Lab 9 – Managing State Information

Reference: Chapter 8 of the "PHP Programming with MySQL" textbook.

PHP Sessions http://php.net/manual/en/book.session.php

Aims:

- To be able to manage state information by passing information from one PHP page to another using **sessions**.
- To become aware of how to provide redirection from one script to another in PHP.

Getting Started:

Create a new folder '**lab09**' under the unit folder on the mercury server *~/cos30020/www/htdocs*.
Save today's work in this lab09 folder.

All Web pages must be validated.
You could also create and link an external stylesheet, to the pages, and this should be valid CSS3.

Task 1: Up and down counter using session (9 marks)

The overall task is to create a simple web application that displays an integer and contains three links to update the integer. One link increments the integer by 1, another link decrements the integer by 1, and the last link sets it to 0.

Step 1:
Create a file **`number.php`** that starts up a session, creates a session variable if it does not exists and displays it on the web page.

```
  <?php
    session_start();                      // start the session
    if (!isset ($_SESSION["number"])) {   // check if session variable exists
      $_SESSION["number"] = 0;            // create the session variable
    }
    $num = $_SESSION["number"];           // copy the value to a variable
  ?>

  <!DOCTYPE html>
  <html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="Web application development" />
    <meta name="keywords" content="PHP" />
    <meta name="author"   content="Your Name" />
    <title>TITLE</title>
  </head>
  <body>
  <h1>Web Programming - Lab09</h1>
  <?php
      echo "<p>The number is $num</p>";        // displays the number
  ?>
  <p><a href="numberup.php">Up</a></p>         <!—links to updating page -->
  <p><a href="numberdown.php">Down</a></p>
  <p><a href="numberreset.php">Reset</a></p>
</body>
</html>
```

Step 2:
Create a file **`numberup.php`** that increments the session variable by 1. This page does not contain any HTML tags and redirects to the **`number.php`** after update.

```
<?php
  (___(1)___);                          // start the session
  (___(2)___);                          // copy the value to a variable
  $num++;                               // increment the value
  $_SESSION["number"] = $num;           // update the session variable
  header("location:number.php");        // redirect to number.php
?>
```

Step 3:
Create a file **`numberdown.php`** that decrements the session variable by 1. This page does not contain any HTML tags and redirects to the **`number.php`** after update.

```
<?php
  (___(3)___);                          // start the session
  (___(4)___);                          // copy the value to a variable
  (___(5)___);                          // decrement the value
  (___(6)___);                          // update the session variable
  (___(7)___);                          // redirect to number.php
?>
```

Step 4:
Create a file **`numberreset.php`** that clears out all session variables and redirects to the **`number.php`** after reset.

```
<?php
  session_start();              // start the session
  (___(8)___);                  // unset all session variables
  (___(9)___);                  // destroy all data associated with the session
  (___(10)___);                 // redirect to number.php
?>
```

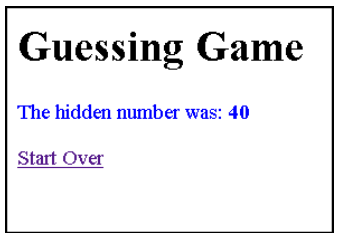
Test in the browser, and check that the page is valid.

Task 2: Creating a simple "Guessing Game" (6 marks)

The overall task is to create a simple web application that generates and uses **sessions** to store a random number between 0 and 100.

Step 1:
Create a file **`guessinggame.php`** that will be the main page for the game. In this page, a user inputs their guess; the page displays the number of times the user has guessed; whether their number is higher or lower than the generated number; and congratulates them when they guess correctly. (Checking always if the input data is "in-range" and is numeric). It also include a `'Give Up'` link to `giveup.php`, and a `'Start Over'` link to `startover.php`.

Hint:
The PHP's `rand()` function can be used to generate a random integer.
The `rand()` function accepts two arguments: the first argument specifies the minimum integer to generate; and the second argument specifies the maximum integer to generate.
For example, the statement `$randNum = rand(10,20)` generates a random integer between `10` and `20` and assigns the number to the `$randNum` variable.



Step 2:
Create a file `giveup.php` that displays the random number generated for the current game. The value of the random number is accessed via the session variable.

Step 3:
Create a file `startover.php` that has no html tags, it simply destroys the session then **redirects** the user back to `guessinggame.php`

Hint: Use `header("location:URL")` function to provide redirection.