

W02 Assignment: Picture Album

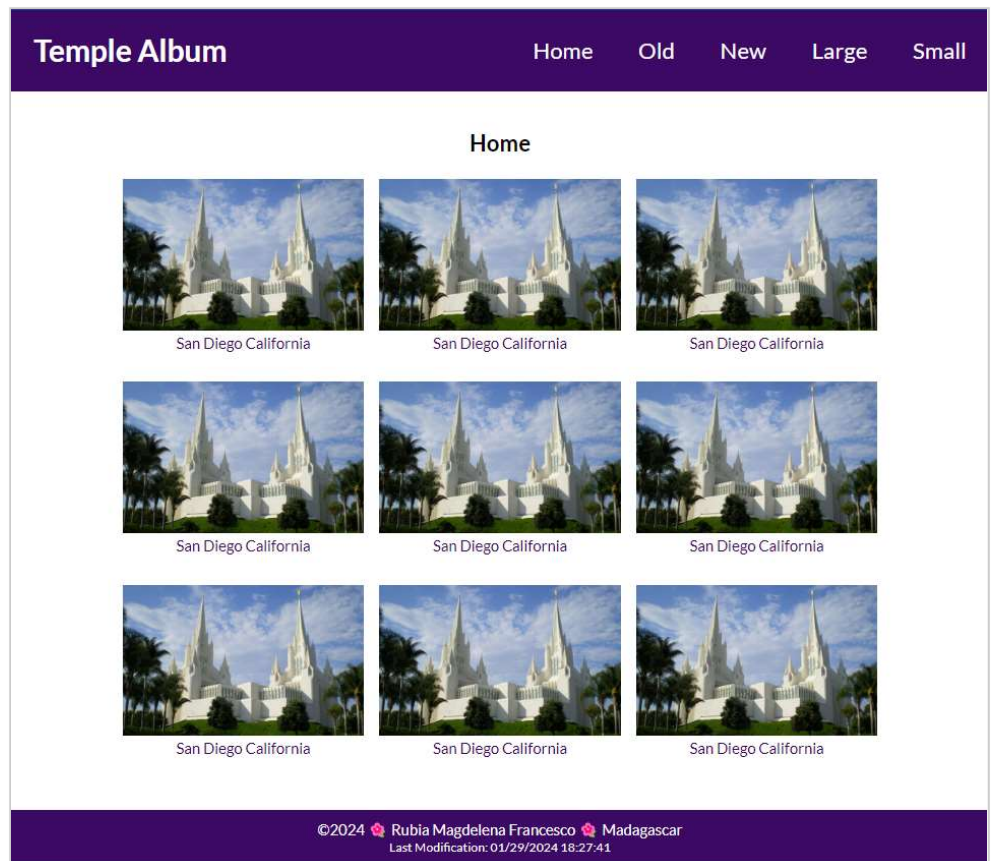
Overview

This assignment applies the concepts presented in the learning activities to a picture album page that is responsive in small and larger views. This page will utilize a 'hamburger' menu and respond to user requests.

► Associated Course Learning Outcomes

Task

Design, develop, test, and then deploy a temple album page using your own selections of temple pictures.



Screenshot Example in Larger View



Screenshot Example in Mobile View

► Demonstration of Responsive Behavior

Instructions

Step 1: File and Folder Setup

1. In VS Code, open the **wdd131** local repository folder if it does not open automatically because it was your last working directory.
2. Create a new file named "**temples.html**".
3. Add two CSS files named "**temples.css**" and "**temples-large.css**" to the "styles" directory.
4. Add a JavaScript file named "**temples.js**" to the "scripts" directory.

Step 2: HTML

1. In the **templates.html** document, include the standard HTML document and **<head>** elements.

Refer to the [development standards](#) if you need a review.

2. Add links to the **CSS** files in the proper order to support mobile first dev.
3. Add a deferred **<script>** reference to the **JavaScript** file.
4. In the **<body>**, create a basic layout using a **header**, a **main**, and a **footer** as main line elements.
5. The **<header>** element contains:
 - ✓ a heading that matches up with the title, and
 - ✓ a **<nav>** menu with the following text links:
 - Home
 - Old
 - New
 - Large
 - Small
6. The **<main>** element contains the following:
 - ✓ a heading element with content matches the current menu selection,
 - ✓ at least (9) **<figure>** elements with temple images and captions. Use the built in figure captions elements and name the temple.

It is **OK** to have a single placeholder temple image at this point in the course.

7. The **<footer>** contains the same dynamic and static information found on your home page from last week.

©2024 🌸 Rubia Magdalena Francesco 🌸 Madagascar
Last Modification: 01/29/2024 19:11:06

Screenshot Example of Common Footer

Step 3: CSS

Use the external **templates.css** and **templates-large.css** files to layout and style the page as shown in the example screenshots. Your design must support a responsive view in mobile and larger views.

The **templates.css** file is used for the mobile view and the **templates-large.css** file is used for the larger view.

Most of your CSS should be located in the **templates.css** file in mobile first design.

1. Use your own color schema.

Please note: You are responsible to practice good design principles of alignment, color contrast, proximity, repetition, and consistent white space.

2. Use your own typography choice by using the [Google Fonts API](#) service to select one or two fonts to use on the page.

Video Demonstration:  [Using Google Fonts](#) [1:56 minutes]

3. Use **CSS Flex** on the navigation **nav**.

Video Demonstration:  [CSS Flex Navigation Menu](#) [7:58 minutes]

CodePen Source Code:  [Navigation Menu using CSS Flex](#)

4. The navigation must employ a **hover** affect. See the **CodePen** above for an example.

► Check Your Understanding

5. The **main** element has a limited width and is centered on the screen horizontally.

► Check Your Understanding

6. Layout the main column **figure** elements using **CSS Grid**. In the mobile view, there should only be one (1) column.

The application of CSS Grid to support a responsive view is up to you.

Some of the options could be:

- use **grid-template-column** specific **fr** changes in the media query
- use **grid-template-column** property with a **repeat** function and **auto-fit** and **minmax** function.

CodePen Source Code:  [Grid Column Layouts and Image Effects](#)

Look at the CSS for the .container class.

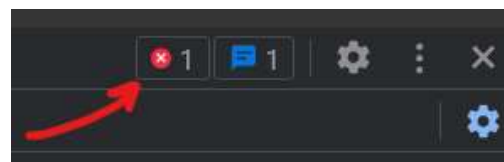
Step 4: JavaScript

Remember that all of your JavaScript references in this class should be to external files.

1. Support the **footer** copyright year and date last modified output through a valid JavaScript reference. You already have one or could use the **temples.js** file.
2. Apply a responsive hamburger effect to your existing navigation menu using JavaScript.
 - ✓ The hamburger button should only show in the mobile view.
 - ✓ Clicking the hamburger button toggles the navigation menu items from viewable to not viewable.
 - ✓ Use a symbol, such as an 'X' to close the hamburger menu.

Step 5: Testing

1. Continuously check your work by rendering the page locally using Live Server in VS Code.
2. Use the browser's DevTools to check for JavaScript runtime errors in the console or click the red, error icon in the upper right corner of DevTools.




"DevTools" is an abbreviation for "Developer Tools." It refers to a set of tools or utilities provided by web browsers to help developers debug, profile, and analyze web pages during the development process. The tools are typically accessed by pressing the F12 function key or selecting the menu option for the browser's developer tools.

3. Use DevTools [CSS Overview](#) to check your color contrast.
4. Generate the DevTools **Lighthouse** report and run diagnostics for **Accessibility**, **Best Practices**, and **SEO** in **both the mobile and Desktop** views.

It is best to test your page in a **Private** or **Incognito** browser window.

5. Hard reload the page using **Empty Cache and Hard Reload** in **DevTools** with the **Network** tab open to view the total transferred bytes at the bottom of the tab. Check that the page is 500 kB or less. The usually fixes at this point are:
 - Make sure that all images are optimized.
 - Third party libraries are not allowed because these libraries typically add a lot of unused code to the page and the learning outcome is to understand the basic concepts in web frontend development.

Audit and Submission

1. Commit your local repository and push or upload your work to your GitHub Pages enabled **wdd131** repository on GitHub.
2. Use this  [audit tool](#) to check some basic HTML and CSS standards and requirements.
3. **Share** your work by posting your URL in the course's **Microsoft Teams weekly forum** and comment on your peers' work by providing constructive feedback.
4. Return to Canvas and submit your the correct GitHub Pages enabled URL.

`https://your-github-username.github.io/wdd131/temple.html`

Copyright © Brigham Young University-Idaho | All rights reserved