

CSE3002-INTERNET AND WEB PROGRAMMING

LAB ASSIGNMENT-8,9,11

NAME: YOGESWARI SAHU

REGISTRATION NUMBER: 18BCE0928

FACULTY: JAYAKUMAR K

SLOT: L13+L14

LAB ASSIGNMENT-8

QUESTION:

8. Write a PHP program to demonstrate the concept of File handling

i. Write a PHP program to write 100 integers in to a text file. Read 10 numbers at a time from the file using PHP script and find the numbers which occurs odd number of times.

Input : 4, 5, 4, 5, 2, 2, 3, 3, 2, 4

Output : 2, 4

PROCEDURE:

The tags and attributes used in HTML:

<html>-

Root element of an HTML page(contains everything in the html)

<head> -

Tells information about the document and contains any links

<title> -

Gives the topic/name of the document(when we show it on browser the name of the tab is that name)

<body>-

Contains the body, i.e. the content of the body

<meta>-

Lets your document be responsive

PHP-

script starts with <?php and ends with ?>

Built in Functions:

echo -

used to output data to the screen.

array() -

Creates an array. An array stores multiple values in one single variable.

array_push()-

Inserts one or more elements to the end of an array

fopen -

Opens a file or URL

fwrite-

Writes to an open file (binary-safe)

fclose-

Closes an open file

feof-

Checks if the "end-of-file" (EOF) has been reached for an open file

fgetc-

Returns a single character from an open file

file_get_contents -

Reads a file into a string

str_replace-

Replaces some characters in a string (case-sensitive)

str_split-

Splits a string into an array

Loops used:

for-

loops through a block of code a specified number of times

foreach -

loops through a block of code for each element in an array

while-

loops through a block of code as long as the specified condition is true

Algorithm/Procedure:

- 1) Made a file test.txt
- 2) Wrote random numbers arranging from 1-100 in the file and then close the file
- 3) We open the file again to perform the main operation and then we retrieve all the content of the file in the form of string and we split it

with space as a delimiter and convert it into an array using
file_get_contents function and str_split

- 4) We then iterate over every ten elements of the array and find the numbers which occurs odd number of times using loops and if condition
- 5) Finally we print the result on the screen
- 6) At the end, we close the file

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>18BCE0928 YOGESWARI SAHU</title>
</head>
<body>
    <?php
    echo "18BCE0928 YOGESWARI SAHU <br>";
    $file = fopen("test.txt","w");
    for($n=1;$n<=100;$n++){
        fwrite( $file,rand(0,9) . "\t");
    }
    fclose($file);
    $fp = fopen("test.txt", "r");
    echo "The contents in the file are:";
    while(!feof($fp)) {
        $c=fgetc($fp);
        echo $c;
    }
    echo "<br>";
    $file_chars = file_get_contents('test.txt');
    $newstr1=str_replace("\t","", $file_chars);
    // echo $newstr1;
    $newstr2=str_split($newstr1);
    $counts = array();
    for ($x = 0; $x <= 9; $x++) {
        echo "<br>";
        $temp=array();
        for ($y = 0; $y <= 9; $y++){
            array_push($temp,$newstr2[$x*10+$y]);
        }
        echo "<b>The 10 characters input are:</b> ";
        foreach($temp as $value){
            echo $value . "\t";
```

```

    }

    $ctemp=array();
    $chk=0;
    for ($i = 0; $i<10; $i++)
    {
        $count = 0;
        for ($j = 0; $j<10; $j++)
        {
            if ($temp[$i] == $temp[$j])
                $count++;
        }
        if ($count % 2 != 0){
            for($j=0;$j<count($ctemp);$j++)
            {
                if($ctemp[$j]==$temp[$i])
                    $chk=1;
            }
            if($chk==0)
                array_push($ctemp,$temp[$i]);
        }
    }
    echo "<br>";
    echo "<b>Output:</b> ";
    foreach ($ctemp as $value)
    {
        echo $value . "\t";
    }
    echo"<br>";

}
fclose($fp);

?>
</body>
</html>

```

OUTPUT:

```
18BCE0928 YOGESWARI SAHU x +
localhost/wpda8/
18BCE0928 YOGESWARI SAHU
The contents in the file are:6108011446004583116573388180713688679502877454090754722870777503119449959882140296861276668542857933
The 10 characters input are: 6108011446
Output: 18
The 10 characters input are: 0045831165
Output: 4836
The 10 characters input are: 7338818071
Output: 8
The 10 characters input are: 3688679502
Output: 379502
The 10 characters input are: 8774540907
Output: 87
The 10 characters input are: 5472287077
Output: 5480
The 10 characters input are: 7503119449
Output: 7503
The 10 characters input are: 9598821402
Output: 5140
The 10 characters input are: 9686127666
Output: 968
The 10 characters input are: 8542857933
Output: 4279
```

LAB ASSIGNMENT-9

QUESTION:

Write a PHP program to demonstrate the concept of Session Management and Cookies

- a. Create a cookie with a value of username and his emailId. The cookie should expire in 3 days. Check whether the cookie is present in the website, when the user enters his username, the emailed should be given by the cookies.
- b. Perform the following operations:
 - i. Create a cookie.Check whether the cookie is enabled or not. Print the status.
 - ii. Delete the created cookie before an hour
 - iii. Check whether the cookie is disabled or not. Print the status

PROCEDURE:

The tags and attributes used:

<html>-

Root element of an HTML page(contains everything in the html)

<body>-

Contains the body, i.e. the content of the body

PHP-

script starts with <?php and ends with ?>

Built in Functions:

echo -

used to output data to the screen.

setcookie() -

A cookie is created with the setcookie() function.

isset -

checks whether a variable is set, which means that it has to be declared and is not NULL

Answers:

a) Create a cookie with a value of username and his emailId. The cookie should expire in 3 days. Check whether the cookie is present in the website,when the user enters his username, the emailed should be given by the cookies.

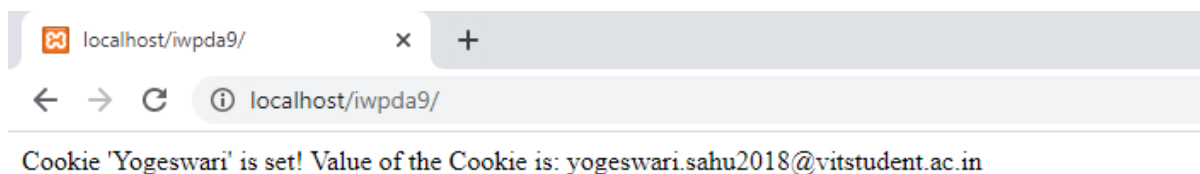
Algorithm/Procedure:

- 1) Create a cookie named "Yogeswari" with the value "yogeswari.sahu2018@vitstudent.ac.in" using setcookie().
- 2) The cookie will expire after 3 days as given by adding $86400 * 3$ secs to the current time.
- 3) We then retrieve the value of the cookie "Yogeswari" (using the global variable `$_COOKIE`). We also use the `isset()` function to find out if the cookie is set.
- 4) If it is, it will show the cookie name and its value i.e. "Yogeswari" with the value "yogeswari.sahu2018@vitstudent.ac.in".

CODE :

```
<!DOCTYPE html>
<html>
<body>
<?php
$cookie_name = "Yogeswari";
$cookie_value = "yogeswari.sahu2018@vitstudent.ac.in";
setcookie($cookie_name, $cookie_value, time() + (86400 * 3));
?>
<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set! \n ";
    echo "Value of the Cookie is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

OUTPUT:



b)

i) . Create a cookie. Check whether the cookie is enabled or not. Print the status.

Algorithm/Procedure:

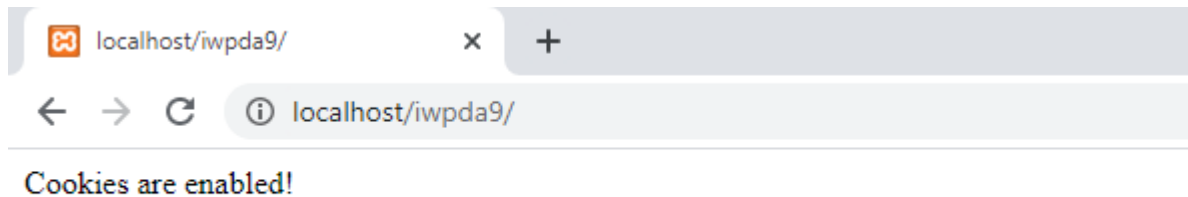
- 1) Create a cookie named " sample_cookie " with the value " sample_value " using setcookie().
- 2) The cookie will expire after 1 days as given by adding $86400 * 1$ secs to the current time.
- 3) We then see if the cookies are enabled using count(\$_COOKIE)

CODE :

```
<!DOCTYPE html>
<html>
<body>
//i)To create a sample cookie and check whether it is enable or not
<?php
    setcookie("sample_cookie", "sample_value", time() + (86400 * 1));
    if (count($_COOKIE) > 0) {
        echo "Cookies are enabled!";
    } else {
        echo "Cookies are disabled!";
    }
?>

</body>
</html>
```

OUTPUT:



ii. Delete the created cookie before an hour

iii. Check whether the cookie is disabled or not. Print the status

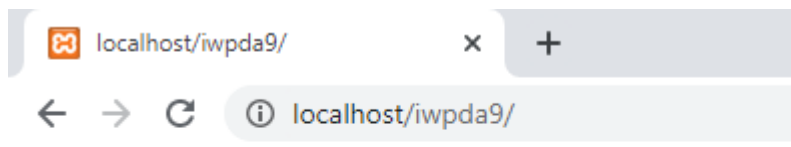
Algorithm/Procedure:

- 1) 1) To delete a cookie, use the `setcookie()` function with an expiration date in the past
- 2) We then see if the cookies are enabled or disabled using `count($_COOKIE)`

CODE :

```
<!DOCTYPE html>
<html>
<body>
<?php
    setcookie("sample-cookie", "", time() - 3600);
?>
<?php
    if (count($_COOKIE) > 0) {
        echo "Cookies are enabled!";
    } else {
        echo "Cookies are disabled!";
    }
?>
</body>
</html>
```

OUTPUT:



LAB ASSIGNMENT-11

QUESTION:

Write a program to demonstrate the concept of data storage and parsing in XML

- i. Develop a thesaurus tool by creating a schema for thesaurus. When a word is entered the synonyms or antonyms must be displayed based on the user request.
- ii. XSLT - Create a student mark maintenance system using XML. Create a webpage to display all the students consolidated mark statement with pass (green color) or fail (red color) using XSLT.

PROCEDURE:

The tags and attributes used:

<html>-

Root element of an HTML page(contains everything in the html)

<head> -

Tells information about the document and contains any links

<title> -

Gives the topic/name of the document(when we show it on browser the name of the tab is that name)

<body>-

Contains the body, i.e. the content of the body

<meta>-

Lets your document be responsive

<style>-

Defines style information for a document

Properties:

margin properties are used to create space around elements, outside of any defined borders.

padding properties are used to generate space around an element's content, inside of any defined borders.

background-color property specifies the background color of an element.

float property specifies how an element should float.

<h1>,<h2>,<h3>,<h4>,<h5>,<h6>-

The different sizes of headings we can use in which h1 is the largest and h6 is the smallest

<div>-

Converts the document into sections to which we can give specific classes or ids

Attributes:

style- The style attribute is used to add styles to an element, such as color, font, size, and more.

class- used to specify a class for an HTML element.

id- used to specify a unique id for an HTML element.

<script>-

used to embed a client-side script (JavaScript).

<p>-

defines a paragraph.

<button>-

defines a clickable button.

DOM-

document.getElementById(id)- To Find an element by element id

element.innerHTML = new html content- Used to Change the inner HTML of an element

XMLHttpRequest object-

can be used to request data from a web server.

onreadystatechange property-

specifies a function to be executed every time the status of the XMLHttpRequest object changes.

responseXML-

Returns the response data as XML data

Algorithm/Procedure:

- 1) Assign a input tag where the user can give the word for which the synonyms or antonyms must be displayed based on the user request.
- 2) Assign a paragraph tag with a demo id and button when clicked the function search is fired.
- 3) A XMLHttpRequest object is assigned to variable xhttp
- 4) When readyState property is 4 and the status property is 200, the response is ready, responseXML returns the response data as XML data
- 5) When the server response is ready, an HTML table is built, nodes (elements) are extracted from the XML file
- 6) We finally update the element "demo" with the HTML table filled with XML data.

i)

CODE :

index.html

```
<!doctype html>

<html>

    <head>

        <title>18BCE0928</title>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-
scale=1.0">

        <style>

            #lt {

                float:left;

                margin: 0px 30px 0px 0px;
```

```

    }

    #rt {

        float:left;

        margin: 0px 30px 0px 0px;

    }

</style>
</head>
<body>

    <h1>Thesaurus</h1>

    <h2>Yogeswari Sahu 18BCE0928</h2>

    <input type="text" required id="txt" placeholder="Enter word">

    <button id="btn" onclick="search()">Search</button>

    <p id="demo"></p>

    <div id="lt"></div>

    <div id="rt"></div>

    <script>

        function search() {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {

        var xmlDoc = this.responseXML;

        var txt=document.getElementById("txt").value.toLowerCase();

        var i,j;

        var synonym="";

        var antonym="";

        var flag="0";

        var x=xmlDoc.getElementsByTagName("entry");

```

```

for(i=0;i<x.length;i++){
    if(x[i].children[0].childNodes[0].nodeValue==txt){
        for(j=0;j<x[i].children[1].children.length;j++){
            synonym +=x[i].children[1].children[j].childNodes[0].n
odeValue + "<br>";
        }
        for(j=0;j<x[i].children[2].children.length;j++){
            antonym +=x[i].children[2].children[j].childNodes[0].n
odeValue + "<br>";
        }
        flag=1;
        break;
    }
}

if(flag==0){
    document.getElementById("demo").innerHTML = "word was
not found in the thesaurus";

    document.getElementById("lt").innerHTML="";
    document.getElementById("rt").innerHTML="";
}else{
    document.getElementById("demo").innerHTML="";
    document.getElementById("lt").innerHTML = "synonyms:<br>" + synonym +
"</br>";
    document.getElementById("rt").innerHTML = "antonyms:<br>" + antonym +
"</br>";
}
}

};

xhttp.open("GET", "thesaurus.xml", true);

```



```
xhttp.send();  
}  
  
    </script>  
  
    </body>  
  
</html>
```

thesaurus.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<!DOCTYPE thesarus SYSTEM "thesarus.dtd">  
  
<thesaurus>  
  
    <entry>  
  
        <word>bad</word>  
  
        <synonym>  
  
            <word>awful</word>  
  
            <word>atrocious</word>  
  
            <word>cheap</word>  
  
            <word>unacceptable</word>  
  
        </synonym>  
  
        <antonym>  
  
            <word>good</word>  
  
            <word>fortunate</word>  
  
            <word>ok</word>  
  
        </antonym>  
  
    </entry>  
  
    <entry>  
  
        <word>good</word>  
  
        <synonym>
```

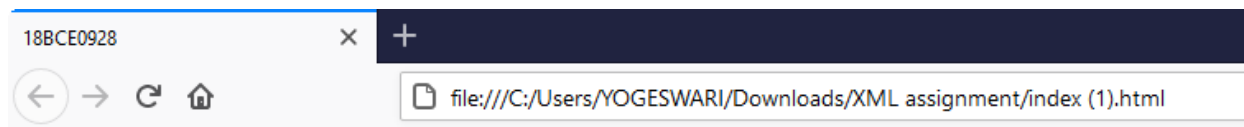
```
<word>gain</word>
<word>favour</word>
<word>boon</word>
<word>prosperity</word>
</synonym>
<antonym>
  <word>evil</word>
  <word>loss</word>
</antonym>
</entry>
<entry>
  <word>new</word>
  <synonym>
    <word>advanced</word>
    <word>current</word>
    <word>different</word>
    <word>recent</word>
  </synonym>
  <antonym>
    <word>antiquated</word>
    <word>familiar</word>
    <word>common</word>
  </antonym>
</entry>
<entry>
  <word>Successful</word>
```

```
<synonym>
    <word>Propitious</word>
    <word>Felicitous</word>
</synonym>
<antonym>
    <word>Destitute</word>
    <word>Untoward</word>
</antonym>
</entry>
</thesaurus>
```

thesaurus.dtd

```
<!ELEMENT thesaurus (entry)*>
<!ELEMENT entry (word|synonym|antonym)*>
<!ELEMENT word (#PCDATA)>
<!ELEMENT synonym (word)*>
<!ELEMENT antonym (word)*>
```

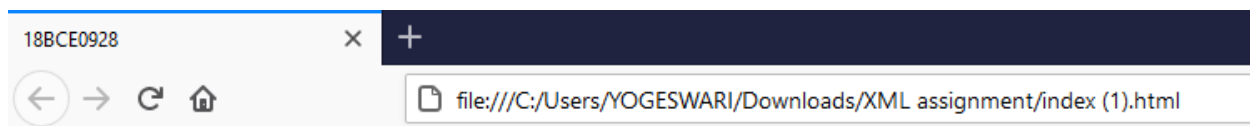
OUTPUT:



Thesaurus

Yogeswari Sahu 18BCE0928

synonyms: antonyms:
gain evil
favour loss
boon
prosperity



Thesaurus

Yogeswari Sahu 18BCE0928

Word was not found in the thesaurus

ii)

Tags/Elements used:

<xsl:template> element -

used to build templates.

match attribute-

used to associate a template with an XML element. The match attribute can also be used to define a template for the entire XML document. The value of the match attribute is an XPath expression (i.e. match="/" defines the whole document).

<html>-

Root element of an HTML page(contains everything in the html)

<body>-

Contains the body, i.e. the content of the body

<xsl:value-of> element-

select values from the XML elements

<xsl:for-each> element-

can be used to select every XML element of a specified node-set

<xsl:if> element-

used to put a conditional test against the content of the XML file

CODE :

students.xml

```
<?xml version = "1.0"?>
```

```
<?xml-stylesheet type = "text/xsl" href = "students.xsl"?>
```

```
<class>
```

```
  <student rollno = "393">
```

```
    <firstname>Ayush</firstname>
```

```
    <lastname>Chakaldar</lastname>
```

```
    <marks>85</marks>
```

```
  </student>
```

```

<student rollno = "493">
    <firstname>Yogeswari</firstname>
    <lastname>Sahu</lastname>
    <marks>95</marks>
</student>
<student rollno = "593">
    <firstname>Ranveer</firstname>
    <lastname>Singh</lastname>
    <marks>90</marks>
</student>
<student rollno = "800">
    <firstname>Rishi</firstname>
    <lastname>Raghu</lastname>
    <marks>38</marks>
</student>
</class>

```

students.xsl

```

<?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>
            <body>
                <h2>Students</h2>
                <h2>Yogeswari Sahu 18BCE0928</h2>
                <table border = "1">
                    <tr bgcolor = "#aacd32">

```

```

<th>Roll No</th>

<th>First Name</th>

<th>Last Name</th>

<th>Marks</th>

<th>Pass/Fail</th>
</tr>
<xsl:for-each select="class/student">
  <xsl:if test="marks < 40">
    <tr bgcolor="red">
      <td>
        <xsl:value-of select = "@rollno"/>
      </td>

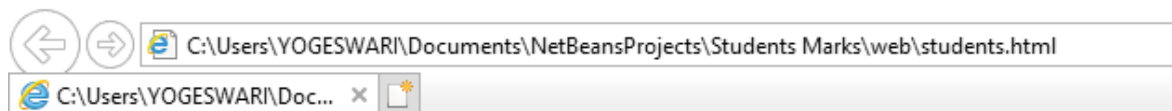
      <td><xsl:value-of select = "firstname"/></td>
      <td><xsl:value-of select = "lastname"/></td>
      <td><xsl:value-of select = "marks"/></td>
      <td>Fail</td>
    </tr>
  </xsl:if>
  <xsl:if test="marks > 40">
    <tr bgcolor="green">
      <td>
        <xsl:value-of select = "@rollno"/>
      </td>

      <td><xsl:value-of select = "firstname"/></td>
      <td><xsl:value-of select = "lastname"/></td>
      <td><xsl:value-of select = "marks"/></td>

```

```
 Pass |
```

OUTPUT:



Students

Yogeswari Sahu 18BCE0928

Roll No	First Name	Last Name	Marks	Pass/Fail
393	Ayush	Chakaldar	85	Pass
493	Yogeswari	Sahu	95	Pass
593	Ranveer	Singh	90	Pass
800	Rishi	Raghu	38	Fail