Practicals

Practical 9 – PHP Basic Session and Login

Sessions:
Sessions will be used to ensure that only valid users can view any page. At the head of each page a check can be made for a session variable – if it is not set then access to that page is refused (by redirecting back to the login page).

Sessions are passed automatically between pages either using cookies (first choice if enabled) or in the URL of the page (containing the unique session id assigned to the session). Default is a cookie that expires when the browser is closed.

PHP sessions must be started using `session_start();` which should be placed at the top of each page and stopped using `session_destroy();`

The global session variable (behaving like an associative array) can be set to a value, e.g.

```
$_SESSION['colour'] = "blue";
$_SESSION['login'] = $username;
```

At the head of each page whether a session variable has been set can be checked (or alternatively the value can be checked). If a session variable is not set or does not move an appropriate value then the user can be redirected to the login page

1. Getting Started

To get started, create the following files:

main1.php

```
<?php
session_start();

echo "Start of main 1<br>";
echo "login session variable is " . $_SESSION['login'] .
"<br><br>";

if (!isset($_SESSION['login'])){
   echo "Setting login session variable<br>";
   $username = 'name';// Add your name in place of 'name'
   $_SESSION['login']= $username;

/*These 3 lines of code are only here because there is no
login yet. The session login variable should be set by
the login checker. When it is, a location () command will
replace this and redirect back to the login page*/

}
echo "Inside main 1<br>";
echo "login session variable is " . $_SESSION['login'] .
"<br><br>";
```

```
echo "MENU<br>";
echo "<a href='page1.php'>Page 1</a><br>";
echo "<a href='logoutfile.php'>Logout</a><br>";
?>
```

page1.php

```
<?php
session_start();
if (!isset($_SESSION['login'])){
       header("location: main1.php");  // redirects to
main page
}
echo "Inside Page One<br>";
echo "login session variable is " . $_SESSION['login'] .
"<br><br>";
echo "MENU<br>";
echo "<a href='main1.php'>Return to Main 1</a><br>";
echo "<a href='logoutfile.php'>Logout</a><br>";
 ?>
```

logoutfile.php

```
<?php
session_start();
session_destroy();
echo "You are now logged out<br><br>";
echo "<a href='main1.php'>Go to Main 1</a><br>";
echo "<a href='page1.php'>Go to Page 1</a><br>";
?>
```

When you are done:

- Run main1.php and examine the effect on the session variable when you choose the links for page1.php and then return to main1.php
- Notice the effect of choosing logout.php on the session variable.
- Try going to Page 1 after logout or running Page 1 from localhost.

2. Basic Login System

Firstly a table with user details (name & password) must be created:

Create a user table in MySQL:

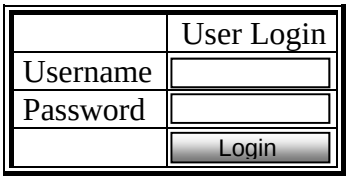
User(<u>ID</u>, username, password)

With username and password as char data types.

<u>Basic Login Page</u>
Create a table within a table to hold the user name & password textboxes and login submit button

- Outer table has border = 1 with one cell
- Inner table has border = 0 (transparent) with2 column and 4 rows

- Password textbox should use type="password" (does not echo password on screen)
- Input login name and password values can be posted to a login check script.



Login Check

- Receives the posted username and password details from the login page
- Plugs them in to an SQL retrieval query getting values from the user table using the received username and password which is then run
- If a row has not been returned then redirect back to the login page (a more sophisticated approach would check the username and password individually and comment to the user whether a username existed or whether the password was incorrect)
- If a row has been returned then
  - Set appropriate session variables e.g.
    - $_SESSION['login'] = "Yes";
    - $_SESSION['username'] = $username;
  - Go to the main menu

Sample login_check.php file
(with username & password in code rather than taken from a database check to the User table)

```
<?php
session_start();
$username = $_POST["username"];
$password=$_POST["password"];

if ($username=="John" && $password=="pass"){
     $_SESSION['login']= $username;
    header("location: main1.php");
}
else{
    header("location: login.html");
}
?>
```

Modify this file so that a query to the User table is carried out using the data from the form.

3. PHP and AJAX for Login pages

Logins can be handled by PHP except that for individual checking of the username and then, if OK, the password will require recreation of the login page more than once to allow the user to re-enter login details. In a real implementation recreating the whole page multiple times across the internet may be slow.

An alternative is to use AJAX (Asynchronous Javascript with XML) to allow communication between a webpage mediated by Javascript using XMLHttpRequest calls. This works without the need for a complete page refresh but the process is driven by Javascript rather than PHP.

The request object sends the values input to the login textboxes (using DoCallback( )). The PHP file receives the posted values and the echo values are sent as textReponse values to the Javascript LoginStatus function where they are handled as either alert boxes or, on successful login, a redirection to newpage.php is achieved using document.location. Note that the session variable $_SESSION['Login'] is set on successful login & checked for when newpage.php is opened.

EXERCISE

- Cut and paste the 4 code files below (Ajax_Login.html, Check_Login1.php, newpage.php & ajax.js) into files with these specified names into the htdocs directory
- Run Ajax_Login.html on localhost (using the username & password user1 & pass1 as in the PHP code). Try entering the username & password, incorrectly then correctly.

Note the .js file referenced in Ajax_Login (ajax.js). It contains the definition of the XMLHttpRequest Object (allowing for different Browsers) and the Response Handler (processReqChange( )) which checks that a reply has been received (req.ReadyState == 4 and req.status == 200) and if so evaluates the responseText (defined as what – see var what in Ajax_login.html)

Ajax_Login.html

```
<html> <head>
  <script>
 var url = "Check_Login1.php";
 var what = "LoginStatus(req.responseText)";

function CheckLogin()
        {
        var username =
document.getElementById("username").value;
        var password =
document.getElementById("password").value;

DoCallback("username="+username+"&password="+password);
         }
```

```
function LoginStatus(Status)
       {
        if(Status == 1)
                    alert("username left empty");
         else if (Status == 2)
                    alert("Wrong username");
         else if (Status == 3)
                    alert("Username OK but Wrong
Password");
         else if (Status ==4)
                   document.location = "newpage.php";
        else
            document.location = "Ajax_Login.html";
          }
  </script>
 <script src="ajax.js" type="text/javascript"></script>
 </head>
 <body>
  <br>
Username: <input type="text" id="username" ><br><br>
Password  <input  type="password" id="password"><br><br>
<input type="button" value="Check Login"
onClick="CheckLogin()">

</body></html>
```

Check_Login1.php

```
<?php
session_start();
if(!empty($_POST["username"]))  // if username not empty
    {
    $username = $_POST["username"];
    $password = $_POST["password"];

    if($username == "user1") {          // Simulation of
database login query for $username

          if($password == "pass1") {       // Simulation
of database login query for $password

               $_SESSION['login'] = $username;
               echo 4;     // Success!
          }
          else echo 3;     // username OK but password
incorrect
    }
    else echo 2;      //username incorrect
}
else echo 1;     // usernameis blank

?>
```

newpage.php – page with a session login check

```
<?php
session_start();
if (!isset($_SESSION['login'])) {
     header("location: Ajax_Login.html");
}
else{
     print "In newpage.php<br>";
   session_destroy( );  // simulation of logout
}
?>
```

ajax.js

```
function DoCallback(data)
{
    // branch for native XMLHttpRequest object
    if (window.XMLHttpRequest) {
        req = new XMLHttpRequest();
        req.onreadystatechange = processReqChange;
        req.open('POST', url, true);
        req.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
        req.send(data);
    // branch for IE/Windows ActiveX version
    } else if (window.ActiveXObject) {
        req = new ActiveXObject('Microsoft.XMLHTTP')
        if (req) {
            req.onreadystatechange = processReqChange;
            req.open('POST', url, true);
            req.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
            req.send(data);
        }
    }
}

function processReqChange() {
    // only if req shows 'loaded'
    if (req.readyState == 4) {
        // only if 'OK'
        if (req.status == 200) {
            eval(what);
        } else {
            alert('There was a problem retrieving the XML
data: '  req.responseText);
        }
    }
}
```