DIG4503 **Assignment 2**

Due By: *03/19/12 11:59pm*
Delivery Method: Your assignment folder should be ~/public_html/dig4503/assignment2

Overview

Design and develop a Flickr photo viewer with the Flickr API that uses PHP and jQuery. Your photo viewer will return a set of photograph thumbnails that correspond to the search term entered by the user. Once the user clicks on a photo thumbnail, a larger version of the photo should be displayed on the same page.

Specifications

General

- Code should be indented to show element parent/child relationships.
- The root directory for this assignment should be **~/public_html/dig4503/assignment2**
- The image directory for this assignment should be **~/public_html/dig4503/assignment2/img**
- You should have one XHTML page and two PHP scripts in the root folder for this assignment. The XHTML page should be named **viewer.html**, the first PHP script should be named **getThumbs.php** and second PHP script should be named **getPhoto.php**.
    - **viewer.html** should contain valid XHTML and be the only page that the user loads via a traditional HTTP request.
    - **getThumbs.php** should contain a PHP script that performs a cURL request for data from Flickr using the flickr.photos.search REST API and should return an XML response.
    - **getPhoto.php** should contain a PHP script that performs a cURL request for data from Flickr using the flickr.photos.getInfo REST API and should return an XML response.
- The <title> of **viewer.html** should be "DIG4503 – first_name last_name Assignment 2". For example, "DIG4503 – Jon Friskics – Assignment 2".
- No **spaces or uppercase letters** should be used in file names or directory names.

XHTML Validation

- **viewer.html** should validate as XHTML Strict 1.0 (http://validator.w3.org).
- **NOTE:** if you plan to use an HTML5 doctype, then please use unobtrusive JavaScript to insert an HTML5 badge or ribbon image that is absolutely positioned in the top-right corner of your page.

CSS

- All styles must be documented in an external CSS file called **styles.css** and linked to the document using the <link /> element in the <head> of the document.
- All presentational HTML attributes should be replaced with CSS rules.
- You may (and are encouraged to) use a reset stylesheet.
- You may design your site on a grid and implement it using the 960.gs grid system.
- The CSS directory for this assignment should be **~/public_html/dig4503/assignment2/css**

General – JavaScript

- The JavaScript directory for this assignment should be **~/public_html/dig4503/assignment2/js**
- All JavaScript code must be documented in an external JavaScript file named **a2.js** and linked in the <head> of **viewer.html**.
- You should only use unobtrusive JavaScript, and no JavaScript should appear in your HTML outside of the <head> element.
- No jQuery plugins should be used in this assignment (patience please!)
- You **must** use a jQuery library hosted from Google's CDN and should attempt to use as much jQuery as possible vs. standard JavaScript.

Behaviors – Image Loading

- When the user visits **viewer.html**, they should be prompted to enter a search term. Once they have entered the term and hit the submit button, an XHR request for **getThumbs.php** should occur that performs a cURL request of the Flickr API that returns 40 results for photos that match the search term entered by the user as XML.
- That XML response should be parsed using one of the following methods:

  OUTPUT 1 – XML to PHP
  - **getThumbs.php** should loop through the XML response and output a set of <img /> elements that contain thumbnail URLs as the **src** attribute and the photo_id as the **alt** attribute of each image ONLY for images that were taken on a Monday.

  OUTPUT 2 – XML from PHP
  - **getThumbs.php** should return an XML object to a2.js, and a2.js should loop through the XML response and dynamically generate a set of <img /> elements that contain thumbnail URLs as the **src** attribute and the photo_id as the **alt** attribute of each image ONLY for images that were taken on a Monday.

Behaviors – Image Filtering

- **getThumbs.php** should use **date_taken** information embedded in each photos' XML response to determine the day of the week that the photo was taken. Only photos taken on a Monday should be allowed to continue past the image filtering process.

Behaviors – Full-size Image Retrieval

- **getPhoto.php** should perform a cURL request of the Flickr API that returns a single photo using the **photo_id** information returned by getThumbs.php to display a full-sized image with photo author, photo title, and date taken metadata.

Presentation

- Each photo thumbnail in **viewer.html** should be displayed using an unordered list element and photo thumbnails should be arranged in a grid-like pattern.
- The width of the entire layout **should not exceed 960px**. A sample mockup is included in the assignment folder for inspiration. You are encouraged to design a layout that breaks the conventions illustrated in this wireframe.

Content

- All content outside of layout-related assets should come from Flickr API requests.
- 40 images should be returned from each search request.
- Each full-sized image should be accompanied by text elements indicating the **photo author, photo title,** and **date the photo was taken** in the format MonthName DayNumber, Full Year
  - ex: February 18, 2012

Rubric

Implementation General specs followed (-10 per error)
Implementation XHTML spec followed (-10 per error, -5 per warning)
Implementation CSS specs followed (-10 per error)
/20 Site Design + Presentation specs followed
/80 Behavior Scripting
- /30 Thumbnail images load into viewer.html as requested
- /20 Only images taken on a Monday are displayed as requested
- /30 Full-size image retrieval works as requested

-100 if all requested site content is not included
-100 if any XHTML tables (for site layout), frames, or iframes are used
-100 if jQuery is NOT used for most JavaScript tasks
-100 if .phps files are not included for each .php file in your assignment folder (excluding .php files with sensitive password information)

/100 TOTAL