Lab 07: Sessions and Cookies

In this lab exercise, you will learn how to store values in variables accessible across multiple pages within one application. There are two methods that are covered in this lab exercises, i.e. session and cookie. Both methods are not permanent. For a permanent method, a database may be use. More on how to use database to store values of variable will be covered in different topic.

1.1 Sessions

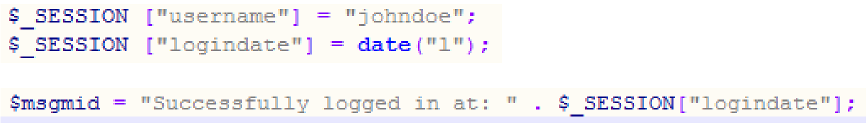
By default, session variables last until the user closes the web browser. They hold information about one single user, and are available to all pages in one application.

A session in PHP is started with function: session_start ( );

This function keyword must be placed before any HTML tag being transmitted to the web browser. It is highly recommended to place it at the start of PHP page.

Session variables are set and retrieved with the PHP global variable: $_SESSION [<variable name>]. This keyword may appear anywhere within a PHP page, after session_start ( );

Figure 1 below illustrates how to set values to session variables and access them.



```
$_SESSION ["username"] = "johndoe";
$_SESSION ["logindate"] = date("l");
```

```
$msgmid = "Successfully logged in at: " . $_SESSION["logindate"];
```

Figure 1: Setting and retrieving values stored in session variables.

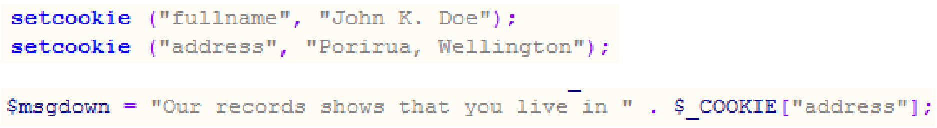
1.2 Cookies

Unlike session variables which are stored at the web server, a cookie variable (or a cookie in short) is stored at the user's computer, and therefore often used to identify the user. A cookie is generally in form of a small file. Each time the same computer requests a page with a browser, the cookies are sent along.

Cookies are set using keyword: setcookies (<variable name>, <value>, <expiry>). Values are retrieved using keyword: $_COOKIE [<variable name>].

There is no requirement to start a cookie function prior to using this keyword.

Figure 2 below illustrates how to set values to cookie variables and access them.

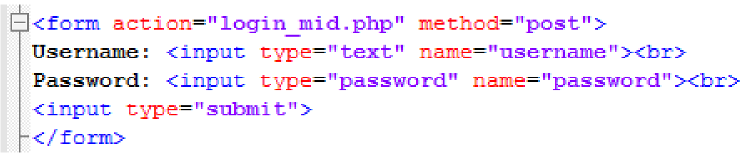


```
setcookie ("fullname", "John K. Doe");
setcookie ("address", "Porirua, Wellington");
```

```
$msgdown = "Our records shows that you live in " . $_COOKIE["address"];
```

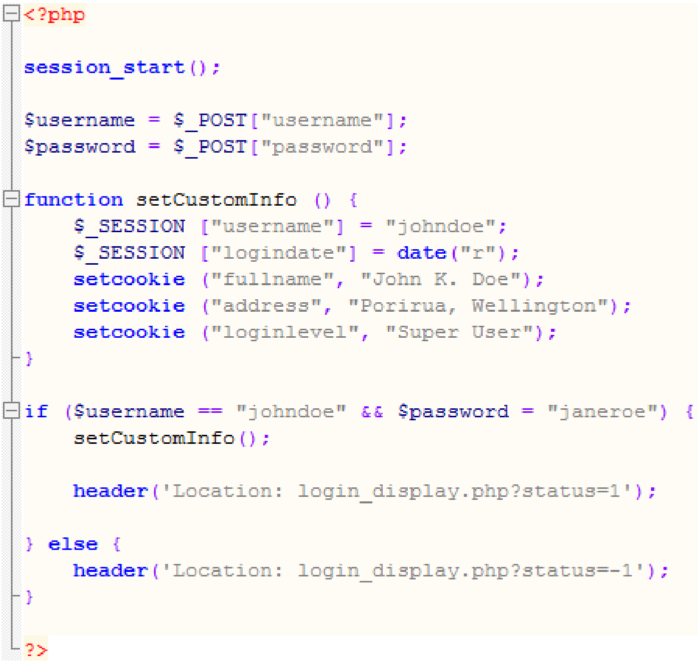
Figure 2: Setting and retrieving values stored in cookies.

Sample Session and Cookie in Action



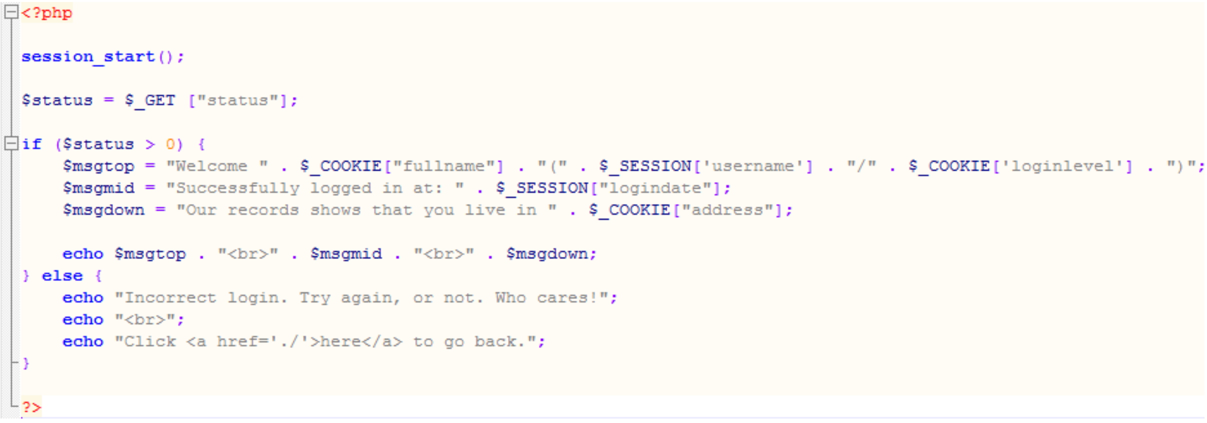
```
<form action="login_mid.php" method="post">
Username: <input type="text" name="username"><br>
Password: <input type="password" name="password"><br>
<input type="submit">
</form>
```

Figure 3: An HTML page, with login form posting to login_mid.php.



```
<?php

session_start();

$username = $_POST["username"];
$password = $_POST["password"];

function setCustomInfo () {
    $_SESSION ["username"] = "johndoe";
    $_SESSION ["logindate"] = date("r");
    setcookie ("fullname", "John K. Doe");
    setcookie ("address", "Porirua, Wellington");
    setcookie ("loginlevel", "Super User");
}

if ($username == "johndoe" && $password = "janeroe") {
    setCustomInfo();

    header('Location: login_display.php?status=1');

} else {
    header('Location: login_display.php?status=-1');
}

?>
```

Figure 4: A PHP page, login_mid.php, setting the session and cookie variables.



```
<?php

session_start();

$status = $_GET ["status"];

if ($status > 0) {
    $msgtop = "Welcome " . $_COOKIE["fullname"] . "(" . $_SESSION['username'] . "/" . $_COOKIE['loginlevel'] . ")";
    $msgmid = "Successfully logged in at: " . $_SESSION["logindate"];
    $msgdown = "Our records shows that you live in " . $_COOKIE["address"];

    echo $msgtop . "<br>" . $msgmid . "<br>" . $msgdown;
} else {
    echo "Incorrect login. Try again, or not. Who cares!";
    echo "<br>";
    echo "Click <a href='./'>here</a> to go back.";
}

?>
```

Figure 5: A PHP page, login_display.php, displaying the session and cookie variables.

You tasks:

1. Create a page that allows the user to enter some information.
2. Post the information to an intermediary PHP page, and save them into either session or cookie variables.
3. Redirect to another PHP page to display the information.
4. Allows for error checking and handling.
5. Include an option to upload a file together with information the user enters, if you wish.
6. Provide options to erase or replace information stored on session/cookie variables.

-- END --