(or a media query) to hide/show the thumbnails. A small JavaScript snippet can add a class like `grid-view` to the `<body>` when the user clicks “Grid”, switching the layout CSS. This approach lets the user toggle without a page reload and can remember the preference (via localStorage) if desired. - **Separate Pages:** As an ultra-light solution (no JS), you can create two versions of the index: e.g. `index.html` (default list view) and `index_images.html` (grid view). Chloe Scheffe does this – her navigation has “.list” and “.jpgs” links that swap between text index and an image gallery index [15] [16]. In your static generator, you could generate both pages. This method ensures each view is just static HTML (fast), though the user experiences a page refresh when toggling.

On small screens, ensure the grid view remains user-friendly: you might show a two-column grid of thumbnails or even a single column of larger images (depending on what looks best) using CSS media queries. The list view on mobile would simply stack the project entries vertically (which is inherently mobile-friendly).

- **Responsive, Mobile-First Design:** Design **mobile-first**, meaning structure your HTML/CSS for a small-screen experience first, then enhance for larger screens. This ensures the site looks and works great on phones (likely the first visit for many). Use modern CSS layout (flexbox or grid) to create a responsive gallery and list. Navigation should collapse elegantly – e.g., use a simple hamburger menu or an always-visible minimal top bar with just your name/logo and a menu icon. Given the minimal style, even a centered title with a dropdown menu could suffice.

- **Navigation & Information Architecture:** Keep navigation simple to maintain focus on the work. You might have just a few links: **Work** (the main portfolio list/grid), **About** (with a bio and maybe client list or resume highlights), and **Contact**. If you have categories of projects (branding, web design, product design), you could implement filters or separate sections, but given a minimal approach, a single list with categories labeled on each project might be enough. Consider a “light” and quick navigation – for example, a fixed header with your name that, when clicked, always brings back to the index, plus a toggle for list/grid, and maybe an info icon for About. Avoid heavy multi-level menus.

- **Project Pages & Visuals:** When showcasing individual projects, maintain the minimal style: use ample white space and typography to narrate the project. High-quality images of your work should be present, but optimize their display:

- Use *lazy-loading* for project images so that long case study pages don’t choke mobile data [17].
- Ensure images have responsive sizes (e.g., via the `srcset` attribute or `<picture>` element) so mobile devices load smaller versions and retina screens get crisp versions.
- For a *high-end feel*, you can add subtle interactive touches: for example, a hover effect on project thumbnail (e.g., slight zoom or a color overlay reveal), or smooth CSS transitions when switching views. Keep these subtle and CSS-based for performance.

- You might also incorporate a theme toggle (light/dark mode) if it suits your branding, but only if you have time – it’s a nice-to-have for personal sites these days.

- **Draw Inspiration from Top Portfolios:** Continue to draw inspiration from minimal portfolio examples. Chloe Scheffe’s site shows that even without heavy graphics, a list of work can feel compelling and “archival” in a good way. Other examples: *Mike Pierce’s* portfolio (text-driven with