

trade-offs if you rely on their auto-formatting. In a static site you control the mobile experience directly and can ensure it's as light as possible.

- **Flexibility & Design Freedom:** Building it yourself gives you **total control** over the look and features. You can implement the list/grid toggle, custom layouts, interactive touches, or any quirky design ideas with no platform constraints. Cargo does allow custom HTML/CSS and some scripting, but within the limits of their system. If something isn't supported (or if their template doesn't allow a certain layout), you might have to hack around it. For example, if you wanted a truly custom navigation or dynamic filter, it's much easier when you own the code. Also, Cargo templates can be a bit rigid – as one user noted, *templates have limitations; the flexibility of Cargo is nice, but templates "kinda suck" in terms of complete control* ²⁶. In contrast, a static site is essentially a blank canvas.
- **Ease of Use & Maintenance:** Cargo's advantage is its **convenient editor** – you can log in and drag-and-drop content or fill forms to update your site, no coding needed. If you dislike touching code, this is a plus. It's also a hosted solution, meaning no builds or deployments to worry about; you just use their UI. With a static generator, you'll maintain the site via code. However, as discussed, your update frequency is low, and the content structure will be straightforward (projects and pages). Once the static site is set up, adding a new project might be as simple as copying a Markdown file and changing text. The build and deploy can be automated. Additionally, integrating a light CMS (Netlify CMS) can nearly replicate the ease of logging into an interface to add content, mitigating this difference.
- **Workflow Integration:** Since you are already a developer/designer, having your portfolio in git might actually streamline your workflow – you can version control changes, use issues/pull requests for major revamps, etc. Cargo doesn't provide that level of versioning or local development; it's all on their platform. Many professionals prefer having their portfolio in a repository, because it's easier to experiment in a branch, roll back if needed, and collaborate (e.g., if you had a developer friend help, etc.). Also, static site code can be open-sourced on your GitHub if you want to show it as part of your work.
- **Cost:** Cargo is a paid service for custom domains. It's **free to try**, but to use your kev.studio domain and unlock full features, it's about **\$13–14 per month** ²⁷ (roughly \$150+ per year). In contrast, GitHub Pages is free for unlimited sites, and Netlify or similar hosts are also free at the scale of a personal site. By going static, you essentially eliminate ongoing fees (aside from your domain registration cost). Over years, this is a non-trivial saving. While \$13/mo isn't huge, getting top performance and full control *for free* is quite attractive.
- **Support and Community:** With Cargo, you have the Cargo support team and community if something goes wrong or if you need help using their system ²⁸. With a custom static site, you won't have dedicated support, but static sites generally "just work" once built. There is a vast community for whatever SSG you choose (Jekyll, Eleventy, etc.), and plenty of guides on platforms like Stack Overflow if you hit a snag. In practice, the issues you'll face are likely simpler (e.g., "how do I add this script" or "why isn't this layout working in CSS"), which are general web development questions with lots of documentation.

Bottom Line: For a portfolio that is updated infrequently and where you desire maximum performance, customization, and a bespoke feel, a custom static site is likely the superior choice. It will give you a