COSC 360
Lab 8 – Arrays and Superglobals in PHP

Information:

Submit all your files file through GitHub classroom and the paste your repository link in Canvas. You will need to use your local web development stack to complete this lab.

You can find further practice and info for PHP at `https://www.w3schools.com/php/`.

Instructions:

In this lab, you will build two pages that can be used to examine the basic of information sharing between the client and server. Previously, you have seen how a HTML form can be used to transmit data from the client to the server, but have not been able the handle it on the server side. This lab will require the completion of two different pieces of code. The first will be the test page which will contain a form that will be used to submit data to the server and the second will be a PHP page that will receive and process the requested, ultimately generating content based on data passed to in. The two pages will allow you to submit a `firstname` and `key`, look the results up in a collection (which is loaded from a file) and display an associated image demonstrating how to use files to create dynamic content without having create multiple pages.

1. Using the invitation link for lab 8, clone the repository for Lab 8 to your local machine. Make sure that you place your completed files back in the repo folder when committing and pushing your final lab results. To ease development, you may want to check out the lab into your `htdocs` folder so that XAMPP will be able to host and render the files. If your files are not in the `htdocs` folder, XAMPP will not be able to process the php files.

2. Edit the HTML file called `lab8-1.html`. Create a simple form as shown in Figure 1 that will allow you to enter a name and key value. When the submit button is clicked, the form will be submitted to `lab8-1_action.php` (which you will need to also create) using a `GET` request. Complete the `Lab8-1.html` file. The name of the name input field should be `firstname` and the name of the key field should be `key`. Both are text input types. Test your page to ensure that it is generating the correct query string.

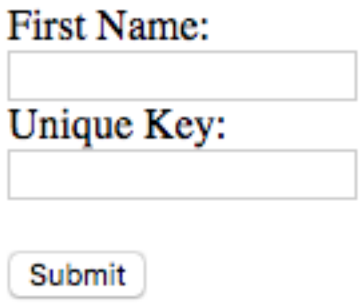


Figure 1 - lab7-1.html

3. Create a new PHP file called `lab8-1_action.php`. This will be the file that will receive and process the results and generate the appropriate content. In order to receive data from a request, you will need to use the appropriate `superglobal` variable (p.548 in second edition/ p. 373 in first edition). This week, you will need to use the `$_GET`, `$_POST` and `$_SERVER` variables (as appropriate). The `$_GET` will allow you to access data from a GET request while `$_POST` will allow you to access data from a POST. Data sent to the server in this fashion is accessed using the field name as the key. To test

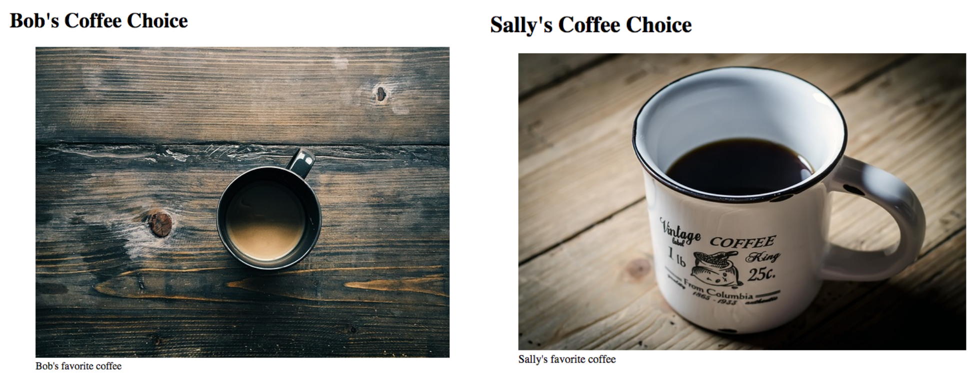
your initial file, have your file echo out the contents of the `firstname` and `key`. Review p.548 (p. 374 in 1st edition) on details for using superglobals. Test your files to ensure data is being transmitted correctly.

4. One of the conditions that needs to be handled is the case where no data is transmitted to a page when it is expecting data. In your PHP file, if you try to print out data for a variable that does not exist, it will generate an error. Modify your code so that it will only print out the values for `firstname` and `key` if the values are set. Review p. 549 (p. 375 in first edition) for details of the method required.
5. Data can be sent either with a POST or a GET. Modify your PHP file to be able to handle the data being sent with either method. You will be able to determine this using the `$_SERVER` superglobal (details on p. 559 (p. 377 in 1st edition). Test to ensure that it will be able handle both a POST and GET from your form (by changing the form method).
6. Store the `firstname` and `key` in variables so that you are able to use them in your code to generate dynamic content. A text file has been provided for you called data.txt contains `key`, `firstname`, `caption` and `image_path` for two users. Modify your PHP code so that the file is read into memory using in-memory file access. The file formation is:

   ```
   <key>,<name>,<caption>,<image_path>

   example:
   0101, Bob, Bob's favorite coffee, images\coffee1.jpg
   ```

   The goal is to be able to access the contents of the file as an array, thus you will need to select the correct in-memory file function. You will need to be able to use the array to check if a key is present, in addition to checking if the name is correct (valid, case-insensitive match). If it is, display the image located at the indicated path formatted as a figure with caption. The image files have been provided for you in the starter files. (Hint: you will want to store the data as an array of arrays where the first key is the numeric key and stored in a given position is a second collection containing the relevant user and image information. This will require you manipulating the data to explode each line of the file into a collection using the explode function. Details on p 569-570 (p. 394 1st edition). You may want to add some debug statements to print out the contents on the array you are building. Use the function print_r() to pretty print the contents of your array for debugging purposes.
7. Modify your PHP code such that it will display the users name, along with the image and caption on the page as shown in Figure 2. The heading should be an <h1> element with the image a <figure> with <figcaption>. If the user and/or key is invalid or not provided, display a message indicating that. You will need to examine how to compare strings with PHP (Hint: you can't just use the == operator). You will have to search for an appropriate PHP method that can handle this in a case-insensitive fashion). Test your code to ensure that both users images and captions display correctly. **The image and user information must be dynamically generated from information in the data.txt file.** This will allow for the addition of users and images without having to modify the PHP itself. Figure 2 shows examples of output.


Bob's Coffee Choice

Bob's favorite coffee

Sally's Coffee Choice

Sally's favorite coffee

Figure 2 - Rendered HTML

Make sure you add your new files to your repository. Commit your HTML and PHP to your repository and push back to GitHub as well as submit your repository url via canvas.