

Morgan Rosen

Sep 23rd

It was eye-opening to use Flexbox and CSS Grid for this project. Flexbox was simple to work with at first, I could easily line up images in a row or column but when I tried to organize the gallery into individual sections like Architecture and Nature, it got disorganized. I had to add in extra wrappers and make a few adjustments just to get elements to align. CSS Grid did, however, make it incredibly easier to place pictures exactly where I wanted them, especially the bottom one with the featured photo. It was nice being able to define rows and columns and knowing that everything would stay organized. The Grid layout handled my JavaScript filtering wonderfully. When I clicked on a category button, the items concealed or appeared without holes, which did not always happen with Flexbox. Writing the buttons to the layout using JavaScript was experimentation. I had to figure out how to add data-category and toggle classes so that the layout wasn't collapsing, but when it was completed, it felt snug and stable. The filtering capability made the gallery interactive and easy to use. Clicking a button instantly shows only the pictures that I want to view, so it's clear that something happened. Something I discovered is that there is no animation or visual effect for the active filter, adding some more highlight on the clicked button would make it more refined. For accessibility, keyboard users can tab through the buttons, but it's not indicated which one is in focus. Screen reader users will also struggle without descriptive text. These are things I can fix. With Grid, it was all easier because the images stay in neat rows and columns even when some of them are hidden away. If I

were using Flexbox, I would have had to play around more with spacing and wrapping.

Overall, choosing Grid not only made it possible to achieve a clean layout but also simpler to add interactivity that is well-timed and natural-feeling.