

. Using Cookies with PHP

UNIT – 5

5.1 Purpose of Cookies

Cookie एक Small Text File है जो उपयोगकर्ता को Computer पर Small Quantity में Data को Store करती है. इसको आम तौर पर Username जैसे जानकारी का Track रखने के लिए उपयोग किया जाता है. PHP के साथ आप Cookie Value को Create और Retrieve कर सकते हैं.

Client Browser में Cookie store करने का main reason end user को track करना होता है। जिससे उस information को future में use किया जा सके। Cookie का सबसे अच्छा example ' Remember Password ' Mechanism है जिसमें किसी user के username , password cookie के रूप में browser में store कर दिया जाता है , और जब हम उस login पेज दुबारा visit करते हैं तो हमें username , password fields fill मिलते हैं।

cookies mainly 2 तरह की होती है

- **analytical cookies** –इस तरह की cookie के माध्यम से आप किसी पुराने visitor को recognize कर सकते हैं तो आप और visitors की सख्या का भी पता लगा सकते हैं
- **functional cookies** –इस तरह की cookies को functional tasks को perform करने के लिए ही प्रयोग की जाती है जैसे की user की पुरानी choice आप इस तरह की cookies के द्वारा पता लगा सकते हैं

website के द्वारा cookies को कई कारणों से प्रयोग किया जाता है

- user experience को improve करने के लिए
- user को identify करने के लिए
- number of visitors count करने के लिए
- user की पुरानी choices को याद रखने के लिए

php में cookies को कुछ functions की मदद से ही manage किया जाता है

- **setcookie()**-ये function को cookies को create करने के लिए प्रयोग किया जाता है
- **isset()**-इस function के माध्यम से आप check कर सकते हैं की cookies को पहले से created है या नहीं

5.2 Cookies Myths

- **Myth #1: All cookies are malicious.** False. Malware and spyware are real internet threats that users need to take seriously; however, as stated above, cookies are not all designed with evil intents. Simplification, speed, and personalization are often their core functionalities. Many viruses are designed to be able to read the cookies on your machines, but they are designed to read everything. Cookies are just one piece of a virus's security breach potential.
- **Myth #2: Cookies spy on everything you do online.** False. Cookies are small stores of data designed for specific functions on specific websites. Not everything is being stored by them. It may appear that you are being followed when ads appear for products you've recently searched for on unrelated sites; however, there's a backend connection at work on such occasions. Cookies can allow targeted ads based on the websites you've visited, but they are only discovering the underlying code of sites pre-designed for such a connection.
- **Myth #3: Cookies are linked to individuals.** False. Cookies can identify a user using a specific browser, but personal identifying information isn't necessarily attached to this user. In fact, if the user were to close one browser (e.g., Chrome) and open another (e.g. Internet Explorer), a cookie could not be 100% sure that these sessions were linked to the same person. Cookies neither follow someone across devices (PC to tablet to phone); nor do they store private information to be recalled at any time and place.
- **Myth #4: Deleting cookies makes your computer faster.** False. Cookies are nearly inconsequential when it comes to speed and space on your computer. Due to browser restrictions, these tiny files are rarely much larger than 4 kilobytes. To give you some perspective, this blog is 15 kilobytes as a MS Word document; a video longer than 10 seconds is usually over 25 megabytes (25000 kb). Sure, cookies take up some space on your machine, but they are hardly the root of memory drain.
- **Myth #5: Blocking cookies reduces pop-ups.** False. Blocking cookies usually has the opposite effect. Many pop-up advertisements use cookies to track whether a user has already seen a certain ad. If a user has already closed a pop-up window, a cookie often stores that information, so that the same ad will not appear

repeatedly, annoying the user and thus becoming ineffective. If a user blocks cookies, a website will not know that a viewer has already seen a pop-up and will therefore show the pop-up every single time a user goes to a certain website.

cookie is created

Note: You might have to reload the page to see the value of the cookie.

- **Output:**

5.3 Setting Cookies

Setting Cookie In PHP: To set a cookie in PHP, the **setcookie()** function is used. The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

Syntax:

```
setcookie(name, value, expire, path, domain, security);
```

Parameters: The setcookie() function requires six arguments in general which are:

- **Name:** It is used to set the name of the cookie.
- **Value:** It is used to set the value of the cookie.
- **Expire:** It is used to set the expiry timestamp of the cookie after which the cookie can't be accessed.
- **Path:** It is used to specify the path on the server for which the cookie will be available.
- **Domain:** It is used to specify the domain for which the cookie is available.
- **Security:** It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

Below are some operations that can be performed on Cookies in PHP:

- **Creating Cookies:** Creating a cookie named Auction_Item and assigning the value Luxury Car to it. The cookie will expire after 2 days(2 days * 24 hours * 60 mins * 60 seconds).

```
<!DOCTYPE html>
<?php
    setcookie("Auction_Item", "Luxury Car", time() + 2 *
24 * 60 * 60);
?>
<html>
<body>
    <?php
        echo "cookie is created."
```

```
?>
<p>
    <strong>Note:</strong>
    You might have to reload the
    page to see the value of the cookie.
</p>

</body>
</html>
```

5.4 Retrieving, Expiring and Deleting Cookies

Creating cookies

php में cookies को create करने के लिए ही आप **setcookie()** function का प्रयोग करते हैं इस function के 6 parameters होते हैं इस function का format आपको निचे दिया जा रह है

```
<?php

setcookie(name-of-cookie,value,expire-time,path,domain-name,security);

?>
```

- **name-of-cookie-** ये cookie का नाम होता है ये एक unique नाम होता है इसी नाम के द्वारा आप cookie की value को access की जाती है
- **value** –ये वह value होती है जिसे आप store करना चाहते हैं तो ये value को कुछ भी हो सकती है जैसे की कोई string या integer की value आदि

- **expire time**-ये वो time होती है जो जब तक के लिए आप cookie को store करना चाहते हैं
- **path**-ये उस directory का path होता है जो जहा पर आप cookie को store करना चाहते हैं
- **domain name**- ये domain का नाम होता है जो यदि आपकी website बड़ी है तो आप उसका नाम यहाँ प्रयोग कर सकते हैं
- **security**-यदि आप इसको 1 पर set करते हैं तो आप cookie को केवल secure HTTPS के मध्यम से ही भेजी जा सकती है

php में **cookie** को create करने का **example** आपको निचे दिया जा रहा है

```
<?php

setcookie("name","yrName",time()+3600,"/cookies/", "",0);

setcookie("country","india",time()+3600,"/cookies/", "",0);

?>
```

Accessing cookies

php में cookie को **\$_COOKIE[]** array में store की जाती है यदि आप किसी एक **cookie** की value को access करना चाहते हैं तो आप **\$_COOKIE['cookie-name']** के variable के द्वारा access कर सकते हैं जैसे की आप यदि कोई पुराना user वापस आता है तो आप उसके नाम के साथ ही welcome message print को करवा सकते हैं एक ध्यान देने योग्य बात ये है की cookies को access करने से पहले आपको पता कर लेना चाहिए की user की request में कोई cookie को send की गयी है या नहीं इसके लिए आप **isset() function** का प्रयोग करते हैं पिछले example में आपने cookie को create की थी अब आप **cookie** को access करने का example आपको निचे दिया जा रहा है

```
<?php

if(isset($_COOKIE["name"]))

{

    echo "GOOD morning".$_COOKIE["name"];

}

?>
```

Deleting cookie

cookie का प्रयोग करने के बाद आप **cookie** को delete भी कर सकते हैं इसके लिए 2 तरीके हैं

- setcookie() function को सिर्फ cookie के नाम के साथ ही call कीजिये
- setcookie() function को past की कोई data के साथ ही call कीजिये

```
<?php

setcookie("name"); //here name is the

name of cookie.

?>
```

Example

php में **cookie** को create और प्रयोग करने का simple example आपको निचे दिया जा रहा है

example.php

```
<html>

<head>

<title>php cookie demo</title>

</head>

<body>

<form action="file1.php" method="post">

Enter your name"<input type="text"
```

```
Name="username"><br/>

<br/>

<input type="submit" value="submit">

</form>

</body>

</html>
```

File1.php

```
<?php>

$username=$_COOKIE["name"]&&

$_COOKIE["name"]==$username)

{

Echo "welcome back".$_COOKIE["name"];

}

Else
```

```
{

Echo "hello".$username."<br/>";

Echo "cookie was not set<br/>";

Echo "setting cookie now ....<br/>";

Setcookie("name",$username,time()+3600,"/",0);

Echo "cookie set";

}
```

उपर दिए गए example में एक form को create किये गया है की इस form के माध्यम से user से उसका नाम को enter करवाया जाता है form को submit करने पर आप **file1.php** file को execute की जाती है जब भी user को पहली बार visit कर रहे हैं तो उसके नाम से **cookie** से पहले से set नहीं होगी इसलिए आप user को message को show किया जायेगा की **cookie** पहले से set नहीं थी और अब set कर दी गयी है जब user को वापस return करता है तो user को उसके नाम के साथ ही **welcome back message** को show किया जायेगा

Cookies can store only string values. It can't store array values. But we are try to add array values into cookies in PHP.

5.5 Storing Arrays in Cookie

Store array into cookies in php using serialize():

Consider the following PHP script:

```
<?php
error_reporting('E_ALL ^ E_NOTICE');
$val=array('name'=>'Guru','country'=>'USA');
$val1=serialize($val);
setcookie('values',$val1);
?>
```

Where,

serialize() - Generate a storable values. Suppose you use above PHP code without serialize array value. Then you'll get error like this:

After serialize the array values, you can store array values into cookies using setcookie() in PHP.
Now the cookie contains array values.

Retrieve array values from cookies in PHP using unserialize():

Consider the following example:

```
<?php
error_reporting('E_ALL ^ E_NOTICE');
$val=array('name'=>'Guru','country'=>'USA');
$val1=serialize($val);
setcookie('values',$val1);
$dat=$_COOKIE['values'];
$data=unserialize($dat);
foreach($data as $key => $vl)
{
    echo $key.' : '.$vl.'<br>';
}
?>
```

Where,

unserialize() - takes serialized values and converts it into PHP value again. If you didn't use unserialize() in above PHP code, then you'll get error like this:

\$_COOKIES[""] is used to retrieve values from cookies.
Now you can get array values from cookies. The output is:

```
name : Guru
country : USA
```

Store multiple values into cookies in PHP using json_encode():

You can also store multiple values into cookies using json_encode() method in PHP.
json_encode() returns JSON representation of a value. Consider the example:

```
<?php
$val=array('name'=>'Guru','country'=>'USA');
$ar=json_encode($val);
setcookie('cook',$ar);
?>
```

After encoded the array values, you can store it into cookies. If you use above code without json_encode(), then you'll get warning error message.

Retrieve array values from cookies in PHP using json_decode();

You can retrieve array values from cookies using json_decode() like unserialize() in PHP.
json_decode() takes JSON encoded values and convert it into PHP value again. Consider the example:

```
<?php
$val=array('name'=>'Guru','country'=>'USA');
$ar=json_encode($val);
setcookie('cook',$ar);
$return=$_COOKIE['cook'];
$arr=json_decode($return, true);
foreach($arr as $key1 => $values)
{
    echo $key1.' : '.$values.'<br>';
}
?>
```

Now you can get output like this:

name : Guru

country : USA