State Management

Objective

In this chapter learner able to understand:

- Basic PHP Session Syntax
- User Authentications with Sessions
- Cookies
- Retrieve Cookies Value
- Session

Session with PHP

A PHP session variable is used to store information about, or change settings for a user session. Session variables hold information about one single user, and are available to all pages in one application.

PHP Session Variables

A PHP session solves this problem by allowing you to store user information on the server for later use (i.e. username, shopping items, etc).

Session information is temporary and will be deleted after the user has left the website. If you need a permanent storage you may want to store the data in a database.

Starting a PHP Session

Before you can store user information in your PHP session, you must first start up the session.

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

Storing a Session Variable

```
<?php
	session_start();
	$_SESSION['views']=1;
?>
```

Counter…

```
<?php
        session_start();
        if(isset($_SESSION['views']))
        $_SESSION['views']=$_SESSION['views']+1;
        else
        $_SESSION['views']=1;
        echo "Views=". $_SESSION['views'];
?>
```

Destroying a Session

If you wish to delete some session data, you can use the unset() or the session_destroy() function.

The unset() function is used to free the specified session variable:

<?php unset($_SESSION['views']); ?>

<?php session_destroy(); ?>

Note: session_destroy() will reset your session and you will lose all your stored session data.

Cookie

A cookie is a small bit of information stored on a viewer's computer by his or her web browser by request from a web page.

The information is constantly passed in HTTP headers between the browser and web server.

The browser sends the current cookie as part of its request to the server and the server sends updates to the data back to the user as part of its response.

Count..

The size of a cookie depends on the browser but in general should not exceed 1K (1,024 bytes).

The information can really be anything... it can be a name, the number of visits to the site, web based shopping cart information, personal viewing preferences or anything else that can be used to help provide customized content to the user.

Type Of Cookies

1.Session / Temporary cookie

session cookies as the name specifies, these are cookies that expires once the session expires or once the browser closed.

2. Persisent cookie

To create persistent cookies, we have to set the expire parameter in the setcookie

Setting a Cookie

PHP provides an easy way to set a cookie: the function setcookie.

setcookie (*name* , *value* , *expire* , *path*, *domain* , *secure*)

Setcookie parameters

Parameter	Meaning	Example value
name	The name that the cookie will use for storage and retrieval.	username
Value	The value stored in the cookie	Raj
expire	The date when the cookie will expire and be deleted. If you do not set this expiration date, then it will be treated as a session cookie and be removed when the browser is restarted.	Time()+60*60*24*7 tells the cookie to expire in a week
path	The path on the server in which the cookie will be available on. If set to '/', the cookie will be available within the entire domain.	/testing .If set to '/testing/', the cookie will only be available within the /testing/ directory and all sub-directories such as /testing/test/ of domain
domain	The domain that the cookie is available	To limit access to only www on site example.com use ww.example.com. To grant access to all domains, use .example.com.
secure	If set to 1, cookies are sent only over a secure HTTPS connection. HTTPS connections use encryption between the client and the browser to secure data.	True for secure and False for insecure, which is the default.

Creating a cookie

```
<?php
    setcookie("username",“raj");
    echo 'Cookie created.';
?>
```

Accessing a Cookie

```
<?php
    if (!isset($_COOKIE['username']))
    {
        echo ("Opps, the cookie isn't set!");
     }
    else
     {
         echo ("The stored username is ". $_COOKIE['username'] . ".");
     }
?>
```

Destroying a Cookie

Cookies can be destroyed or deleted by the client or the server.

Clients can easily delete their cookies by locating the Cookies folder on their system and deleting them.

Count..

The server can delete the cookies by:

Resetting a cookie by specifying expiration time

Resetting a cookie by specifying its name only

```
<?php
	setcookie("username","", time()-10 );
	echo ‘Cookie has been Deleted!!!';
?>
```

Chapter Summary

In this chapter, you have learned:

- Sessions are used for sharing variables containing the information of the same user between different web pages accessed by a particular user.
- Cookies are used for identifying users.
- For creating a cookie there is a built-in function setcookie() available:
- For deleting a cookie we have to provide expire parameter to setcookie() function.
- For start a session there is a built-in function start_session() available:
- When session is start we need to set session variables, for this we use $_SESSION as follows:
- For destroying a session we have to use session_destroy() or unset() built-in functions of PHP.

Thank you