Assigned:	2013-10-10
Due:	2013-10-17

Requirements

For this project, you will need to write at least two pages: One for selecting files to upload, and one for displaying the images that have been uploaded. You will use a form element, with at least one input element whose type is "file".

You can style the pages if you would like, but it is not required.

index.php page

This page can be very simple – you need a form, a submit button, and a file uploading input. You can use the following, or make your own:

```
<form method="post" enctype="multipart/form-data" action="gallery.php5">
      <input type="file" name="image[0]" />
      <input type="file" name="image[1]" />
      <br/>
      <input type="submit" name="new-image" value="Add Image"/>
</form>
```

Note that we have added **enctype** and **action** to the form tag. The Action points to another webpage that will be called on the form submission. The enctype tells the form that we will have multiple types of items submitted – This will result in having a $_POST array and a $_FILES array.

gallery.php page

The gallery page will need to load in images from the "images/" folder on the server and display them with a foreach loop. This may be similar to:

```
<? foreach( $imageArray as $path ) { ?>
      <img src="<?=$path?>" />
<? } ?>
```

Where $imageArray is an array we have built in the back-end scripts.

PHP Scripts

When I wrote this, I had separate scripts for each set of functionality:

- imge-loader.php5 – Gives the gallery page a list of files to draw

- uploader.php5 – Takes the $_FILES data and does the necessary error checking and uploads the files to the “images/” folder.

File Uploader

Your file uploading script needs to do the following:

- Check the file **type** to make sure it is a valid type (jpg, jpeg, gif, png)
- Check the file **name** to make sure it has a correct extention (.jpg, .jpeg, .gif, .png)
- Check the file **size** to make sure it does not go over some limit (this can be 30,000 bytes or another number that you choose).
- If there are no errors, copy the file from the /tmp/ directory to the images/ folder.
- Output error messages appropriately if there is a problem.

When the Submit button is hit on the form, your $_FILES array will look something like:

```
Array
(
    [image] => Array
        (
            [name] => Array
                (
                    [0] => unsure.png
                    [1] => index.jpg
                )

            [type] => Array
                (
                    [0] => image/png
                    [1] => image/jpeg
                )

            [tmp_name] => Array
                (
                    [0] => /tmp/phppnCOPG
                    [1] => /tmp/phpzfy8FC
                )

            [error] => Array
                (
                    [0] => 0
                    [1] => 0
                )

            [size] => Array
                (
                    [0] => 27429
                    [1] => 9512
                )

        )

)
```

Note that each key of the array is a different property of a file, and within these properties there is a numbered array of our different images (two have been uploaded in this example).

You may want to build a better array format, like:

```
[0] => Array
        (
            [name] => unsure.png
            [type] => image/png
            [tmp_name] => /tmp/phpqgqwSn
            [error] => 0
            [size] => 27429
        )
```

This is done in the sample code for Image Upload on the class website.

Image Loader

The image loader script should search the "images/" directory and keep an array of any files that have .jpg, .jpeg, .gif, or .png.

You can use the **scandir** function of PHP to get a list of files. See the "Resources" page of this document for links to useful functions.

PHP Layout

Note that in my Image Upload example, I am not using classes, and only a few functions. This is because, when learning a new concept, it's just easier to understand when the process isn't hidden behind abstraction and objects.

Keep your PHP code clean. Use classes and functions as you see fit. You could have an "image uploader" class that knows where the images go and has functionality for each step: Verifying file type, file size, what kind of errors there are, moving the file over, aggregating any error messages, displaying error messages, etc.

Resources

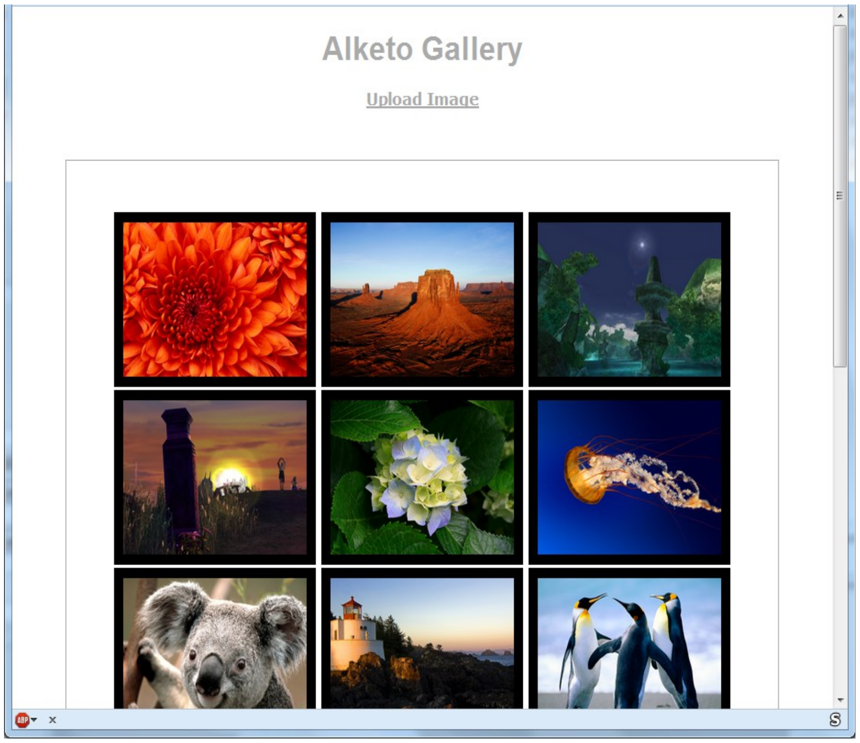
The class webpage has file uploading and file system samples, under the "Sample Code/PHP/Image Upload/" folder.

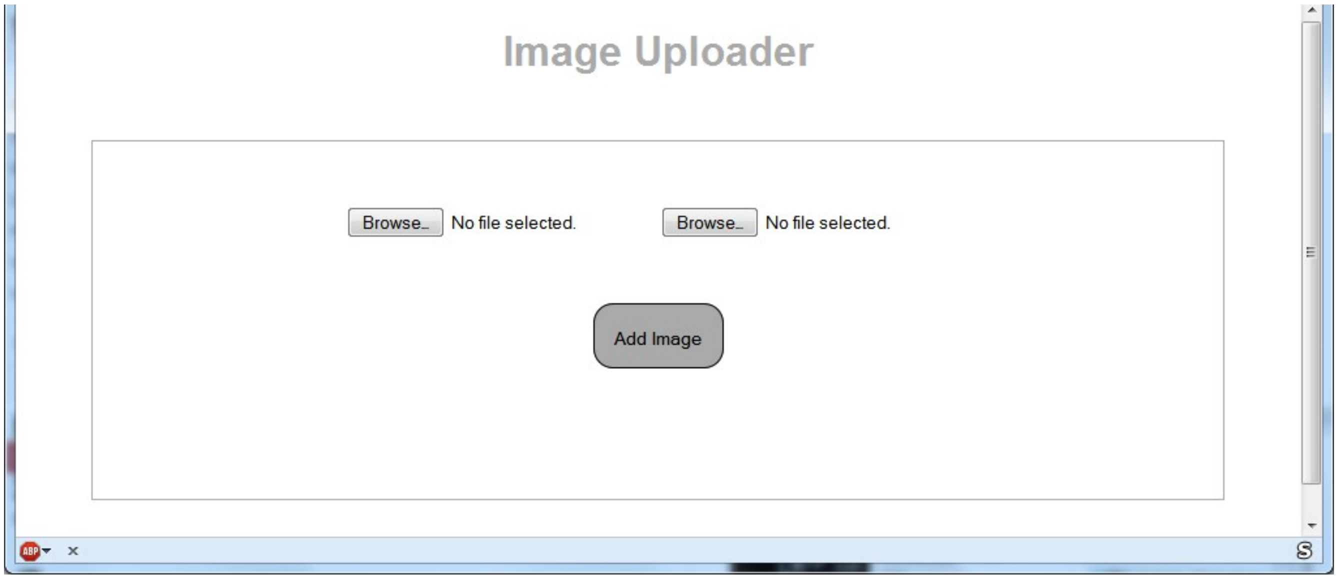
Other useful links:

- http://php.net/manual/en/function.mkdir.php
- http://php.net/manual/en/function.file-exists.php
- http://php.net/manual/en/function.scandir.php
- http://www.w3schools.com/php/php_file_upload.asp

- http://www.htmlgoodies.com/beyond/php/article.php/3877766/Web-Developer-How-To-Upload-Images-Using-PHP.htm

Sample Images





Grading Rubric (10 points total)

Error Checking and Handling = 2 pts

- All error cases are checked for
- Error messages are generated and displayed

File Uploading = 2 pts

- Files are uploaded to the proper directory when there are no errors
- Images/Files not uploaded if there are errors

Gallery = 2 pts

- All .jpg, .jpeg, .gif, and .png files in the "images/" folder are displayed

Coding Style = 2 pts

- Everything is easy to read
  - I won't grade CSS since it is optional
- Proper XHTML
  - No deprecated tags or attributes, all required tags present.