Facilitator Meeting

Instructions 8 – Session, Include, and Hidden Fields in PHP

Aims:

This lab activity will help you learn to use PHP sessions, include files, and hidden fields in a basic form submission workflow. You will build a simple login form, manage user sessions, and pass data using hidden fields.

the Mercury server to your facilitator before the due date to be marked off. Tasks will not be marked if the email is not received.

All delivered web pages should conform to HTML5 and must be validated.

NOTE: To validate that the code generated by your PHP script is valid HTML5 you must view the code received by your browser. You cannot successfully validate HTML directly from PHP source file.

Task1: Create a Simple HTML Login Form

1. Create a folder named lab08 in the htdocs folder of XAMPP and add the files in this folder.

2. Create an HTML form (`login.html`) that accepts a username and password. Use the POST method to submit the form to `process.php`.

3. In the form, include a hidden field that stores a fixed value like a user token (initial letter of your first name+your student ID).

Example code for login.html:

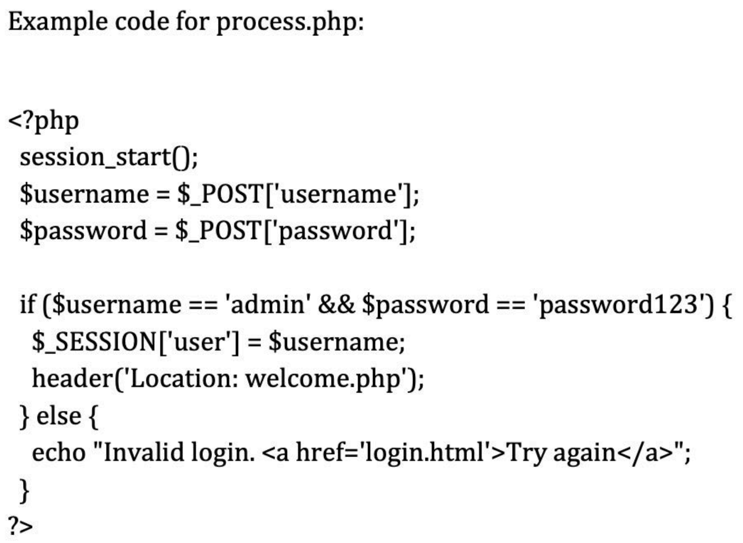
```
<form method="post" action="process.php">
  <label for="username">Username:</label>
  <input type="text" name="username" required><br>

  <label for="password">Password:</label>
  <input type="password" name="password" required><br>

  <input type="hidden" name="token" value="abc123">
  <input type="submit" value="Login">
</form>
```

Task 2: Create PHP to Process the Login Form and Start a Session

1. In `process.php`, start a session using `session_start()`.

2. Use `$_POST` to capture the username and password from the form. If the username is correct 'YourName' and the password is 'YourStudentID', set a session variable `$_SESSION['user']` to store the username.

3. Redirect the user to a `welcome.php` page if login is successful. If unsuccessful, return to the `login.html` page with an error message.



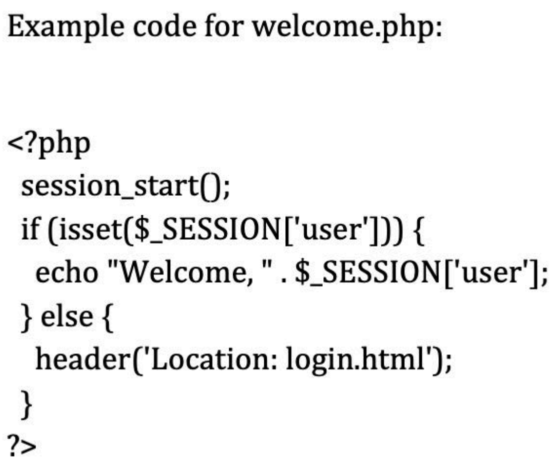
Example code for process.php:

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

 if ($username == 'admin' && $password == 'password123') {
   $_SESSION['user'] = $username;
   header('Location: welcome.php');
 } else {
   echo "Invalid login. <a href='login.html'>Try again</a>";
 }
?>
```

Task 3: Create the Welcome Page

1. In `welcome.php`, start a session and check if the user is logged in by checking the session variable `$_SESSION['user']`.

2. If the user is logged in, display a welcome message with their username. Otherwise, redirect them back to `login.html`.



Example code for welcome.php:

```
<?php
 session_start();
 if (isset($_SESSION['user'])) {
   echo "Welcome, " . $_SESSION['user'];
 } else {
   header('Location: login.html');
 }
?>
```

Task 4: Create an Include File

1. Create a file named `header.php` and 'footer.php' that contains a simple HTML header section (e.g., title, navigation bar) and footer (copyright, navigation links).

2. Use the `include` statement in both `login.html` and `welcome.php` to insert the header from `header.php`.

Task 5: Create an array and a table

1. Define Another Username and Password:

In the process.php file, define another username and password for your best friend. Use their actual name as the username and a nickname or something you like to call them as the password. Think about how you previously checked the username and password for one user. Now, just add another condition to check for a second user. Use an if statement to compare the entered username and password with your friend's credentials.

2. Define Movie Arrays:

In the same process.php file, define two arrays: one with your 5 favourite movies and another with 5 movies your friend likes (you can make an educated guess for your friend's list). Arrays in PHP are lists of items. To store your favourite movies, create a variable and assign an array to it. Similarly, create another array for your friend's favourite movies. Arrays look like this: $my_movies = ["movie1", "movie2", "movie3"];.

3. Display Movies Based on Login:

After successful login, use an if statement to check who has logged in (you or your friend). Then, use a for loop to display the respective list of favourite movies in a table on the welcome.php page.

Hints: After checking which user has logged in, you need to select the appropriate array of movies (either yours or your friend's). Use an if statement to decide which array to display.

To display the list of movies in a table, think about how to loop over an array using a for loop. Remember, you can access array elements by their index, like this: $my_movies[0] gives you the first movie.

For the table, you can create HTML <table> tags and use a loop to fill in the rows with movie names.

Remember that $_SESSION is used to store data across pages. After checking the login details, store the relevant movie list in a session variable so that it can be accessed later on the welcome.php page.

If the login is successful, redirect the user to the welcome.php page using the header() function in PHP.

4.*Validate the HTML generated by your PHP script* and upload lab08 folder on Mercury.