Project 4: Interactive Photo Gallery

Sections of this Guide:

- **How to approach this project** includes detailed guidance to help you think about how to organize your code, project and files.
- **How to succeed at this project** lists the grading requirements for the project, with hints, links to course videos to refresh your memory and helpful resources.

How to Approach this Project

- ❑ **Download the project source files from the Interactive Photo Gallery project instructions page in your Techdegree curriculum.**
- ❑ **Create GitHub repo and push project files - Share Your Projects with GitHub**
- ❑ **Set up the rest of the directory by creating the necessary files and folders.**
  - ❑ Along with the project source files, you'll need at least a js folder with a scripts.js file, css folder with a styles.css file, and an index.html file.
- ❑ **Create the basic markup structure and styling for your gallery.**
  - ❑ There are many ways to structure the HTML for this project. You could use an unordered list for all of the elements in the gallery, but perhaps the simplest approach is best. That might look something like this: **-->**

```
-- body
  -- wrapper - a main or div tag
    -- input - for the search box
    -- inner wrapper - a div or section tag
       And then for each item in the gallery:
       -- a tag - this links to the larger photo
       that will be displayed in the lightbox
       -- img tag - this will display the
       thumbnail
```

- ❑ **Hookup the lightbox plugin**: *details below in the "How to Succeed at this Project" section.*
- ❑ **Create the search functionality**: *details below in the "How to Succeed at this Project" section.*
- ❑ **Double check everything, validate your files, request an informal review in Slack, and then submit.**

Need help? Visit the Project 4 Slack Channel

How to succeed at this project

Here are the things you need to do pass this project. Make sure you complete them ***before*** you turn in your project.

❑ Design

Make sure your design matches the layout of the mockups.

- ❑ Project includes the thumbnail gallery and a search box.
- ❑ The layout doesn't have to be pixel perfect, but general spacing, position and arrangement of the elements should closely match the layout of the mockup.
- ❑ **NOTE:** For an example of what markup to use for the items in the gallery, see the bullet point, **Create the basic markup structure and styling for your gallery,** in the **How to approach this project section** above.
    - ❑ *Related video:* **Debugging CSS with Chrome Dev Tools**
    - ❑ *Related video:* **Practice the Box Model**
    - ❑ *Related video:* **CSS Flexbox**
    - ❑ *Related video:* **CSS Grid Layout**

❑ Lightbox functionality

Implement a plugin for a lightbox that includes ALL of the following features:

- ❑ When gallery thumbnail image is clicked, larger image appears with overlay.
- ❑ Background overlay hides the gallery and covers the entire window.
- ❑ Images in lightbox include full captions.
- ❑ Lightbox images can be navigated by clicking right and left arrows.
- ❑ The lightbox can be closed.
- ❑ There are a number of lightbox plugin options you can use, but here is a suggested one: **Lightbox 2 Plugin**. It's fairly common, has good documentation, and is relatively easy to use.
    - ❑ *Related video:* **Using jQuery Plugins**
    - ❑ *Recommended Plugin Link:* **Lightbox 2 Plugin**
- ❑ **TIP**: Instead of downloading the lightbox source files which would have to be included in your project directory and linked to correctly in your HTML, it is much simpler to just use the CDN links to connect your project to the plugin files.
    - ❑ *Lightbox plugin CDN link*: **CSS**
    - ❑ *Lightbox plugin CDN link*: **JS & jQuery** - **Note**: This link also hooks up jQuery to your project, so you don't need a separate link for that.

Need help? Visit the **Project 4 Slack Channel**

❏ Content Filtering

Implement a plugin or write your own JS for the search functionality:

- ❏ Gallery photos update in real time as the user types into the search box, only photos that match the caption text appear in the gallery.
- ❏ Case insensitivity has been added so searches will ignore letter case.
- ❏ Entire caption is searchable, not just title or keywords.
- ❏ As with the lightbox, there are many plugins for implementing search functionality. Here's a popular one that is relatively easy to hook up:
    - ❏ *Hideseek plugin home page:* **Hideseek Search Plugin**
    - ❏ *Hideseek plugin docs:* **Example of Setting Up Hideseek**
    - ❏ *Hideseek plugin CDN link:* **Hideseek CDN link**

However, accomplishing the search functionality with your own code is an excellent bit of web dev practice and experience, so feel encouraged to give it a try. Just break the task down into small pieces.

First try to get the value of the search field whenever a user types in that field, and log that value to the screen with a console.log() statement. Checkout the **keyup event listener** for this. And be sure to make the value case insensitive by using something like the **toLowerCase()** method.

Next, start trying to target the entire caption by using the **getAttribute()** method, then **looping** and logging the all lowercase version of the captions to the console.

Next comes the fun part. Start trying to find ways to check if the current value of the search input is **included** within any of the captions, and if so, log the associated image to the console.

Lastly, use a **conditional** so that if there's a match, **display** the container of the image, and if not, **hide it**. Piece of cake!

Helpful links:

- ❏ *Related video:* **Debugging JavaScript in the Browser**
- ❏ *Related video:* **Practice Variables, Input & Output**
- ❏ *Related video:* **Practice Selecting DOM Elements**
- ❏ *Related video:* **Practice Traversing the DOM**
- ❏ *Related video:* **Practice Manipulating the DOM**
- ❏ *Related video:* **Practice If and If Else Statements**
- ❏ *Related video:* **Practice Loops**
- ❏ *Related video:* **Practice Functions**

Need help? Visit the **Project 4 Slack Channel**