CSCI 2910
Client/Server-Side Programming

Topic: Sessions in PHP
Reading: Williams & Lane pp. 338-368

Today's Goals
Today's lecture will cover:
• An introduction to sessions, their purpose, and their use
• Starting and stopping a session
• Using session variables

Purpose of a Session
• In general, HTTP is a "stateless" system, i.e., clients access documents through links without regard to past interactions
• This is not acceptable when it comes to managing a complex interaction with a client such as:
  – the use of a shopping cart;
  – logging into a database or other secure site; or
  – tracking a user's settings/data values as he or she progresses through a site.

Mechanics of a Session
• Session is identified using a session ID (32 digit hexadecimal value)
• The session ID is transmitted between the client and server with each HTTP request and response
• Client keeps track of a session through the use of a cookie
• Server keeps track of a session through locally stored text files or a database

Mechanics of a Session (continued)
• Databases are used for large traffic applications while text files are used for lower traffic.
• The server maintains the session variables in the text file or database.
• To prevent security risks due to someone hijacking an old session and to avoid clogging the server with unused sessions, the server will clean up old sessions after a specified timeout period.

Implementing a Session
• `session_start()` – creates a new session or finds an existing session. Basically, it identifies a session and accesses the session's variables if it is an existing session.
• Once a session has been started, the session's variables are accessed through a superglobal associative array called `$_SESSION`. (This is the same sort of array as `$_GET` and `$_POST`.)
• Example: `$_SESSION['variable_name']`
Implementing a Session (continued)

• Because of the dual purpose of session_start(), i.e., it can initialize a session or access an existing one, the PHP code must have a method for identifying whether a session has already been initiated.
• isset($_SESSION[variable_name]) can be used to determine if the session has already initialized a particular variable. For example:

```php
if (!isset($_SESSION['quantity']))
    $_SESSION['quantity'] = 0;
else
    $_SESSION['quantity']++;  
```

Implementing a Session (continued)

• A variable can be removed from a session using the unset() function
• Example:

```php
unset($_SESSION['quantity']);
```
• All session variables can be removed by simply re-initializing the $_SESSION array
• Example:

```php
$_SESSION = array();
```

Implementing a Session (continued)

• Since the client receives its session ID through a cookie in one of the HTTP header files, it must be sent before any HTML is generated for the client's output.
• Therefore, session_start() must be executed before any output is generated.

Session Variable Types

• A session variable can be of any type or object
• If a session variable is an object, be sure to define the object before running session_start().
• If an existing session that uses an object is opened before the object is defined, it will cause problems.
• The following slide presents an example

```php
<?php
    session_start();
    if(!isset($_SESSION['count']))
        $_SESSION['count'] = 0;
    else
        $_SESSION['count']++;  
?>

<html>
<head><title>Hello, World! in PHP</title></head>
<body>
<?php
    print "You've visited ".$_SESSION['count']." times.";
?>
</body></html>
```

Ending a Session

If you receive a signal from the client that they are ending the session, e.g., clicking on a link to log out, there is a method you can use to force a session to end.

– First, you must open the session so that it is identified as the one to be closed.
– Next, you must end the session with the function session_destroy()
– Last, use the header function to direct an output to the client.
Ending a Session (continued)

```php
<?php
// Begin by accessing the session
session_start();
// Close the session
session_destroy();
// Direct output to the client
header("Location: logout.html");
?>
```