

Assignment #2: Javascripts + JQuery

Due October 26 (Friday) by 11:59PM.

There are three projects in this assignment. Follow the instructions for each project to complete them. You have been provided with the necessary files to complete these projects.

Project 1: *Photo Sharing Site*

Overview

Demonstrate your ability to create JavaScript objects and arrays as well as work with inner functions. The final project will look similar to that shown in [Figure 1](#).

Instructions

1. In the `project1` folder, you have been provided with the HTML file ([project01.html](#)) that includes the markup (as well as images and stylesheet) for the finished version. Preview the file in a browser. You will be replacing the markup for the four country boxes with a JavaScript loop.
2. In the file `data.js`, create an array named `countries` that contains four object literals. Each object literal should contain four properties: `name`, `continent`, `cities`, and `photos`. The `cities` and `photos` properties should be arrays containing the city names and image filenames respectively.
3. In the file `functions.js`, create a function named `outputCountryBox()` that has the signature shown below. This function is going to generate the markup (using `document.write`) for a single country box.

```
function outputCountryBox(name,continent,cities,photos)
```

4. Inside the `outputCountryBox()` function, create two inner functions named `outputCities()` and `outputPhotos()`. These two functions will have the responsibility to generate the markup for the cities and photo boxes.
5. Replace the markup for the country boxes with a loop through your array of countries. Within this loop, call your `outputCountryBox()` function, passing it the relevant data from the country objects.

Test

1. Test the page in the browser. Verify the correct data is displayed.

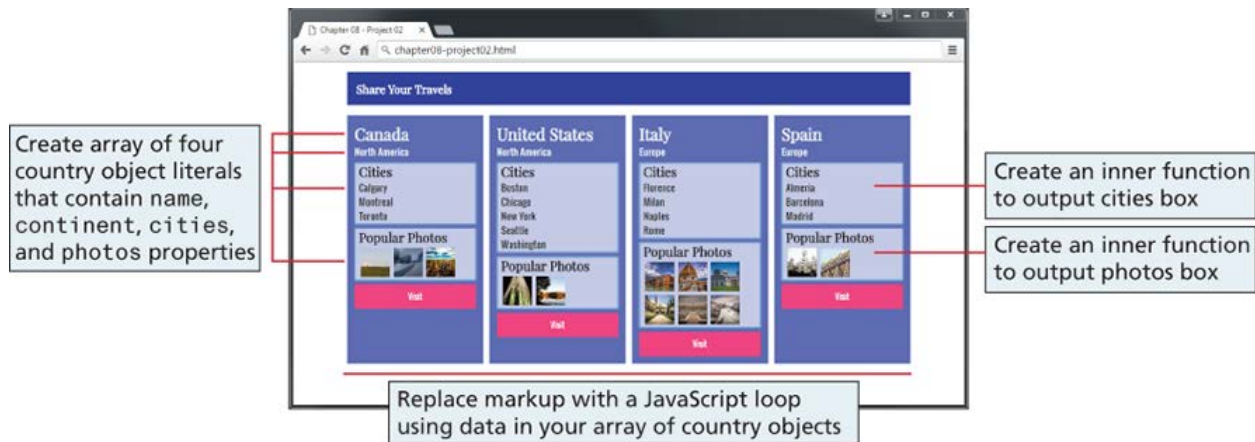


Figure 1 Completed Project 1

Project 2: *Art Store (Part 1)*

Overview

You will demonstrate your ability to respond to events, select and modify elements via the DOM, and to validate form data.

Instructions

1. In the `project2` folder, you have been provided with the HTML file (`project02.html`) that represents the data entry form shown in [Figure 2](#). Examine this file in browser.
2. You will notice that some of the form elements have the CSS class `highlightable` specified in their `class` attribute. Add listeners to the `focus` and `blur` events of all elements that have this `highlightable` class. In your event handlers for these two events, simply toggle the class `highlight` (which is in the provided CSS file). This will change the styling of the current form element. Be sure to set up these listeners *after* the page has loaded.

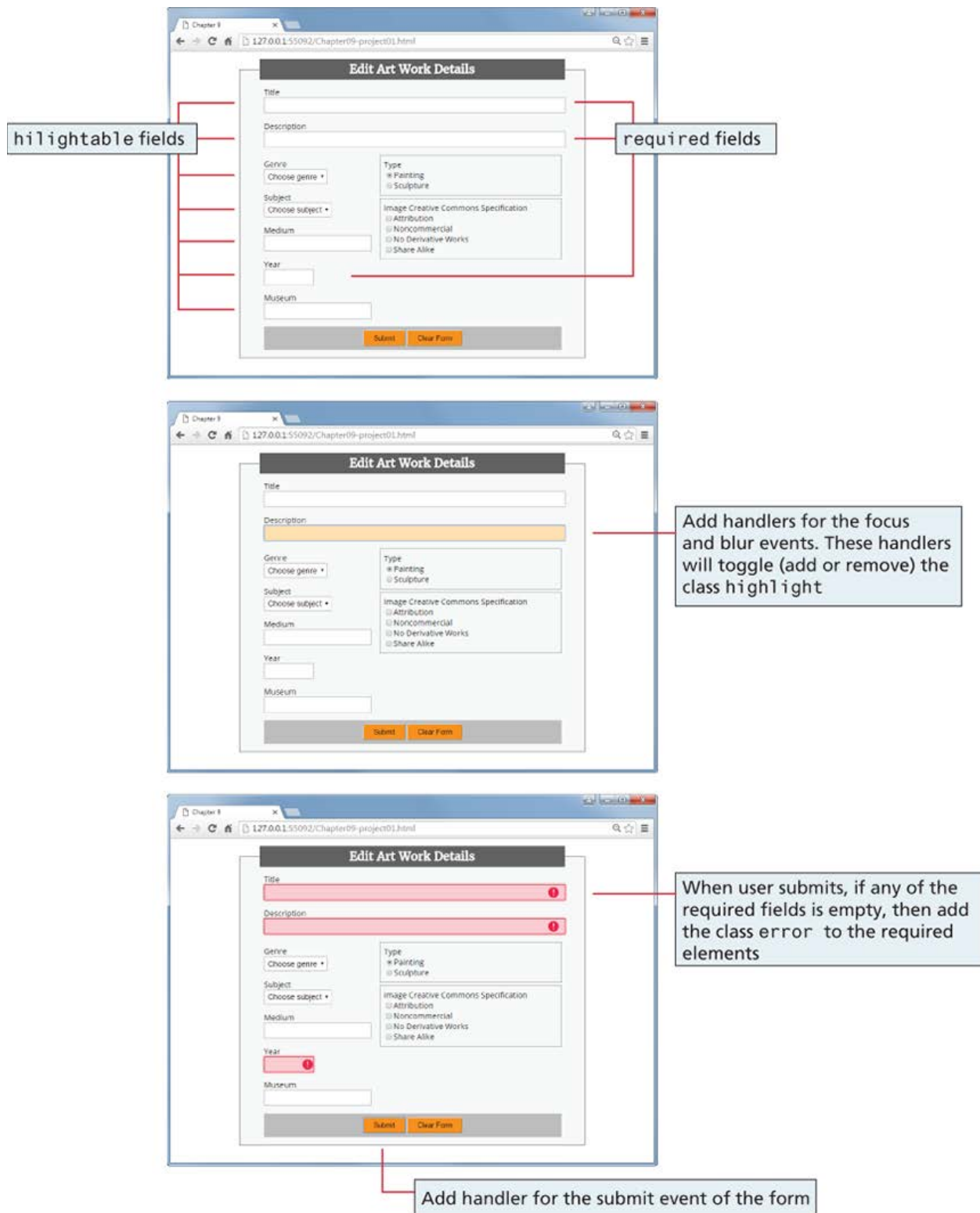


Figure 2 Finished project 2

3. You will notice that some of the form elements have the CSS class `required` specified in their `class` attribute. We will not submit the form if these elements are empty. Add an event handler for the submit event of the form. In this handler, if any of the required form elements are empty, add the CSS class `error` to any of the empty elements. As well, cancel the submission of the form (hint: use the `preventDefault()` method).
4. Add the appropriate handler for these required controls that will remove the CSS class `error` that have changed content.

Test

1. Test the form in the browser. Verify the highlighting functionality works by tabbing from field to field. Try submitting the form with blank fields to verify the error formatting works. Verify the error formatting is removed if you add content and then resubmit.

Project 3: *Art Store (Part 2)*

Overview

Use jQuery to respond to events and to programmatically modify HTML and CSS as shown in [Figure 3](#).

Instructions

1. Examine [project03.html](#) in the browser and then editor. You have been supplied with the necessary CSS and HTML.
2. Import jQuery in the `<head>` of the page.
3. Use jQuery to respond to click events on the painting thumbnails. Replace the `src` attribute of the `` element in the `<figure>` so that it is displaying the clicked painting. Hint: get the `src` attribute of the clicked element and then replace the `small` folder name with `medium` folder name.
4. As well, change the `<figcaption>` so that it displays the newly clicked painting's title and artist information. This information is contained within the `alt` and `title` attributes of each thumbnail.
5. Set up event listeners for the `input` event of each of the range sliders. The code is going to set the `filter` and the `-webkit-filter` properties on the image in the `<figure>`. Recall from Chapter 7 that if you are setting multiple filters, they have to be included together separated by spaces.
6. Add a listener for the click event of the reset button. This will simply remove the filters from the image.

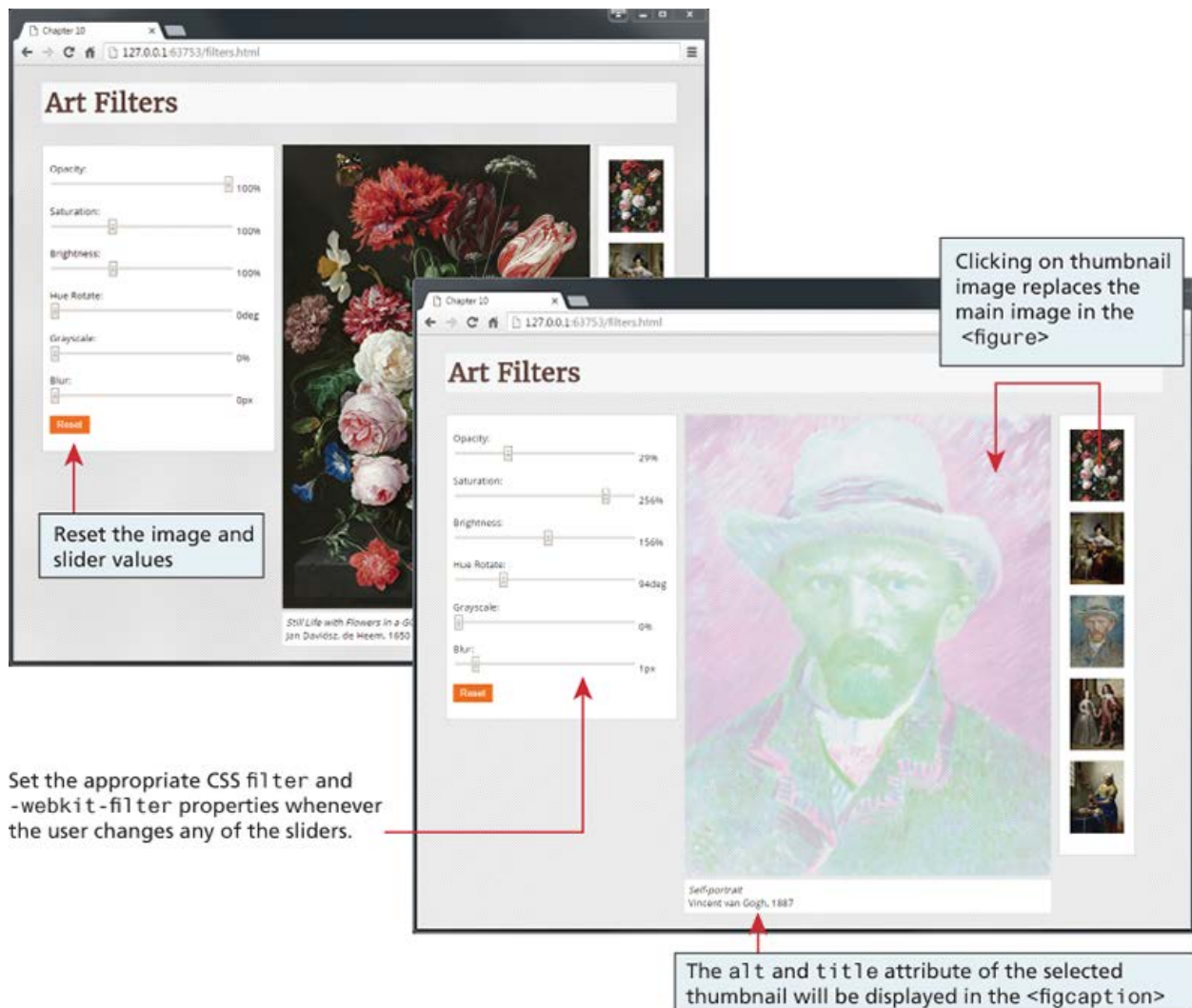


Figure 3 Completed Project 3

Testing

1. To test, click on the thumbnails and verify the correct caption is displayed. Ensure the filters work as expected.

Submitting

Put your assignment in a folder named `assign2_yourname` (e.g., `assign2_johnsmith`). Put all resources used by your assignment into this folder and compress them into a zip file.

Upload the `assign2_yourname.zip` file to the Assignment 2 submission folder under dropbox on elearning.