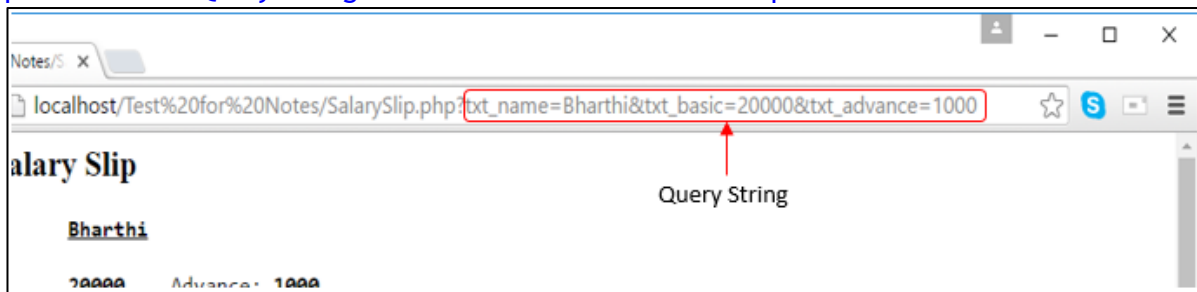




**PHP-QUESTION BANK**

**Question-Answer  
PHP-6: Working with FORM**

1. a. Explain the working of QueryString using an example.  
When a form is submitted using GET method, the form’s data is sent to the server in the form of “variable=value” pairs. A variable and its value are separated by equality sign (=) and different “variable=value” pairs are separated by ampersand sign (&). This complete set of “variable=value” pairs is called Query String and is visible in the URL. Example:



The query string is specified by the values following the question mark (?).  
When a form is submitted by GET method, this query string is visible in the URL. A PHP script retrieves this string in `$_SERVER["QUERY_STRING"]`.  
Query Strings are used to pass information from a browser to a web server. Several different processes can generate a query string. Query strings are generated by sending a form, by a user typing a query in the address box of the browser or in the anchor tag.

- b. Differentiate between  
(i) GET and POST methods of sending data

GET	POST
Data sent by GET method is restricted to 1024 characters.	In the POST method there is no restriction on data size to be sent.
Data is visible in the URL while being sent to the server. Therefore, GET method should not be used to send sensitive data like passwords etc.	Data is invisible as it is embedded within the body of HTTP request. Hence, data being sent is secure.
GET cannot be used to send binary data, like images or word documents, to the server.	POST can be used to send ASCII as well as binary data.

- (ii) `$_GET` and `$_POST` variables.

`$_GET` receives the data if a form is submitted by GET method, whereas `$_POST` receives the data if a form is submitted by POST method.

## c. Find the error(s) in the following code:

```

<?php
$name = $nameErr = "";
OK=false;
if ($_SERVER("REQUEST_METHOD") == "POST")
{
    $name = $_POST["name"];
    if (!empty($name))
        OK=true;
    else
        nameErr = "Please enter name";
}
?>
<html>
<body>
<form name = form1 action=POST
    method = "<?php echo $_SERVER["PHP_SELF"]; ?>" >
Name <input type = textbox name = "name">
<?php echo $nameErr; >
<br>
<input type = submit name = submit value = "SUBMIT">
</form>
<?php
    if $OK==true
        echo "Hello ".$name."<P>How are you";
?>
</body>
</html>

<?php
$name = $nameErr = "";
$OK=false; // - 1
if ($_SERVER["REQUEST METHOD"] == "POST") // - 2
{
    $name = $_POST["name"];
    if (!empty($name))
        $OK=true; // - 3
    else
        $nameErr = "Please enter name"; // - 4
}
?>
<html>
<body>
<form name = form1 action method=POST // - 5
    method action = "<?php echo $_SERVER["PHP_SELF"]; ?>" > // - 6
Name <input type = textbox name = "name">
<?php echo $nameErr; ?> // - 7
<br>
<input type = submit name = submit value = "SUBMIT">
</form>
<?php
    if ($OK==true) // - 8
        echo "Hello ".$name."<P>How are you";
?>
</body>
</html>

```

2. Create a form which takes the name of the user as input and prints a greeting in the format given below and if the user submits the form with the name field blank, it prints an appropriate error message.

**Hello Francis!**  
**How are you today?**

```
<?php
$name = $nameErr = "";
$OK=false;
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $name = $_POST["name"];
    if (!empty($name))
        $OK=true;
    else
        $nameErr = "Please enter name";
}
?>
<html>
<body>
<form name = form1 method=POST
    action = "<?php echo $_SERVER["PHP_SELF"]; ?>" >
Name <input type = textbox name = "name"> <?php echo $nameErr; ?>
</P>
<input type = submit>
</form>
<?php
    if ($OK)
        echo "Hello ".$name."</BR>How are you";
?>

</body>
</html>
```

3. Create the following form and based on the user selection print a message in the format given below:

Please select your favourite car

Nissan

Toyota

Mitsubishi

**Your favourite car is: Nissan**

```
<?php
$car = $carErr = "";
$OK=false;
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    if (!empty($_POST["Car"]))
    {
        $car = $_POST["Car"];
        $OK=true;
    }
    else
```

```

        $carErr = " **Please select a Car";
    }
    ?>
<html>
<body>
<form name = form1 method=POST
    action = "<?php echo $_SERVER["PHP_SELF"]; ?>" >
Please select your favourite car <?php echo $carErr; ?> <br>
<input type = radio name = Car value = Nissan>Nissan </BR>
<input type = radio name = Car value = Toyota>Toyota </BR>
<input type = radio name = Car value = Mitsubishi>Mitsubishi </BR>
<input type = submit name = submit value = "SUBMIT">
</form>
<?php
    if ($OK==true) // - 6
        echo "Your favourite car is ".$car;
?>

</body>
</html>

```

4. Create the following form and do form validation to ensure that all the fields are filled/selected by the user. When the user submits the form, the message "Order Placed" with the details of the order must be displayed.

### Make Your Pizza

Size of the Pizza

Small 8" -Serves 1 person  
 Medium 12" -Serves 2 persons  
 Large 16" -Serves 4 persons

Select Type

Toppings:  Cheese  Corn  Tomato  Chicken

---

### Contact Details

Name:

Address:

Phone Number(s):

```

<!-- PHP6-FormValidationAns.php -->
<?php
$sizeErr = $typeErr = $topErr = $nameErr = $addressErr = $phoneErr =
"";
$size = $type = $name = $address = $phone = "";
$toppings = array();
$OK = false;
if ($_SERVER["REQUEST_METHOD"] == "POST")
{

```

```
$OK=true;
if (!empty($_POST["size"]))
    $size = $_POST["size"];
else
{
    $sizeErr = "Please select pizza Size";
    $OK = false;
}
if (!empty($_POST["type"]))
    $type = $_POST["type"];
else
{
    $typeErr = "Please select pizza Type";
    $OK = false;
}
if (!empty($_POST["Toppings"]))
    $toppings = $_POST["Toppings"];
else
{
    $topErr = "Please select 1 or more toppings";
    $OK = false;
}
if (!empty($_POST["txtName"]))
    $name = $_POST["txtName"];
else
{
    $nameErr = "Please enter your name";
    $OK = false;
}
if (!empty($_POST["txtAdd"]))
    $address = $_POST["txtAdd"];
else
{
    $addressErr = "Please enter your Address";
    $OK = false;
}
if (!empty($_POST["txtPhone"]))
    $phone = $_POST["txtPhone"];
else
{
    $phoneErr = "Please enter your Phone number(s)";
    $OK = false;
}
}
?>
<html>
<style>
.error {color:RED}
</style>
<body>
<form name = form1 method = "POST" action =
"<?php echo $_SERVER['PHP_SELF'] ?>" >
<h2>Make Your Pizza</h2>
Size of the Pizza <span class = error><?php echo $sizeErr ?></span>
</BR>
<input type = radio name = size value = Small>Small 8" -Serves 1
```

```

person</BR>
<input type = radio name = size value = Medium>Medium 12" -Serves 2
people</BR>
<input type = radio name = size value = Large>Large 16" -Serves 4
people
<P>
Select Type <?php echo $typeErr ?>
<Select name = "type">
<option>Regular</option>
<option>Thin Crust</option>
<option>Double Cheese</option>
</Select>
<P>
Toppings:
<input type = checkbox name = "Toppings[]" value = Cheese>Cheese
<input type = checkbox name = "Toppings[]" value = Corn>Corn
<input type = checkbox name = "Toppings[]" value = Tomato>Tomato
<input type = checkbox name = "Toppings[]" value = Chicken>Chicken
<BR><?php echo $topErr ?>
<HR>
<h2>Contact Details</h2>
<pre>
Name: <input type = textbox name = txtName> <?php echo $nameErr ?><BR>
Address <input type = textbox name = txtAdd><?php echo $addressErr ?><BR>
Phone Number(s): <input type = textbox name = txtPhone> <?php echo
$phoneErr ?><BR><input type = submit name = submit>
</form>
</body></html>
<?php
    if (isset($_POST['submit']) && $OK==true)
    {
    echo "<h2>Thanks for placing the order. Your order details are:</h2>";
    echo $size." ".$type." Pizza with ";
    foreach ($toppings as $top)
        echo $top." ";
    echo "toppings<p>";
    echo "Pizza will be delivered to ".$name." at <br>".$address;
    echo " <br>Phone: ".$phone;
    }
?>

```

5. **What are Superglobals? Give 4 example**  
**Superglobals are built in Variable in PHP that are always available in all scope and we can access them from any functions class or file**  
**Main Superglobals are**  
**\$GLOBALS**  
**\$\_SERVER**  
**\$\_REQUEST**  
**\$\_POST**  
**\$\_GET**  
**\$\_FILES**  
**\$\_COOKIE**  
**\$\_SESSION**  
**\$\_ENV**

6.	<p>Define preg_match() function ? How to use to check \$name contain only letter and space preg_match() function searches for pattern, returning true if the pattern exists and false otherwise.</p> <pre>If(preg_match("/^[a-zA-Z ]*\$/",\$name)) echo " valid"; else echo "Invalid";</pre>
7.	<p>Explain how filter_var() function is used to validate Email When we use FILTER_VALIDATE_EMAIL filter , in filter_var() function it validate whether the value is a valid email address, returning true if it is valid and false otherwise.</p> <pre>If(filter_var(\$email,FILTER_VALIDATE_EMAIL) echo " valid"; else echo "Invalid";</pre>
8.	<p>Explain &lt;fome action= "&lt;?php echo \$_SERVER["PHP_SELF"]?&gt;" statement \$_SERVER["PHP_SELF"] is used to display the path of currently used script file name So the above statement action attributes states that after the form submission form data will be handle by the PHP Script present in the same file.</p>
9.	<p>Explain \$_SERVER["REQUEST_METHOD"] It is used to tell whether data is sent using GET method or POST method when form is submitted</p>

1.	<p><b>Write the command(s) to open a text file called data.txt which exists in the directory C:\temp.</b></p> <pre>\$file = fopen("c:\temp\data.txt", "r");</pre>																											
2.	<p><b>Explain the PHP fopen() function and PHP fclose() functions.</b></p> <p><b>fopen() :</b> fopen() function is used to open a file. The syntax of fopen() is:  <pre>fopen(filename, mode);</pre> fopen() takes two parameters:  i) filename: Specifies the name of the file to be opened.  ii) mode: specifies the mode in which the file should be opened.</p> <p><b>fclose() :</b> fclose() is used to close an open file. fclose() takes a single parameter (the file reference) and returns a Boolean value indicating the successful or failed closure of file.  Syntax: <pre>fclose(file);</pre></p>																											
3.	<p><b>What are different modes for file opening in PHP?</b></p> <p>Different file opening modes in PHP are:</p> <table border="1" data-bbox="231 840 1114 1153"> <thead> <tr> <th>SNo.</th> <th>Mode</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>r</td> <td>'Read only' mode.</td> </tr> <tr> <td>2.</td> <td>r+</td> <td>Read/Write mode.</td> </tr> <tr> <td>3.</td> <td>w</td> <td>Write only mode.</td> </tr> <tr> <td>4.</td> <td>w+</td> <td>Read/Write mode.</td> </tr> <tr> <td>5.</td> <td>a</td> <td>Append mode.</td> </tr> <tr> <td>6.</td> <td>a+</td> <td>Read/Append mode.</td> </tr> <tr> <td>7.</td> <td>x</td> <td>Creates the file and opens it in Write only mode.</td> </tr> <tr> <td>8.</td> <td>x+</td> <td>Creates the file and opens it in Read/Write mode.</td> </tr> </tbody> </table>	SNo.	Mode	Description	1.	r	'Read only' mode.	2.	r+	Read/Write mode.	3.	w	Write only mode.	4.	w+	Read/Write mode.	5.	a	Append mode.	6.	a+	Read/Append mode.	7.	x	Creates the file and opens it in Write only mode.	8.	x+	Creates the file and opens it in Read/Write mode.
SNo.	Mode	Description																										
1.	r	'Read only' mode.																										
2.	r+	Read/Write mode.																										
3.	w	Write only mode.																										
4.	w+	Read/Write mode.																										
5.	a	Append mode.																										
6.	a+	Read/Append mode.																										
7.	x	Creates the file and opens it in Write only mode.																										
8.	x+	Creates the file and opens it in Read/Write mode.																										
4.	<p><b>Explain with an example how to use PHP fopen() function.</b></p> <p>fopen() function is used to open a file in a specific mode.  <b>Example:</b>  <pre>\$file = fopen("sample.txt", "r");</pre> This command opens a text file names "Sample.txt" in 'Read only' mode.</p>																											
5.	<p><b>How does \$_FILES variable work in PHP?</b></p> <p>\$_FILES is a superglobal which stores the details of the uploaded files. The details of the each file include  i) file name,  ii) file type,  iii) file size (in bytes),  iv) temporary location of the file on the server,  v) error code in case some error occurred during of the file upload</p>																											
6.	<p><b>What is the purpose of move_uploaded_file() in PHP?</b></p> <p>When a file is uploaded, it gets stored in a temporary area on the server until it is moved. The file has to be moved from that temporary area, or else it will be destroyed. The function move_uploaded_file() moves an uploaded file to a new location on the server.</p>																											
7.	<p><b>What are PHP cookies? What is the use of a PHP cookie?</b></p> <p>Cookies are small files that the server stores on user's computer when the user visits</p>																											



	<p>a website.</p> <p>Cookies are generally used to identify return visitors, with their approval, so that continuity can be established between visits. Cookies can also be used to keep a user logged into a website indefinitely, track the time of the user's last visit, and much more.</p>
8.	<p><b>Explain with an example how to set a PHP cookie.</b></p> <p>A cookie is created with the <code>setcookie()</code> function. The <code>setcookie()</code> function must be run before any other data is sent to the browser, such as the opening <code>&lt;html&gt;</code> tag or random whitespace.</p> <p><b>Example:</b></p> <pre>&lt;?php     setcookie("Example", "Anamika", time()+60*60); ?&gt;</pre> <p>This code sets a cookie called "Example" that has a value "Anamika". The cookie will expire after one hour (current time + 3600 seconds) of its creation.</p>
9.	<p><b>Explain with an example how to delete a PHP cookie?</b></p> <p>To delete a cookie we need to set the same cookie but with no value and with an expiry date set in the past (i.e., a date that has already expired). This forces the browser to delete the cookie from the user's system.</p> <pre>&lt;?php //set the expiry date to 1 second ago setcookie("Example", "", time()-1); ?&gt;</pre>
10.	<p><b>What is the use of <code>setcookie()</code> function in PHP?</b></p> <p>The <code>setcookie()</code> function is used to create Cookies.</p>
11.	<p><b>What is the importance of sessions used in PHP?</b></p> <p>PHP sessions are of small, temporary files that are stored on the website's server. Sessions can be used to store unique information about, or change unique settings for each visitor, but only for the duration of each separate visit. PHP session allows an application to store information for the current session.</p> <p>The information kept in sessions is secure because it is stored on the server, rather than on the client.</p> <p>Session variables store user information to be used across multiple pages (e.g., username, favourite color etc.). By default session variables last until the user closes the browser.</p>
12.	<p><b>Explain with an example how to register/start a session in PHP.</b></p> <p>A session is started in PHP using <code>Session_start()</code> function. This function must be run on every page, before any other data is sent to the browser, such as the opening <code>&lt;html&gt;</code> tag or random whitespace.</p> <p><b>Example:</b></p> <pre>&lt;?PHP     Session_start();    ?&gt;</pre>
13.	<p><b>Explain with an example how to destroy a session in PHP.</b></p> <p>A session is destroyed by using the <code>session_destroy()</code> function. <code>Session_destry()</code></p>

function ends the session completely, as well as erase all session variables.


Example:

```
<?php
    session_destroy(); ?>
```

#### 14. How is PHP error handling done?

PHP error handling is done in two ways:

- (i) **Default error handling:** The default error handling in PHP is very simple - an error message with filename, line number and a message describing the error is sent to the browser. An example is shown below:

 Notice: Undefined index: uname in C:\wamp\www\PHP5-Example11.php on line 8				
Call Stack				
#	Time	Memory	Function	Location
1	0.0386	365552	{main}()	..\PHP5-Example11.php:0

- (ii) **Custom error handling:** To customize the error handling, PHP provides custom error handling techniques and these are:
- PHP die() method
  - Defining custom error handling functions and error triggers
  - Error reporting or logging

#### 15. What are the different PHP error handling methods?

Different PHP error handling methods are:

- (i) Default error handling  
(ii) Custom error handling

#### 16. What is PHP die() method?

PHP die() method is used to to overrule the default error handling in PHP. die() method is used to terminate the further execution of PHP script and give a specified error message to the user.

Example:

```
<?php
if (!file_exists("myfile.txt"))
    die("File not found");
else $file = fopen("myfile.txt","r");
?>
```

This code will check if the file "myfile.txt" exists. If the file does not exist, then it will give a message "File not found" to the user and terminate the further execution of the script.

#### 17. Write PHP code to read a text file abc.txt and display alternate lines of the file on the screen.

```
<!-- PHP7-ReadAlternate.php -->
<?php
$file = fopen("abc.txt","r");
while (!feof($file))
{
    $line = fgets($file);
    echo $line."<BR>";
    $line = fgets($file);
}
```

```
fclose($file);  
?>
```

**18. Write PHP code to read a text file xyz.txt and count the number of words, lines, and characters in the file.**

```
<!-- PHP7-CountWords.php -->  
<?php  
  
$words = $lines = $char = 0;  
$file = fopen("xyz.txt","r");  
while (!feof($file))  
{  
    $data = fgets($file);  
    $lines++;  
}  
fclose($file);  
$file = fopen("xyz.txt","r");  
while (!feof($file))  
{  
    $data = fgetc($file);  
    $char++;  
    if ($data == ' ')  
        $words++;  
    if (ord($data) == 13)  
    {  
        $words++;  
        $char--;  
        $data = fgetc($file);  
    }  
}  
fclose($file);  
$words++;  
echo "<P>File contains $words Words, $lines lines, and $char  
characters";  
?>
```

**19. Write PHP code to read a text file str.txt and display all the lines beginning with letter "A".**

```
<!-- PHP7-ShowALines.php -->  
<?php  
$file = fopen("str.txt","r");  
while (!feof($file))  
{  
    $line = fgets($file);  
    if ($line[0] == 'A')  
        echo $line."<BR>";  
}  
fclose($file);  
?>
```

**20. Write PHP code to read a text file abc.txt and display the contents of the file on the screen replacing all spaces in the file by the # symbol.**

```
<!-- PHP7-ShowReplaced.php -->
<?php
$file = fopen("abc.txt","r");
while (!feof($file))
{
    $char = fgetc($file);
    if ($char == ' ')
        echo "#";
    else echo $char;
    if (ord($char) == 13)
        echo "<BR>";
}
fclose($file);
?>
```

- 21. Write PHP code to read a text file data.txt and count the number of vowels, consonants and other characters present in the file.**

```
<!-- PHP7-CountVowels.php -->
<?php
$file = fopen("data.txt","r");
$vowels = $conso = $others = 0;
while (!feof($file))
{
    $char = fgetc($file);
    if (ord($char) != 13 && ord($char) != 10)
    {
        $char = strtoupper($char);
        if ($char=='A' or $char=='E' or $char=='I'
            or $char=='O' or $char == 'U')
            $vowels++;
        else if ($char >= 'A' and $char <= 'Z')
            $conso++;
        else
            $others++;
    }
}
fclose($file);
echo "File contains $vowels vowels, $conso consonants, and
    $others other characters";
?>
```

- 22. Write PHP code to read a text file named mega.txt and display the contents of the file after converting all the alphabets to upper case alphabets.**

```
<!-- PHP7-ConvertToUpper.php -->
<?php
$file = fopen("mega.txt","r");
while (!feof($file))
{
```

```
$char = fgetc($file);  
if (ord($char) == 13)  
    echo "<BR>";  
else echo strtoupper($char);  
}  
fclose($file);  
  
?>
```

**23. What is the difference between PHP fgets() and PHP fgetc()?**

fgets() reads one line of text from the specified file, whereas fgetc() read one character from the specified file.