# **PHP Sessions**

**What is a session?**

In general, session refers to a frame of communication between two medium. A PHP session is used to store data on a server rather than the computer of the user. Session identifiers or SID is a unique number which is used to identify every user in a session based environment. The SID is used to link the user with his information on the server like posts, emails etc.

**How are sessions better than cookies?**

Although cookies are also used for storing user related data, they have serious security issues because cookies are stored on the user’s computer and thus they are open to attackers to easily modify the content of the cookie.Addition of harmful data by the attackers in the cookie may result in the breakdown of the application.  
Apart from that cookies affect the performance of a site since cookies send the user data each time the user views a page.Every time the browser requests a URL to the server, all the cookie data for that website is automatically sent to the server within the request.

Below are different steps involved in PHP sessions:

* **Starting a PHP Session**: The first step is to start up a session. After a session is started, session variables can be created to store information. The PHP **session\_start()** function is used to begin a new session.Itals creates a new session ID for the user.

PHP code to start a new session:

<?php

session\_start();

?>

**Storing Session Data**: Session data in key-value pairs using the **$\_SESSION[]** superglobalarray.The stored data can be accessed during lifetime of a session.

Below is the PHP code to store a session with two session variables Rollnumber and Name:

<?php

session\_start();

$\_SESSION["Rollnumber"] = "11";

$\_SESSION["Name"] = "Ajay";

?>

**Accessing Session Data**: Data stored in sessions can be easily accessed by firstly calling **session\_start()** and then by passing the corresponding key to the **$\_SESSION** associative array.

The PHP code to access a session data with two session variables Rollnumber and Name is shown below:

<?php

session\_start();

echo'The Name of the student is :'. $\_SESSION["Name"] . '<br>';

echo'The Roll number of the student is :'. $\_SESSION["Rollnumber"] . '<br>';

?>

**Destroying Certain Session Data**: To delete only a certain session data,the unset feature can be used with the corresponding session variable in the **$\_SESSION** associative array.

The PHP code to unset only the “Rollnumber” session variable from the associative session array:

<?php

session\_start();

if(isset($\_SESSION["Name"])){

    unset($\_SESSION["Rollnumber"]);

}

?>

**Destroying Complete Session**: The **session\_destroy()** function is used to completely destroy a session. The session\_destroy() function does not require any argument.

<?php

session\_start();

session\_destroy();

?>

Example

Myfile.html

<html>

<body>

<a href="http://localhost/san/session1.php">Call Session program</a>

</body>

</html>

Session1.php

<?php

session\_start();

?>

<html>

<body>

<?php

$\_SESSION["user"] = "geu";

echo "Session information are set successfully.<br/>";

?>

<a href="http://localhost/san/welcome.php">Visit next page</a>

</body>

</html>

Welcome.php

<?php

session\_start();

?>

<html>

<body>

<?php

echo "User is: ".$\_SESSION["user"];

echo "Hello: " .$\_SESSION["user"];

?>

<br>

<br>

<a href="http://localhost/san/session1.php">go back</a><br>

<a href="http://localhost/san/logout.php">click here for logout</a>

</body>

</html>

Logout.php

<?php

session\_start();

session\_destroy();

?>

Myfile.html

<html>

<body>

<a href="http://localhost/san/session1.php">Call Session program</a>

</body>

</html>

Session1.php

<?php

session\_start();

?>

<html>

<body>

<?php

$\_SESSION["user"] = "geu";

$\_SESSION["pass"] = "xyz";

echo "Session information are set successfully.<br/>";

?>

<a href="http://localhost/san/welcome.php">Visit next page</a>

</body>

</html>

Welcome.php

<?php

session\_start();

?>

<html>

<body>

<?php

echo "User is: ".$\_SESSION["user"];

echo "password is: " .$\_SESSION["pass"];

?>

<br>

<br>

<a href="http://localhost/san/verify.php">verify password</a><br>

<a href="http://localhost/san/logout.php">click here for logout</a>

</body>

</html>

<?php

session\_start();

?>

Verify.php

<html>

<body>

<?php

$un=$\_SESSION["user"];

$pass=$\_SESSION["pass"];

if($pass=="xyz")

echo "valid user";

else

echo "invalid user";

?>

<a href="http://localhost/san/welcome.php">go back</a><br>

</body>

</html>

Logout.php

<?php

session\_start();

session\_destroy();

?>

**What is a Cookie?**

**A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.**

**Create Cookies With PHP**

**A cookie is created with the setcookie() function.**

**Syntax**

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

Only the name parameter is required. All other parameters are optional.

<?php

$cookie\_name = "user";

$cookie\_value = "John Doe";

setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 30), "/"); // 86400 = 1 day

?>

<html>

<body>

<?php

if(!isset($\_COOKIE[$cookie\_name])) {

echo "Cookie named '" . $cookie\_name . "' is not set!";

} else {

echo "Cookie '" . $cookie\_name . "' is set!<br>";

echo "Value is: " . $\_COOKIE[$cookie\_name];

}

?>

</body>

</html>

**Note: The setcookie() function must appear BEFORE the <html> tag.**

**Delete a Cookie**

**To delete a cookie, use the setcookie() function with an expiration date in the past:**

**<?php**

**// set the expiration date to one hour ago**

**setcookie("user", "", time() - 3600);**

**?>**

**<html>**

**<body>**

**<?php**

**echo "Cookie 'user' is deleted.";**

**?>**

**</body>**

**</html>**