Lecture 13
More PHP

Boriana Koleva
Room: C54
Email: bnk@cs.nott.ac.uk

Overview

- Form Handling
- Files
- Cookies
- Session Tracking
- Architectures for Database Access
- The MySQL Database System
- Database Access with PHP/MySQL

Form Handling

- Forms could be handled by the same document that creates the form, but that may be confusing
  - A separate document to handle the form can be specified as the value of the `action` attribute
- It does not matter whether GET or POST method is used to transmit the form data
- PHP builds an array of the form values
  - $_GET for the GET method
  - $_POST for the POST method
  - subscripts are the widget names
- http://www.cs.nott.ac.uk/~bnk/WPS/popcorn2.html
- http://www.cs.nott.ac.uk/~bnk/WPS/popcorn2.pdf

Files

- PHP is able to create, read and write files on the server system
- **Opening a file**
  - Prepares file for use and associates a variable with the file for future reference
  - `$fptr = fopen(filename, use_indicator)`
  - Every open file has an internal pointer (where the next file operations should take place)
  - Because `fopen` could fail, use it with `die`

```
$file_var = fopen ("test.dat", "r") or
  die ("Error - test.dat can't be opened");
```

File Use Indicators

Use Indicator	Description
`"r"`	Read only. The file pointer is initialized to the beginning of the file.
`"r+"`	Read and write an existing file. The file pointer is initialized to the beginning of the file; if a read operation precedes a write operation, the new data is written just after where the read operation left the file pointer.
`"w"`	Write only. Initializes the file pointer to the beginning of the file; creates the file if it does not exist.
`"w+"`	Read and write. Initializes the file pointer to the beginning of the file; creates the file if it does not exist. Always initializes the file pointer to the beginning of the file before the first write, destroying any existing data.
`"a"`	Write only. If the file exists, initializes the file pointer to the end of the file; if the file does not exist, creates it and initializes the file pointer to its beginning.
`"a+"`	Read and write a file, creating the file if necessary; new data is written to the end of the existing data.

Files

- Use `file_exists(filename)` to determine whether file exists before trying to open it
- Use `fclose(file_var)` to close a file
- **Reading from a file**
  1. Read all or part of the file into a string variable
     - `$str = fread($file_var, #bytes)`
     - To read the whole file, use `filesize(file_name)` as the second parameter
     - `$file_string = fread ($file_var, filesize("test.dat"));`
  2. Read the lines of the file into an array
     - `$file_lines = file(file_name);`
     - Need not open or close the file

Files

- **Reading from a file (continued)**
  3. Read one line from the file
     - `$line = fgets(file_var, #bytes)`
     - Reads characters until eoln, eof, or #bytes characters have been read
  4. Read one character at a time
     - `$ch = fgetc(file_var)`
     - Control reading lines or characters with eof detection using `feof` (`TRUE` for eof; `FALSE` otherwise)

```
while(!feof($file_var)) {
      $ch = fgetc($file_var);
}
```

Files

- Writing to a file
  - $bytes_written = fwrite($file_var, $out_data);
  - `fwrite` returns the number of bytes it wrote
- Files can be locked (to avoid interference from concurrent accesses) with `flock`
  - Takes 2 parameters – file variable and intreger that specifies particular operation
    - 1 – file can be read by others
    - 2 – no other access
    - 3 – unlocks file

Cookies

- Create a cookie with `setcookie`
- `setcookie(cookie_name, cookie_value, lifetime);`
- `setcookie("voted", "true", time() + 86400);`
- Cookies must be created before any other HTML is created by the PHP document
  - Because cookies stored in HTTP header
- Cookies are obtained in a script the same way form values are gotten, using the `$_COOKIES` array (cookie names as keys)
  - Use `IsSet` to check if a particular cookie came with the request

Session Tracking

- An alternative to cookies
- For session tracking, PHP creates and maintains a session tracking id
- Create the id with a call to `session_start` with no parameters
- Subsequent calls to `session_start` retrieve any session variables that were previously registered in the session (in `$_SESSION` array)
- Session variable are created or changed by assignments to the $_SESSION array

Session Tracking

- Example: count number of pages visited
  - Put the following code in all documents

```
session_start();
if (!IsSet($_SESSION[“page_number”]))
      $_SESSION[“page_number”]= 1;
$page_num = $_SESSION[“page_number”];
print("You have now visited $page_num");
print(" page(s) <br>");
$_SESSION[“page_number”]++;
```

Architectures for Database Access

- A two-tier system has clients that are connected directly to the database server
- Client tasks:
  - Provide a way for users to submit queries
  - Run applications that use the results of queries
  - Display results of queries
- Database server tasks:
  - Implement a data manipulation language, which can directly access and update the database
- However, because the relative power of clients has grown considerably, we could shift processing to the client, but then keeping all clients current with application updates is difficult

Architectures for Database Access

- A solution to the problems of two-tier systems is to add a component in the middle
  - create a three-tier system
- For Web-based database access, the middle tier can run applications (client just gets results)

Architectures for Database Access

- PHP & Database Access
  - An API for each specific database system (e.g. MySQL API)
  - Convenient for Web access to databases, because PHP is run on the Web server

The MySQL Database System

- A free, efficient, widely used SQL implementation
- Available from http://www.mysql.org
  - Installed on CS servers (Avon, Bann, Clyde, Mersey, Roach and Severn)
- You need to create an account
  - create_mysql
  - Remeber to make a note of the password you are given, you should change this as soon as you log into *mysql*

The MySQL Database System

- mysql [-u USERNAME] [DATABASE NAME] -p
  - USERNAME is the MySQL username
  - DATABASE NAME is the name of the database
    - which is also usually the same as your Unix username
  - the -p option ensures you are prompted for a password
- Change password with
  - set PASSWORD=PASSWORD('YourNewPassword');

The MySQL Database System

- Tables created with CREATE TABLE command

```
CREATE TABLE Equipment (Equip_id INT UNSIGNED
      NOT NULL AUTO_INCREMENT PRIMARY KEY,
            Equip  CHAR(10) );
```

- To see the tables of a database:

```
SHOW TABLES;
```

- To see the description of a table (columns):

```
DESCRIBE Equipment;
```

- Other commands – INSERT, SELECT, DROP, UPDATE, DELETE same as SQL commands
- MySQL Reference Manual for version 4.0.13

http://support.cs.nott.ac.uk/help/docs/databases/mysql/standard/

Database Access with PHP/MySQL

- To connect PHP to a database, use `mysql_connect`, which can have three parameters:
  - Host (default is localhost)
  - Username (default is the username of the PHP script)
  - Password (default is blank, which works if the database does not require a password)
  - `$db = mysql_connect();`
  - Usually checked for failure
- Terminate the connection to the database with `mysql_close`
- Select a database with `mysql_select_db("cars");`
- http://severn.cs.nott.ac.uk/~bnk/dbConnect.php
  http://www.cs.nott.ac.uk/~bnk/WPS/dbConnect.pdf

Database Access with PHP/MySQL

- Requesting MySQL Operations
  - Call `mysql_query` with a string parameter, which is an SQL command

```
$query = "SELECT * from States";
$result = mysql_query($query);
```

- Dealing with the result:
  - The number of rows in the result

```
$num_rows = mysql_num_rows($result);
```

  - The number of fields in a result row

```
$num_fields = mysql_num_fields($result);
```

Database Access with PHP/MySQL

- Dealing with the result (continued):
  - Get the rows with `mysql_fetch_array`
  - Returns an array of the next row

```
for ($row_num = 1; $row_num < = $num_rows;  $row_num++) {
    $row = mysql_fetch_array($result);
    print "<p> Result row number" . $row_num . " State_id: ";
   print htmlspecialchars($row["State_id"]);
   print "State: ";
   print htmlspecialchars($row["State"]);
   print "</p>";
}
```

- http://www.cs.nott.ac.uk/~bnk/WPS/dbForm.html
- http://www.cs.nott.ac.uk/~bnk/WPS/dbQuery.pdf

Database Access with PHP/MySQL

- When values from a DB are to be put in HTML, you must worry about HTML special characters
  - E.g. "Apples & grapes <raisins too>"
  - To get rid of the HTML special characters, use the PHP function, `htmlspecialchars($str)`
  - Replaces the special characters in the string with their corresponding HTML entities
- Another problem with PHP and HTML forms is the string special characters (", ', \, and NULL), which could come from $_GET and $_POST
  - E.g. O'Reilly (value of textbox and returned in `$_POST`)
  - To fix these, `magic_quotes_gpc` in the PHP.ini file is set to ON by default
  - This backslashes these special characters

Summary

- Form Handling
- Files
- Cookies
- Session Tracking
- Architectures for Database Access
- The MySQL Database System
- Database Access with PHP/MySQL