CGT 141/CPT 141 Lecture 24 Wk 15
Server-Side Web Programming

PHP

What is PHP?

- PHP stands for Hypertext Preprocessor or Personal Home Page
- It is a server-side scripting language, scripts are executed on the server
- An open-source software that is free to download and use: www.php.net
- Supports many databases
- Cross-platform – able to develop and run on Windows, Unix, Linux, etc.

When was PHP created?

- PHP was created by Rasmus Lerdorf in 1994
- Evolved much since its creation
- Original version was created to handle user logs and server-side form generation
- The modules that make up PHP are written in C
- Was originally not a scripting language, simply a set of server tools to assist web maintenance
- Lerdorf thought that these tools may prove to be useful for others, and posted them on UseNET
- PHP has grown incredibly since then
    - Currently at version 4.3

What can PHP do?

- Connect to databases such as MySQL to display information dynamically
- Link to other web pages to display user information such as information entered into an HTML form
- Allow your browser to interact with files
- Allows for dynamic webpage creation
- Set and edit cookies
- Create graphical content such as graphs and charts
- Much more

What does a PHP file look like?

- Contains text, HTML and scripts
- PHP files are returned to the browser as plain HTML
- .php, .php3 or .phtml file extension

Examples of a PHP file

- There are two basic output statements in PHP

    - echo: <?php echo “Hello World”; ?>
    - print: <?php print “Hello World”; ?>
- PHP scripts start with <?php and end with ?>
- A block of script can be placed anywhere within the document
- Every line ends with a semicolon which is used to separate sets of instructions
- The example below “hello.php” prints the text “Hello World” in the browser

```
<html>
<head>
        <title> Short PHP example </title>
</head>
<body>
        <?php echo "Hello World"; ?>
</body>
</html>
```

- The next example uses variables to print to the web browser
    - The $ symbol specifies a variable
    - Variables may contain strings, numbers or arrays
    - The dot operator concatenates the two strings

```
<html>
      <title> Short PHP example </title>
<head>
</head>
<body>
      <?php
      $var1 = “PHP is fun.”;
      $var2 = “I love using PHP.”;
      echo $var1 . “  “ . $var2;
      ?>

</body>
</html>
```

- PHP is a very strong language. In addition to variables, PHP also offers the ability to use:
    - Arithmetic operators
    - Assignment operators
    - Comparison operators
    - Logical operators
    - Conditional statements
    - Switch statements
    - Looping

- Functions

How can you use PHP?

- To begin using PHP – download from www.php.net/downloads.php
- There are tutorials on the web which assist in downloading and installing PHP
- MySQL is also available for free, which can be used easily with PHP
  - www.mysql.com/downloads/index.html

A useful example of reading information from an HTML form
HTML form:

```
<html>
<head>
        <title> Enter Information </title>
</head>
<body>
        <form action = “information.php” method = “POST”>
        Please enter your first name:
        <input type = “text” name = “fname” />

        Please enter your last name:
        <input type = “text” name = “lname” />

        Please choose a number:
        <input type = “text” name = “num” />

        <input type = “submit” />
        </form>
</body>
</html>
```

The PHP file “information.php” may look something like this:

```
<html>
<head>
        <title> Receive Information </title>
</head>
<body>
        Hello, <?php echo $_POST[“fname”] . “ “ . $_POST[“lname”]; ?>. <br/>
        You have selected the number <?php echo $_POST[“num”]; ?>.
</body>
<html>
```

This would show up on your browser somewhat like this (depending upon what is entered):

Hello, John Doe.
You have selected the number 6.

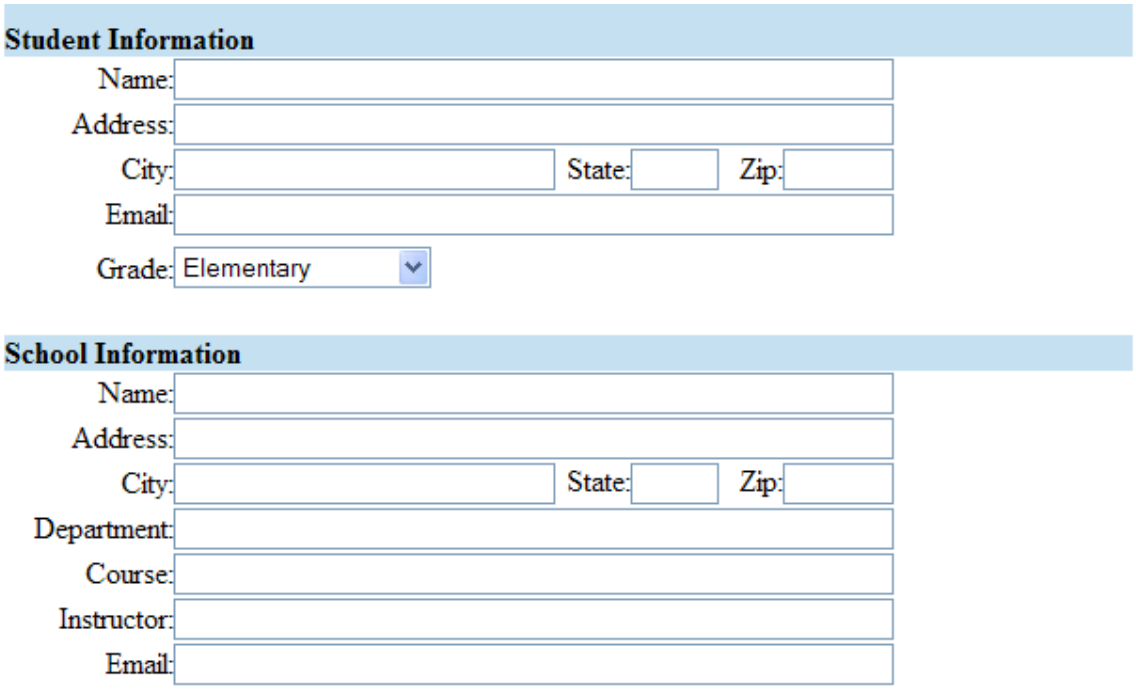
- The $_POST variables automatically collect all of the information that was entered in the form. PHP automatically sets these variables for you.
- If the method of the form is "GET", then the information will be stored by PHP in the $_GET variables rather than $_POST

How PHP is beneficial compared to others (ASP, JSP, ColdFusion)

- PHP is not limited to certain operating systems or web servers
- PHP supports a wide range of databases
- Open source
  - Can add to PHP's abilities to enhance and personalize if needed
  - Cost (JSP is also open source)
- Speed, PHP is run very quickly compared to ASP
- Same basic functionality as others but drawbacks to others are:
  - JSP is not widely supported by web hosting companies not running Sun, Oracle, etc.
  - ASP is widely supported by web hosting companies (Microsoft based) but unless you are able to run your own COM objects, you are limited to VBScript
- PHP offers wide flexibility

Additional useful information about PHP can be found at:
www.w3schools.com
www.php.net

You may see a form such as this:



And then use code similar to this in order to grab the information from the form:

```
<html>
<head>
<title>Successfully Uploaded</title>
</head>

<?php

// Grab Student info from Form post
$StudentName        = $HTTP_POST_VARS["StudentName"];
$StudentAddress     = $HTTP_POST_VARS["StudentAddress"];
$StudentCity        = $HTTP_POST_VARS["StudentCity"];
$StudentState       = $HTTP_POST_VARS["StudentState"];
$StudentZip         = $HTTP_POST_VARS["StudentZip"];
$StudentEmail       = $HTTP_POST_VARS["StudentEmail"];
$StudentGrade       = $HTTP_POST_VARS["StudentGrade"];

// Grab School info from Form post
$SchoolName         = $HTTP_POST_VARS["SchoolName"];
$SchoolAddress      = $HTTP_POST_VARS["SchoolAddress"];
$SchoolCity         = $HTTP_POST_VARS["SchoolCity"];
$SchoolState        = $HTTP_POST_VARS["SchoolState"];
$SchoolZip          = $HTTP_POST_VARS["SchoolZip"];
$SchoolDepartment   = $HTTP_POST_VARS["SchoolDepartment"];
$SchoolCourse       = $HTTP_POST_VARS["SchoolCourse"];
$SchoolInstructor   = $HTTP_POST_VARS["SchoolInstructor"];
$SchoolEmail        = $HTTP_POST_VARS["SchoolEmail"];

?>
<body bgcolor="#C5E1F1">
<p> </p>
</body>
</html>
```