

Technical Analysis

Switching from Flexbox to CSS Grid for this layout felt way more intuitive. With Grid, I can sketch the page in code using grid-template-areas like “header header”, “nav sidebar”, and “main sidebar”, and the browser does the heavy lifting. Making the featured photo larger is trivial with grid-column: span 2;. In Flexbox, getting one card to grow while the rest flow neatly usually needs extra wrappers, order tweaks, or width math, and it often feels like I am fighting the layout.

Responsiveness is also cleaner with Grid. `repeat(auto-fit, minmax(250px, 1fr))` lets the gallery expand and collapse as the viewport changes without a bunch of breakpoint micromanagement. When filtering hides cards with `display: none`, Grid reflows items into the open slots naturally. Flexbox can reflow too, but since it is one dimensional, it is easier to end up with ragged edges unless wrap behavior is tuned carefully.

The main JavaScript wiring challenge was timing and selectors. If the script runs before the DOM exists, `querySelectorAll` returns nothing and the buttons do nothing. Also, if button text does not match data-category, filtering silently fails.

Design and User Experience Considerations

The filter interaction works, but the feedback is minimal. Right now, clicking a button only changes what is visible. I would add a clear active state on the selected filter and a small count like “Showing 4 items” so users know the action succeeded. A quick fade or scale

animation on cards appearing and disappearing would make the change feel smoother without stealing attention from the photos.

Accessibility wise, two easy wins stand out. First, treat the buttons as toggles for assistive tech by updating aria-pressed on the active filter and set type="button" to avoid accidental form submits. Second, add a polite live region that announces the visible count after each filter. For keyboard users, keep strong focus outlines on buttons and make sure the filter row is early in the tab order. Longer term, expose categories in text and tie the filter group to the grid with aria-controls.

Choosing Grid helps interactivity because the structure is explicit. JavaScript only toggles visibility while Grid handles the reflow in a predictable way.