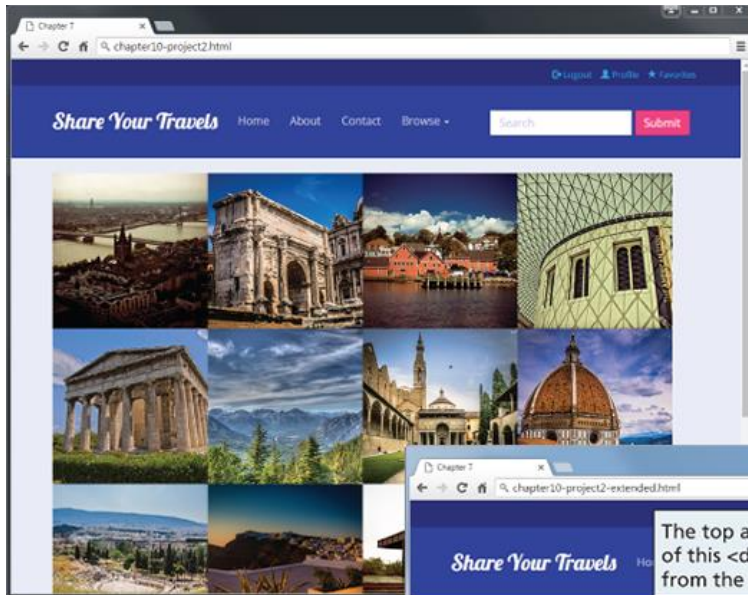


## CS 6314.001/002 – Assignment 2

**Due Date:** February 24, 2020, 11:59PM

This project will build a photo gallery using jQuery for your travel photo sharing site as shown in following picture.



The `images.js` file contains an array of image objects (examine `data.json` to see all the formatted data).

Loop through this array outputting the appropriate `<img>` tags as list items within the provided `<ul>` element. The `alt` attribute should be set to the `title` property of the image object.

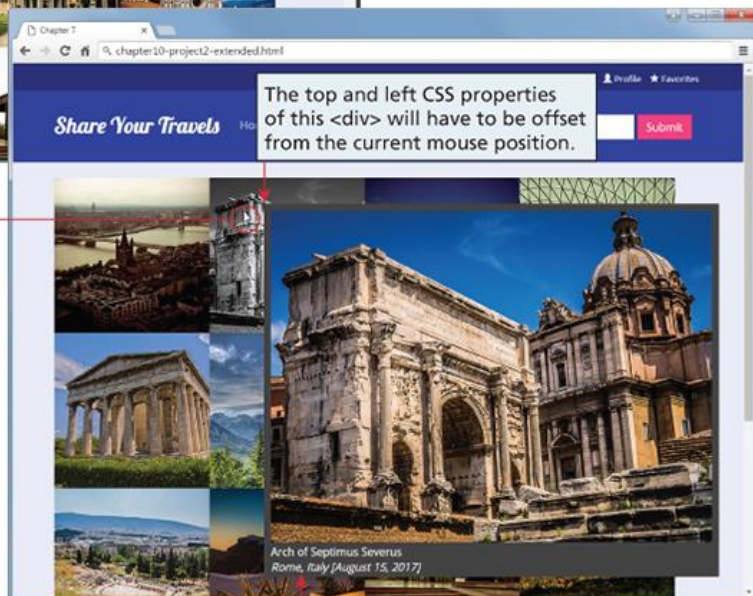
You are going to create handlers for the `mouseenter`, `mouseleave`, and `mousemove` events of each of these images.

The `mouseenter` handler will add the class "gray" to the moused over image.

The `mouseenter` handler will also add a `<div>` with `id="preview"` that will contain a larger version of image and a caption. This preview `<div>` should also be faded in over 1 second.

The `mouseleave` handler will have to remove the gray class and remove the preview `<div>`

The `mousemove` handler will have to recalculate the top and left CSS properties based on the mouse position.



The top and left CSS properties of this `<div>` will have to be offset from the current mouse position.

The caption displays information for the image.

## Requirements

When the document is loaded, you will make an AJAX call, download the data.json file (it contains an array of image objects) and display images on your html page. When the user hovers over the image, you will show the larger version of the image along with the caption. See instructions for details.

## Instructions

1. Examine [HW2.html](#) in the browser and then editor. You have been supplied with the appropriate CSS (the relevant classes are in [gallery.css](#)), html, and JavaScript data files ([data.json](#) which contains the data in an easy-to-read format). The images are supplied in two folders: [images/square](#) (for the gallery) and [images/medium](#) (for the popup).
2. Loop through the images array and using the appropriate jQuery DOM methods, add the appropriate `<img>` tags to the supplied `<ul class="gallery">` element. The image filenames are contained in the `path` property of each image object. Set the `alt` attribute of each `<img>` to the `title` property of the image object.
3. Use jQuery to attach handlers for the `mouseenter`, `mouseleave`, and `mousemove` events of the square images in the gallery.
4. For the `mouseenter` event, use jQuery to add the "gray" class to the square `<img>` under the mouse. If you examine that class, you will see it sets the `filter` property to `grayscale()`. Hint: remember that `$(this)` within an event handler references the DOM object that generated the event.
5. Also for the `mouseenter` event, use jQuery to generate a `<div>` with an `id="preview"` (the styling for `#preview` is already defined in [gallery.css](#)). Within that `<div>` add an `<img>` element that displays the larger version of the image. Underneath that `<img>` add a `<p>` element for the caption (image title, city and date taken). The `alt` attribute of the square image under the mouse contains the image title.
6. You will need to use jQuery to set the `left` and `top` CSS properties for the `#preview<div>`. You can retrieve the x, y coordinates (via the `pageX` and `pageY` properties) of the current mouse position from the event object that is passed to your event handler. You can calculate the new position by offsetting by some amount from the mouse x, y position.
7. Finally, once the `#preview <div>` is constructed, simply append it to the `<body>`.
8. For the `mouseleave` event, remove the "gray" class from the square image under the mouse. Also remove the `#preview<div>` from the body.
9. For the `mousemove` event, simply set the `left` and `top` CSS properties for the `#preview <div>` using the same approach as described in step 6.

## **Testing**

1. Verify the code works when mousing over the images. Be sure that the caption is displaying the correct information.
2. Don't worry if the pop-up image is “off screen” when mousing over images on the edges of the browser.

## **Deliverables:**

Screenshot of the results

HTML and JS files

Zip your files into a single file and name it as HW2-yournetid.zip.