Katie Sun
PUI Lab C
Final Project Write-up

Responsive screen sizes

- **Laptop: 1440px**
- **Mobile: 425px**

Part 1: Website Overview

The **purpose** of this project is to a showcase an illustration challenge I completed that involved painting one random Pantone paint swatch a day, completed over the course of 100 days. Each painting was inspired either by the original shade of the swatch, or the name of the color itself.

This project will be featured on the 'play' section of my professional portfolio, and will offer an interactive experience wherein a visitor can click on a color **swatch to see the painting that specific shade inspired, the hex code, the name of the color, and the day of the challenge on which it was painted** (i.e. PANTONE #B7BF96 Sage Green [Original Shade], Day 9, Painting). Its primary motivation is to show **prospective employers** that I like to challenge myself to experiment creatively outside of my professional and academic career, and in a way that is structurally befitting the specific nature of the project. Most existing 'image gallery' templates only allow for one facet of information (the image itself) and lack the ability to capture the inspiration behind the work in an engaging way, and coding it makes a more multifaceted visualization possible.

To make the information more **interesting and engaging**, I've incorporated several interactions and animations that make the exploration experience more exciting and encourage a user to explore more cards. Additionally, this project will eventually be expanded to all 100 swatches and paintings, and thus I've included 'sort by day,' 'sort by color,' and 'filter cards by color' functionalities to help parse the information more. This also illustrates not only the progression of the *quality* of the work over time, but also uses the grid to create a kind of color gradient that adds to visual appeal.

Part 2: Interactions

- **Hover** over a color swatch card to enlarge the swatch
- **Click** a color swatch to select that color swatch and reveal the corresponding painting, hex code, shade name, and day of the challenge
- **Click** the button on the lower right-hand corner to enable and disable **dark mode**
- **Click** the second button on the navigation bar to **sort the color swatches chronologically** (by date of the challenge, ascending)
- **Click** the third button on the navigation bar **to expand and collapse the menu of options** to filter by color family
- **Click** any of the colored buttons on the navigation bar to **filter color swatches by color family**
- **Click** the first button on the navigation bar to **reset** the color grids and the Pantone card

Part 3: External tools

- **Darkmode.js** (dark mode JavaScript library)
    - *Why you chose to use it:*
      Darkmode.js allows me to easily incorporate both a black and white background for color contrast that can be toggled by the user, rather than having to find a single solution for a largely subjective experience (e.g. a single shade of grey that didn't distract from the color cards or wash out the saturation), or having to manually code background and element color changes on a button coded with JavaScript and CSS.
    - *How you used it:*
      I imported the JavaScript library with JSDelivr CDN and left the button at the bottom right corner, forcing it to the front with a z-index to have it present on all screen sizes. It exists as a toggle option, but I've elected to keep the default as light mode.
    - *What does it add?*
      User feedback during earlier testing revealed a split between individuals who felt that a darker background provided better contrast for the Pantone card text section, and those who felt that a white background highlighted the paintings themselves better. Incorporating the darkmode.js library allows both user segments to choose the background that augments their personal inclinations best. Notably, the library actually automatically sets the default mode based on the user's OS preferred theme for a fully personalized experience and even stores the user's choice if the website is ever revisited.

- **Animate.css** (CSS animation library)
    - *Why you chose to use it:*
      The CSS animation library provided a skeleton structure that allowed me to easily add animations to my project, without having to code them with keyframes or in pure CSS transformations.
    - *How you used it:*
      CSS animations are present when the reset button is clicked to fade the Pantone card in, a way to subtly show a shift in mode from individual paintings back to a 'home' screen. A bounce animation is also used when the filter-by-color-family buttons drop in and out when the filter button is clicked. A horizontal flip animation is used on the Pantone card to mimic the change to a new shade/painting and to draw attention to that section of the page. Finally, a vertical flip animation is used on the 'day of the challenge' counter to draw attention to that change as well.
    - *What does it add?*
      CSS animations allow me to utilize motion to draw attention to certain parts of the interface and thus to dynamically highlight information based on their contextual relevance. They also provide both feedback and forward for the user to encourage certain actions (e.g. this button is clickable!), as well as to indicate more clearly the effects of an interaction (e.g. a transition to a new state). When used subtly, they can augment the user's experience both in terms of satisfaction and effectiveness.

- **CSS animation**
    - *Why you chose to use it:*

We've primarily used CSS to style static objects on the DOM, and I wanted to learn how to use keyframes to create simple animations. CSS animations were more appropriate for my needs as well—the simple gradient animation was relatively small, self-contained, on a loop, and did not require a start/stop trigger. This would also allow more control over individual animation keyframes.

- *How you used it:*
  I created a gradient background for the 'home screen' Pantone card, before the user has selected a specific swatch. The gradient plays in the background on infinite, automatic loop until a card swatch is clicked; clicking the reset button brings it back.
- *What does it add?*
  Rather than simply leaving a blank white space or starting with the Day 1 painting, the inclusion of a gradient background adds not only a 'base' state, but also serves as almost a metaphor/feedforward for the project itself (i.e. in total, 100 colors featured across the rainbow, etc.).

Part 4: Iterations

For this final iteration, I made a few modifications for the UI and the available interactions to add flexibility to the user's experience. Feedback from FP2 revealed a split between users that preferred a black or white background contrast, and thus the discovery of Darkmode.js provided an unexpected solution. In earlier designs, I had included a navigation bar that mimicked my portfolio (on which this project would occupy a section, along with the default project header and description), but user testing revealed that people expected those buttons to be functional and were distracted from the content of the project itself. As such, I removed those header elements. At the final project check-in, feedback also revolved around whether or not the color swatches should be sorted by date (practical) or by shade (visually impactful), and thus I included an additional sort functionality to give weight to both perspectives.

Part 5: Challenges

I personally found the inclusion of the Darkmode.js library to be particularly difficult, given that the code did not adapt well to changes. I attempted to change the location, but the options modification provided in the documentation kept throwing an error once outside the developer console. I also struggled initially with figuring out how to set up my classes and objects in an array in a way that would allow me to dynamically update the DOM, rather than hard-coding separate swatch objects (theoretically feasible in the 20-color slice for this project, but difficult if I were to extend this project after this course to all 100).

Accessibility Checker (WAVE)