6. What is a `Promise` in JavaScript? What problem do they solve?
7. What is the purpose of `async ... await` in JavaScript? What problem does it address?
8. What is cross-origin resource sharing? What relevance does it have for JavaScript applications using asynchronous requests?
9. How are web APIs different from browser APIs and external APIs?

10.6.3 Hands-On Practice

PROJECT 1: Text Viewer

DIFFICULTY LEVEL: Intermediate

Overview

This project uses fetch to retrieve data and applies the DOM techniques from the previous chapter. It also uses one of the array functions from the beginning of the chapter. Figure 10.27 indicates what the final result should look like in the browser.

Instructions

1. You have been provided with the necessary styling and markup already. Examine **ch10-proj01.html** in the editor of your choice. Notice the sample markup for the color scheme list items. This will be eventually commented out

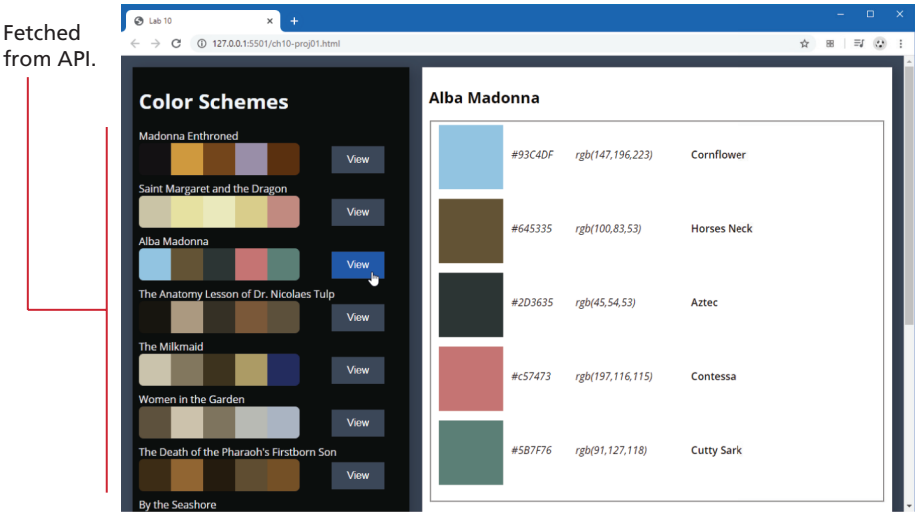

FIGURE 10.27 Completed Project 1

and replaced with JavaScript code that programmatically generates this markup. There is also a loading animation that will need to be displayed/hidden.

2. Examine **ch10-proj01.js** in the editor of your choice. In it, you will see the URL for the external API that will provide the color scheme data. Examine this URL in the browser in order to see the structure of the data.
3. Fetch this scheme data from the API and display it within the `<article>` element. As you can see from the sample supplied markup, this will require creating `<h3>`, `<section>`, `<div>`, and `<button>` elements.
4. Display the loading animation before the fetch and then hide it after the data is retrieved.
5. Set up a single click event handler for *all* the View buttons. This will require using event delegation. When the user clicks a view button, display the scheme details in the `<aside>` element. As you can see from the sample supplied markup, this will require creating `<div>` elements within the supplied `<fieldset>`. You will also have to change the `<h2>` content to the clicked scheme name. Hint: use the `find()` method to retrieve the correct scheme object from the `data-id` property of the clicked button. Also, remember to clear out the previous content of the `<fieldset>` by setting its `innerHTML` to `""`.

Guidance and Testing

1. Break this problem down into smaller steps. First verify the fetch works, perhaps with a simple `console.log` statement. Then write a function that generates the markup for a single color scheme in the `<article>` and test to make sure it works. This will require a loop, so try using `forEach()` instead of a `for` loop.
2. Then add in support for the loading animation.
3. Before generating the scheme details, add in the event handler using event delegation and verify (again using `console.log`) if you are able to retrieve the correct scheme object using `find()`.
4. Finally, write a function that generates the scheme details. This will require a loop, so try using `forEach()` instead of a `for` loop.

PROJECT 2: Text Viewer

DIFFICULTY LEVEL: Intermediate

Overview

This project focuses on the first two sections of the chapter (array functions and prototypes/classes/modules). It also uses fetch to retrieve data. Figure 10.28 indicates what the final result should look like in the browser.