

CS 110

Lab 3: Create an Image Gallery

Overview

In this lab, you will create an image gallery webpage using HTML and CSS. You'll learn about modern CSS styling techniques, including flexbox layouts, hover effects, and transitions. The gallery will showcase different categories of images with descriptions and interactive elements.

Instructions

1. Download the Starter Files

Download the following files from Canvas and save them to your lab3 folder:

- gallery.html
- gallery-styles.css
- images folder containing: Nature.png, Modern_architecture.png, Abstract_art.png, Urban_cityscape.png, Wildlife_photography.png, and Minimalist_composition.png

2. Complete the HTML Structure

Open gallery.html in VS Code. The basic structure is provided, but you need to add the images and hyperlinks for each section.

Important Note about Image Paths:

In `src="images/Nature.png"`, we include `"images/"` before the filename because:

- The images are stored in a folder named `"images"`
- The `"/"` indicates we're going into that folder to find the image
- This is called a relative file path
- it shows the path from our HTML file to the image file
- Learn more about file paths here:
https://www.w3schools.com/Html/html_filepaths.asp

Here's an example for the Nature section:

```
<section>  
  <h2>Nature</h2>
```

```
<div class="image-container">
  <a href="https://www.google.com/search?q=nature" target="_blank">
    
  </a>
</div>
<p>Experience the beauty of untouched landscapes and natural wonders</p>
</section>
```

After adding the necessary anchor and image HTML elements to the first section:

1. Save your file
2. Preview in Chrome using Live Server
3. Check that:
 - The Nature image appears correctly
 - The link opens a Google search for “nature” in a new tab when clicked
 - The heading and description text are visible

For each section’s search link, you’ll need to modify the ‘q=’ parameter in the URL.

For example:

- For a single word: <https://www.google.com/search?q=nature>
- For multiple words: <https://www.google.com/search?q=abstract+art> (use + between words)

Now complete each remaining section one at a time. After adding each section, save and test before moving to the next:

Remember: Keep Chrome open with Live Server running. Each change should be immediately visible after saving.

1. Architecture section:
 - Add href=“<https://www.google.com/search?q=architecture>”
 - Use Modern_architecture.png
 - Save and preview
 - Test that both image and link work
2. Abstract Art section:
 - Add href=“<https://www.google.com/search?q=abstract+art>”
 - Use Abstract_art.png
 - Save and preview
 - Test that both image and link work
3. Urban Life section:
 - Add href=“<https://www.google.com/search?q=urban+life>”

- Use Urban_cityscape.png
- Save and preview
- Test that both image and link work
- 4. Wildlife section:
 - Add href="<https://www.google.com/search?q=wildlife>"
 - Use Wildlife_photography.png
 - Save and preview
 - Test that both image and link work
- 5. Minimalist section:
 - Add href="<https://www.google.com/search?q=minimalist>"
 - Use Minimalist_composition.png
 - Save and preview
 - Test that both image and link work

After completing all sections:

- Verify all six sections are displayed properly
- Test all links open in new tabs
- Confirm all images are showing correctly
- Make sure there are no broken images or links

Note: At this point, you'll notice all the sections are displaying next to each other in a single row, making the page stretch horizontally. This is the default behavior of the flexbox layout we're using. Don't worry - in the next part, we'll fix this by adding CSS properties to make the sections wrap into multiple rows and create a proper grid layout.

3. Understanding and Modifying CSS Styles

Part 1: Universal and Base Styles

1. Examine the universal selector `* { }`:
 - This selector applies styles to ALL elements
 - Understand the purpose of margin/padding reset
 - Note how box-sizing affects layout calculations
2. Explore body and overlay styles:
 - a. Modify body padding:
 - Find body padding: 40px 20px
 - Try these values:

- Change to padding: 20px 10px; /* Smaller spacing */
 - Save and preview in Chrome to see tighter spacing
 - Change to padding: 60px 40px; /* Larger spacing */
 - Save and preview to see more generous spacing
 - Change to padding: 100px 50px; /* Very large spacing */
 - Save and preview to see extreme spacing
 - Change back to padding: 40px 20px;
 - Save and preview to confirm return to original spacing
- b. Experiment with overlay opacity:
- Find background: `rgba(255, 255, 255, 0.9)` in `body::before`
 - Try these values:
 - Change to background: `rgba(255, 255, 255, 0.5)`; /* More transparent */
 - Save and preview to see more of the background showing through
 - Change to background: `rgba(255, 255, 255, 0.7)`; /* Somewhat transparent */
 - Save and preview to see moderate transparency
 - Change to background: `rgba(255, 255, 255, 1)`; /* Completely opaque */
 - Save and preview to see solid white overlay
 - Change back to background: `rgba(255, 255, 255, 0.9)`;
 - Save and preview to confirm return to original opacity

Remember: Keep Chrome open with Live Server running. Each change should be immediately visible after saving.

Part 2: Header Styling

1. Modify the header appearance (after each change, save your file and check the result in Chrome using Live Server):
 - a. Change the header background color:
 - In the header {...} section, find `background: rgba(215, 38, 38, 0.95)`
 - Change it to `background: rgba(255, 255, 255, 0.95)`
 - Save and preview in Chrome to see how the header turns white
 - b. Add a shadow effect:
 - Find the commented line `/box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);/`
 - Remove the `/*` and `*/` to uncomment it
 - Save and preview to see the subtle shadow appear
 - c. Experiment with border-radius:

- Find the border-radius: 15px line
- Try different values: 0px, 30px, 50px
- Save and preview after each change to see how the header corners change

Remember: Always keep Chrome open with Live Server running while making these changes. Each time you save your CSS file, Chrome will automatically refresh to show your changes.

Part 3: Container and Section Layout

Important Note about CSS Selectors: In `.container {...}`, we use a dot (.) before "container" because:

- The dot indicates we're selecting an HTML element with a class attribute
- In this case, we're targeting `<div class="container">`
- Without the dot, CSS would look for a `<container>` element (which doesn't exist)
- This is called a class selector in CSS

1. Explore Flexbox Layout (preview each change in Chrome using Live Server):
 - a. Enable flex wrapping:
 - Find the `.container {...}` section
 - Locate the commented line `/flex-wrap: wrap; /`
 - Remove the `/*` and `*/` to uncomment it
 - Save and preview to see how sections now wrap to new rows when the browser window is narrowed
 - b. Adjust spacing between items:
 - Find the `gap: 20px` line
 - Try different values: 10px, 30px, 50px
 - Save and preview after each change to see how the spacing between sections changes
 - c. Set section widths using flex:
 - Find the `section {...}` block
 - Locate the commented line `/flex: 1 1 calc(50% - 20px); /`
 - Remove the `/*` and `*/` to uncomment it
 - Save and preview to see how sections now take up exactly half the container width (minus the gap)
 - This property does three things:
 - `flex-grow: 1` (allows sections to grow)
 - `flex-shrink: 1` (allows sections to shrink)
 - `flex-basis: calc(50% - 20px)` (sets base width to 50% minus the gap)

2. Add Section Hover Effects:

a. Basic hover effect:

- Find the comment `/* Hover effect for sections */`
- Add this code below it:

```
section:hover {  
  transform: translateY(-5px);  
  box-shadow: 0 6px 20px rgba(0, 0, 0, 0.15);  
}
```

- Save and preview by hovering over different sections

b. Customize the hover effect (try each change and preview):

○ Change translateY value:

- Try -3px (smaller movement)
- Try -8px (larger movement)
- Return to -5px

c. Modify shadow intensity:

○ In the box-shadow property:

- Try `rgba(0, 0, 0, 0.1)` for a lighter shadow
- Try `rgba(0, 0, 0, 0.3)` for a darker shadow
- Return to `rgba(0, 0, 0, 0.15)`

Part 4: Image Effects

1. Add image hover zoom effect where indicated:

```
section:hover .image-container img {  
  transform: scale(1.05);      /* Slight zoom on hover */  
}
```

2. Experiment with different values (save and preview each change):

a. Try different scale values:

- `transform: scale(1.1);` */* More noticeable zoom */*
- `transform: scale(1.2);` */* Large zoom */*
- `transform: scale(1.02);` */* Subtle zoom */* Return to `scale(1.05)` for the best effect

b. Adjust transition timing (in the `.image-container img {...}` block):

○ Change transition: `transform 0.3s ease` to:

- `transition: transform 0.1s ease;` */* Faster */*
- `transition: transform 0.5s ease;` */* Slower */*
- `transition: transform 0.3s ease-in;` */* Different timing function */*
- `transition: transform 0.3s ease-out;` */* Different timing function */*
Return to `transform 0.3s ease` for the smoothest effect

Save and preview after each change. Pay attention to:

- How far the image zooms when hovering
- How smooth the zoom animation feels
- Whether the effect feels natural or too dramatic

5. Testing and Validation

1. Preview your work:
 - Use VS Code's Live Server to view changes in real-time
 - Test the hover effects and transitions
 - Verify all images and links work correctly
2. Validate your code:
 - Use W3C HTML Validator for gallery.html: https://validator.w3.org/#validate_by_upload
 - Use W3C CSS Validator for gallery-styles.css: https://jigsaw.w3.org/css-validator/#validate_by_upload
 - Fix any validation errors

Submission

1. Verify all images display correctly
2. Test all hover effects and transitions
3. Validate both HTML and CSS files
4. Zip the following files:
 - gallery.html
 - gallery-styles.css
 - images folder
5. Submit the zip file to Canvas