totop("layer3")> This is layer 3</p></body>
</body>
</html>
```

Output:

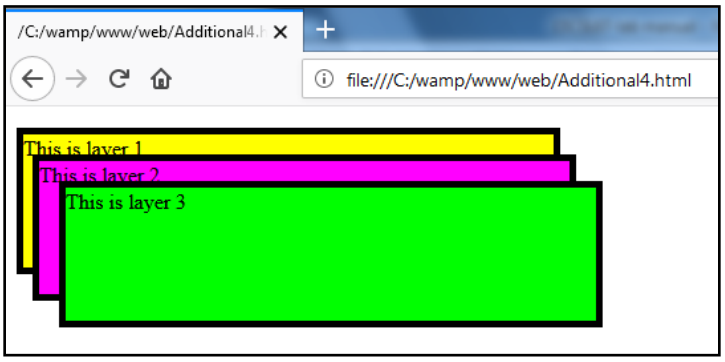


4. Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom.

```
<?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>
 <style type="text/css">
.layer1
{
  position: absolute;
  top:0px;
  left:0px
  z-index:0;
  width:400px;
  height:100px;
  overflow:hidden;
  border:solid thick black;
  background-color:yellow;
}
 .layer2
{
  position: absolute;
  top:20px;
  left:20px;
  z-index:1;
  width:400px;
  height:100px;
  overflow:hidden;
  border:solid thick black;
  background-color:magenta;
}
.layer3
{
  position: absolute;
  top:40px;
  left:40px;
  z-index:2;
  width:400px;
  height:100px;
  overflow:hidden;
  border:solid thick black;
  background-color:lime;
}
</style>
<script type="text/javascript">
 function totop(newtop)
{
```

```
  newdoc=document.getElementById(newtop);
  one=document.getElementById("layer1").style.zIndex;
  two=document.getElementById("layer2").style.zIndex;
  three=document.getElementById("layer3").style.zIndex;
  newmax=Math.max(one,two,three);
  newdoc.style.zIndex=newmax+1;
}
function resetvalue()
{
  document.getElementById("layer1").style.zIndex=0;
  document.getElementById("layer2").style.zIndex=1;
  document.getElementById("layer3").style.zIndex=2;
}
</script>
<p class ="layer1" id="layer1" onmouseover=totop("layer1") onmouseout="resetvalue()">This is layer 1 </p>
<p class ="layer2" id="layer2" onmouseover=totop("layer2") onmouseout="resetvalue()">This is layer 2 </p>
<p class ="layer3" id="layer3" onmouseover=totop("layer3") onmouseout="resetvalue()">This is layer 3 </p>
</body>
</html>
```

Output:

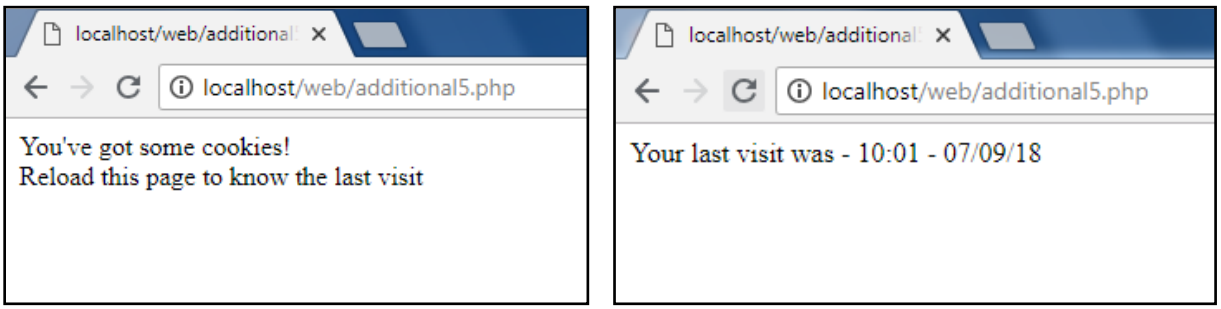


When the cursor is on a block it comes on top and resets to its position when the cursor is moved out

5. 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.

```
<?php
date_default_timezone_set("Asia/Calcutta");
setcookie("visit", date("G:i - m/d/y"));
if(isset($_COOKIE["visit"]))
{
  $visit = $_COOKIE["visit"];
  echo "Your last visit was - ". $visit;
}
else
echo "You've got some cookies!<br />Reload this page to know the last visit";
?>
```

Output:



CHAPTER 10

Viva Questions

1. **Internet :** Network of networks.
2. **Web :** A vast collection of documents, some of which are connected by links.
3. **Difference between Internet and Web :*Internet*** connects computers. ***Web*** connects people through Internet.
4. **Web Browser :** Program used to view HTML documents.
5. **Web Server :** Program that provides documents to the requesting browsers.
6. **Domain Name :** Unique name that identifies an internet resource such as a website.
7. **XHTML :**Xtensible Hyper Text Markup Language
8. **CSS :**Cascading Style Sheets
9. **XML :**Xtensible Markup Language
10. **XSLT :**Extensible Stylesheet Language Transformation
11. **WAMP :** Windows Apache MySQL PHP
12. **UTF :** Universal character Set Transformation Format.
13. **Use of UTF :** It is an industry standard for character encoding of text documents.
14. **Use of !DOCTYPE:** Usually a <!DOCTYPE> declaration is used to distinguish between versions of HTMLish languages
15. **xmlns :** The xmlns attribute specifies the xml namespace for a document
16. **Father of web :** Tim Berners-Lee
17. **127.0.0.1 (localhost) :** It is the IP of WAMP server
18. **Difference between cell padding and cell spacing :** Cell Padding is used to specify the spacing between the contents of a cell and the inner walls of the cell. Cell Spacing is used to specify the distance between cells in a table.
19. **What is the use of Javascript?**

    It is a scripting language for input, output operations, processing of the expressions depending upon expression values, iterations of the values.
20. **Explain alert, confirm and prompt.**

    alert : It is used to print the text or literal or value of a variable in a dialog window. It also displays an OK button. **confirm :** It returns a Boolean value that indicates the user‟s input : true for OK and false for Cancel. **prompt :** It is used to input a value in JS.

21. **Explain document.write() function.**

    It is used to display the string or literal or value of a variable on the webpage. The HTML tags and the values can be concatenated using + operator.

22. **What is a regular expression?**

    A Regular Expression is a sequence of characters that forms a search pattern.

23. **What are position anchors?**

    The leading ^ and the trailing $ are known as position anchors, which match the beginning and ending of the input string, respectively.

24. **Explain document.getElementById() function.**

    The getElementById() method accesses the first element with the specified id.

25. **Give the syntax of function definition in JavaScript**

    function function_name (parameters) { … }

26. **List out the selectors. Which selector does start with .symbol? Which selector with # symbol?**

    Types of selectors : Simple selector forms, Class selectors, Generic selectors, Id selectors, Universal selectors, Pseudo classes Generic selector starts with . id selector starts with #

27. **What is meant by position:absolute?**

    The element is positioned relative to the first parent element that has a position other than static.

28. **What is the difference between absolute and relative position?**

    absolute : The element is positioned relative to its first positioned (not static) ancestor element. **relative :** The element is positioned relative to its normal position.

29. **What is meant by z-index property?**

    The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).An element can have a positive or negative stack order. An element with greater stack order is always in front of an element with a lower stack order.

30. **Can we set z-index with –ve values? If so, what happens?**

    Yes. The specified image or paragraph appears behind the other images or paragraphs.

31. **What is overflow:hidden property?**

    The overflow property specifies what to do if the content of an element exceeds the size of the element's box.

32. **What is the difference between hidden and scroll properties?**

    hidden : The overflow is clipped, and the rest of the content will be invisible. **scroll :** The overflow is clipped, but a scroll-bar is added to see the rest of the content.

33. **What is border property?**

    It is used to set Border style, Border width and Border color.

34. **What is ?**

    It is a CSS tag used to apply styles for a part of the paragraph rather than the whole document.

35. **What is a session in PHP? How do you start a session? What is its use?**

    A session is a way to store information (in variables) to be used across multiple pages. Unlike a cookie, the information is not stored on the users computer.

36. **What is $_SESSION?**

    Session variables are set with the PHP global variable: $_SESSION.

37. **What is $_SERVER in php?**

    $_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

38. **What is $_SERVER['REQUEST_METHOD']?**

    Returns the request method used to access the page (such as POST).

39. **What is $_POST?**

    It is an array of variables passed to the current script via the HTTP POST method.

40. **What is mysql_query() ?**

    It executes a query on a MySQL database.

41. **What is mysql_num_rows() function?**

    It returns the number of rows in a record set.

42. **What is mysql_fetch_row() in php?**

    It returns a row from a record set as a numeric array.

43. **What is mysql_free_result() ?**

    It frees memory used by a result handle