**Part C: Superglobals:**

Q1:Provide a descriptions and usage for all the PHP **superglobals**

Ans:Several predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP superglobal variables are:

* $GLOBALS
* $\_SERVER
* $\_REQUEST
* $\_POST
* $\_GET
* $\_FILES
* $\_ENV
* $\_COOKIE
* $\_SESSION

**1)$GLOBALS**

$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).

* PHP stores all global variables in an array called $GLOBALS[*index*]. The *index* holds the name of the variable.

!DOCTYPE html>  
<html>  
<body>  
  
<?php   
$x = 30;  
$y = 25;   
  
function addition() {  
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];  
}  
  
addition();  
echo $z;  
?>  
  
</body>  
</html>

## 2) $\_SERVER

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

<!DOCTYPE html>  
<html>  
<body>  
  
<?php   
echo $\_SERVER['PHP\_SELF'];  
echo "<br>";  
echo $\_SERVER['SERVER\_NAME'];  
echo "<br>";  
echo $\_SERVER['HTTP\_HOST'];  
echo "<br>";  
echo $\_SERVER['HTTP\_REFERER'];  
echo "<br>";  
echo $\_SERVER['HTTP\_USER\_AGENT'];  
echo "<br>";  
echo $\_SERVER['SCRIPT\_NAME'];  
?>  
  
</body>  
</html>

**3) $\_REQUEST**

PHP $\_REQUEST is used to collect data after submitting an HTML form.

<!DOCTYPE html>  
<html>  
<body>  
  
<form method="post" action="<?php echo $\_SERVER['PHP\_SELF'];?>">  
  Name: <input type="text" name="fname">  
  <input type="submit">  
</form>  
  
<?php  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
    // collect value of input field  
    $name = htmlspecialchars($\_REQUEST['fname']);   
    if (empty($name)) {  
        echo "Name is empty";  
    } else {  
        echo $name;  
    }  
}  
?>  
  
</body>  
</html>

**4) $\_POST**

PHP $\_POST is widely used to collect form data after submitting an HTML form with method="post". $\_POST is also widely used to pass variables.

The example of post has been given in the lab8 partb.

**5) $\_GET**

PHP $\_GET can also be used to collect form data after submitting an HTML form with method="get".$\_GET can also collect data sent in the URL.

The example of post has been given in the lab8 partb.

**6) $\_FILES**

This is used to upload files through a php script.

Usage:

<html>

<body>

<form action="upload\_file.php" method="post"

enctype="multipart/form-data">

<label for="file">Filename:</label>

<input type="file" name="file" id="file" />

<br />

<input type="submit" name="submit" value="Submit" />

</form>

</body>

</html>

upload\_file.php

<?php

if ($\_FILES["file"] > 0)

{

echo "You have selected a file to upload";

}

?>

**7) $\_ENV**

$\_ENV is used to return the environment variables from the web server.

<?php

echo $\_ENV['username'];

?>

**8)$\_COOKIE**

Cookies are small text files loaded from a server to a client computer storing some information regarding the client computer, so that when the same page from the server is visited by the user, necessary information can be collected from the cookie itself, decreasing the latency to open the page.

<?php

setrawcookie();

print\_r($\_COOKIE);

?>

**9)$\_SESSION**

essions are wonderful ways to pass variables. All you need to do is start a session by session\_start();Then all the variables you store within a $\_SESSION, you can access it from anywhere on the server. Here is an example:

<?php

session\_start();

$\_SESSION['w3resource']='The largest online tutorial';

echo $\_SESSION['w3resource'];

?>