PROJECT 2: Share Your Travel Photos

DIFFICULTY LEVEL: Intermediate

HANDS-ON EXERCISES
Project 9.2

Overview

You will demonstrate your ability to use the DOM and to handle events using both event delegation and "regular" event handling.

Instructions

1. Examine **chapter09-project02.html**. You will add event handlers to the thumbnail images and to the larger image. You will not need to make any changes to the supplied markup or CSS.
2. All of your event handlers must execute only after the page has loaded.
3. You are going to add a click event handler to each of the thumbnail images. When the smaller image is clicked, your code will show the larger version of the image in the `<img>` element within the `<figure>` element. This same event handler will also set the `<figcaption>` text of the `<figure>` to the clicked thumbnail image's title attribute. Your event handler must use event delegation (i.e., the click event handler will be attached to the `<div id="thumbnails">` element and not to the individual `<img>` elements).
4. You must also add event handlers to the `mouseover` and `mouseout` events of the `<figure>` element. When the user moves the mouse over the larger image, then you will fade the `<figcaption>` element to about 80% opacity (its initial CSS opacity is 0% or transparent/invisible). When the user moves the mouse out of the figure, then fade the `<figcaption>` back to 0% opacity. You can set the opacity of an element in JavaScript by setting its `style.opacity` property.
5. You can animate (for instance, a fade is an animation) any CSS setting (such as opacity) in JavaScript by setting its `style.transition` property. For instance, in JavaScript, setting an object's transition style property to "opacity 1s" tells the browser to transition the opacity to its next setting across one second.

Test

1. Verify the page changes the larger image when you click on a thumbnail. Hover the mouse over the large image to verify that the caption fades into visibility, and that it fades to invisibility when the mouse moves out of the image (see Figure 9.20).

PROJECT 3: CRM Admin

DIFFICULTY LEVEL: Advanced

HANDS-ON EXERCISES
Project 9.3

Overview

Write a helper script that uses recursion and which could potentially be used on any web page to visually identify the tag name of all elements on a page. Recursion as a programming topic was not covered in this chapter, so this project is only suitable for a programmer already familiar with the technique.

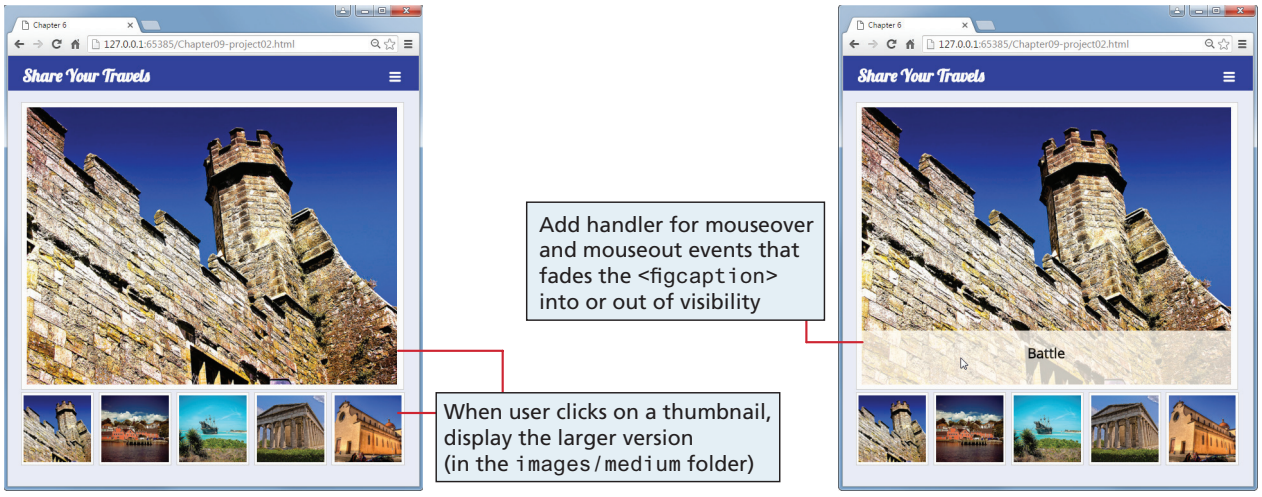

FIGURE 9.20 Finished Project 2

Instructions

1. Examine **chapter09-project03.html**. You will add event handlers to the two buttons at the bottom of the page. You will not need to make any changes to the supplied markup or CSS. All of your event handlers must execute only after the page has loaded.
2. The handler for the Highlight Nodes button should navigate every element in the DOM, and for each element within the body, determine whether it is an element node (`nodeType == 1`) element.
3. If it is an element node, add a new child node to it. This child node should be `<span>` element with the `class=hoverNode`. Its `innerText` should be equal to its parent's tag name.
4. Add an event listener for this child node so that when the user clicks on the new span, an alert popup displays the details the following information about its parent node: id, tag name, class name, and inner HTML.
5. The Highlight Nodes button should hide when the user clicks on it. The Hide Highlight button should then be displayed. When the page is first displayed, the Hide Highlight button should be hidden.
6. When the user clicks the Hide Highlight button, all the `<span>` elements with `class=hoverNode` should be removed. The Hide Highlight button should then be hidden, and the Highlight Nodes button should be displayed.

Test

1. Test by clicking the Highlight Nodes button and the Hide Highlight button (see Figure 9.21).