

How I Built My Site With Claude

A step-by-step playbook for media personalities, creators, and multi-brand humans who want to leave Wix behind.

BY KATOYA RAQUELL PALMER

Founder, ToyBox · Author, *Five Windows* · Writer at katoyapalmer.com

I had three brands, two podcasts, a consulting business, an event series, and one increasingly expensive Wix subscription. This is the exact path I took to move all of it to my own GitHub-hosted, fully analytics-instrumented, search-indexed, mobile-polished, SSL-secured stack — using Claude as my co-pilot. The whole thing now costs me almost nothing to host. Here's the playbook.

Inside this playbook

The story	Why I left Wix and what I built instead
Phase 1	Plan your brand architecture
Phase 2	Set up Claude as your co-pilot
Phase 3	Build the bones — GitHub, domains, DNS
Phase 4	Design your pages with Claude
Phase 5	Deploy to GitHub Pages
Phase 6	SEO and search engine discovery
Phase 7	Install your analytics stack
Phase 8	Mobile, accessibility, security
Phase 9	Legal pages and consent
Phase 10	Email and audience
Phase 11	Real-time visitor pings
Closing	What I'd do differently, what's next

THE STORY

Why I left Wix and what I built instead

I'm a writer, a systems strategist, and the kind of person who runs more than one thing at once. Over the years that turned into: a personal site, a podcast (*In Real Time with Katoya Palmer*), a Substack, three creative sub-brands (Mindfulness in the Middle, SZND, Lipstick on My Fork), a consulting business (ToyBox), and an event series (Golden Hour Unboxed).

All of that lived on Wix. Wix worked — until it didn't. Every new brand meant another page, another premium template, another monthly bill. Customization was clumsy. SEO felt locked-in. Real-time analytics required a paid add-on. Migrating anything was a fight.

I moved everything to GitHub Pages, with Claude as my co-pilot. Hosting is now functionally free. I own the code. I can change anything I want. And I have a professional analytics + SEO + email stack that costs me almost nothing per month.

What you'll have when you're done

End state: 3 live sites (*katoyapalmer.com*, *toyboxconsulting.net*, *goldenhourunboxed.com*), each with its own custom domain, HTTPS, sitemap, analytics, search-engine indexing, mobile polish, legal pages, and content security policy. Total monthly hosting cost: \$0.

This playbook walks you through every phase — in the order I did it, with the why behind each step.

PHASE 1

Plan your brand architecture

Before you build anything, decide what you're actually building. A media personality usually isn't one brand — they're a constellation. You write, you record, you cook, you consult. Each of those audiences wants a slightly different doorway.

The hub-and-spoke model

My architecture has one **hub** and many **spokes**:

- **Hub** — **katoyapalmer.com** — the personal home. Anyone Googling my name lands here. It links out to everything else.
- **Spokes** — Each sub-brand gets its own page on the same domain (*lipstick.html*, *sznd.html*, *mindfulness.html*). One repo, multiple pages, distinct visual identities.
- **Business site** — **toyboxconsulting.net** — a completely separate domain and repo for client-facing work. Different audience, different voice.
- **Event sites** — **goldenhourunboxed.com** — standalone landing pages for time-bound events. They go live, they retire, they don't clutter the main site.

Decide before you write a line of code

Write down, in plain language: what brands you have, who they're for, and whether each one should live on the hub or on its own domain. This will save you a week of rework.

Lesson learned

Don't conflate the brand with the platform. A sub-brand can live as a page on your main domain — it does not need its own domain to feel real. Reserve fresh domains for things that have their own audience, mailing list, or commerce.

PHASE 2

Set up Claude as your co-pilot

You don't have to know how to code to do this. You need to know how to *describe* what you want, and you need a co-pilot that can write, click, and verify on your behalf. That's Claude.

The two Claude products I used

- **1. Claude Cowork** — desktop mode at cowork.claude.com. It can read files on your computer, run code in a sandbox, control your browser, and use connectors to your accounts (Gmail, GitHub, Calendar, Slack, etc.). This is where I did the migration work.
- **2. Claude Design** — for generating beautiful, on-brand HTML/CSS pages from a prompt. I'd describe a vibe, get back a polished page, iterate, then download.

Connect your accounts

In Cowork, install connectors for the services you'll touch:

- **GitHub** — to push code to your site repos
- **Google Drive / OneDrive** — to save and share deliverables
- **Gmail / Outlook** — for verification emails during signups
- **Calendar** — to block migration work time
- **Slack** — for real-time visitor pings later

How to work with Claude

Treat Claude like a sharp intern. Tell it the goal, share the constraints, let it do the click-by-click work. Stay in the loop on anything financial (purchases, subscriptions, payment methods) — Claude won't enter card numbers, and that's by design.

PHASE 3

Build the bones — GitHub, domains, DNS

Step 1 — Create a GitHub account

Go to github.com and sign up. Free is fine. Either use a personal handle or create an organization (I use *ToyBoxConsulting* so the repos read as a business asset, not a hobby).

Step 2 — Create one repo per site

Public repos are required for free GitHub Pages hosting. Name each repo for the site it will hold:

- **ToyBoxConsulting/katoyapalmer** — for katoyapalmer.com (the hub)
- **ToyBoxConsulting/toyboxconsulting** — for toyboxconsulting.net
- **ToyBoxConsulting/golden-hour-landing** — for the event microsite

Step 3 — Buy your domains

I use GoDaddy because that's where my originals lived. Namecheap, Cloudflare, Porkbun — any of them work. Buy the .com if you can get it. Buy the matching .net for defense if you're a brand.

Step 4 — Wire DNS to GitHub Pages

For each domain, in your registrar's DNS panel, set four A records pointing to GitHub's IPs, plus one CNAME for the www subdomain:

```
A @ 185.199.108.153
A @ 185.199.109.153
A @ 185.199.110.153
A @ 185.199.111.153
CNAME www YOURUSERNAME.github.io
```

Then add a file called **CNAME** (no extension) to the root of your repo, containing just the domain name (e.g. *katoyapalmer.com*). In repo Settings → Pages, set the source branch to **main** and root **/**. GitHub will issue a free HTTPS cert via Let's Encrypt — this can take a few minutes to 24 hours.

HTTPS gotcha

If HTTPS doesn't issue after a day, remove the custom domain in Settings → Pages, save, then re-add it. That re-triggers cert provisioning. Worked for me twice.

PHASE 4

Design your pages with Claude

This is the part Wix made you pay for and limited what you could do. With Claude, you describe a feeling and get a page.

My process

1. Open **Claude Design** (or just prompt Claude in chat).
2. Describe the page in plain English: *'A landing page for my Substack about mindfulness for high-achieving women in their 30s. Warm, editorial, lots of whitespace, soft sage and cream palette, serif headlines, sans body. Hero, three-column intro, podcast embed, signup form.'*
3. Claude returns a self-contained HTML file with CSS and any needed JS — no build step, no framework required.
4. Open it in your browser. Like it? Iterate: *'Make the hero quieter, drop the second CTA, swap headline font to Italianno for the script feel.'*
5. When it lands, download. Drop into your repo's folder.

What I built this way

All five katoypalmer.com brand pages, the entire toyboxconsulting.net site, the Golden Hour landing page, and a polished 404 page — each with its own typography, palette, and rhythm. Same architecture, different voices.

Single-file rule

Keep each page as a single HTML file. No build step. No npm. No package.json. You'll thank me when you need to fix a typo at midnight from your phone.

PHASE 5

Deploy to GitHub Pages

Once the bones and the page are ready, deploying is almost anticlimactic.

1. In your repo, click **Add file** → **Upload files**.
2. Drag your HTML files (and any assets/images) in. The main page must be named **index.html**.
3. Write a clear commit message — something like *'Initial site launch'*. Commit.
4. Settings → Pages → confirm **main** branch, / (**root**) folder. Custom domain is your *katoyapalmer.com*.
5. Wait 1–5 minutes. Visit your domain. If HTTPS isn't ready yet, it'll redirect to the github.io URL.

Keep a 404 page

Add a file called **404.html** to your repo root. GitHub Pages serves it whenever someone hits a broken link. Without it, your visitors get a GitHub-branded error page. Branded 404 = professional.

PHASE 6

SEO and search engine discovery

Hosting is solved. Now make sure Google can actually find you. This was a huge unlock — Wix abstracts most of this away, so you can't see it. With your own site, every choice is yours.

On-page SEO checklist (per page)

- **<title>** — 50–60 characters, your name first if it's a personal page
- **<meta name="description">** — 140–160 chars, the sentence Google will show in results
- **Open Graph tags** — title, description, image (1200×630px) for when someone shares your link in iMessage or Slack
- **Twitter Card tags** — same idea, X-flavored
- **JSON-LD Person schema** — tells Google you're a real person. Include name, jobTitle, sameAs (links to your social profiles, podcast, etc.)
- **Canonical URL** — prevents www / non-www duplicates from competing
- **Favicon** — bonus credibility, takes 5 minutes to add

Site-wide files

- **sitemap.xml** — a list of every URL on your site, with last-modified dates. Both Google and Bing read it.
- **robots.txt** — tells crawlers what to index. Allow everything unless you have a reason not to.
- **llms.txt** (optional but nice) — a structured summary for LLMs that crawl the web. Future-proofing.

Submit to the search engines

1. Sign in to **Google Search Console**. Add your domain. Verify via DNS TXT record or HTML meta tag.
2. Submit your sitemap URL: *https://yourdomain.com/sitemap.xml*.
3. Sign in to **Bing Webmaster Tools**. Use 'Import from Google Search Console' — verified in one click.
4. Within a few days, you'll see impressions, clicks, and the search queries bringing people in.

PHASE 7

Install your analytics stack

The single best thing about leaving Wix was unlocking real analytics. I run four tools in parallel because they each answer a different question. All free, all snippet-based — add the script to your <head>, you're done.

- **Cloudflare Web Analytics** — Privacy-first pageviews and referrers. Zero cookies. The 'how many people visited?' tool. Free, no limits.
- **Microsoft Clarity** — Session recordings + heatmaps. You literally watch how people use your site. The 'why did they bounce?' tool. Free, no limits.
- **Google Analytics 4 (GA4)** — Standard audience reporting, conversions, traffic sources. The 'how does this connect to my ads or campaigns?' tool. Free.
- **Albacross** — B2B visitor identification. Tells you which *companies* visited (not individuals). Pings my Slack when a target account hits the site. Free tier covers small businesses.

Cookie consent

GA4 sets cookies, which means EU/UK visitors legally need to consent. The simplest approach: a lightweight consent banner that defers GA loading until the visitor clicks accept. Cloudflare and Clarity (in cookie-less mode) don't need consent, so they run for everyone.

Why all four

You don't need to pick one analytics tool. Each one tells a different story. Cloudflare for the headline number, Clarity for the qualitative truth, GA for the marketing view, Albacross for sales.

PHASE 8

Mobile, accessibility, security

Mobile (do not skip)

Over half my traffic is mobile. Add a single `@media (max-width: 820px)` CSS block per page and test on your actual phone. Common adjustments: bigger headlines, single-column stacking, larger tap targets, mobile-friendly nav (hamburger or bottom bar).

Accessibility basics

- **alt text** on every image — for screen readers and SEO
- Color contrast: body text should hit at least 4.5:1 against its background
- Real `<button>` elements for buttons (not styled `<div>`s) so keyboard navigation works
- `lang="en"` on your `<html>` tag

Security headers

- **Content-Security-Policy** — whitelists which external scripts can run. Stops injection attacks. Be specific about your analytics and embeds.
- **Permissions-Policy** — disables browser APIs you don't use (camera, microphone, geolocation, etc.)
- **Referrer-Policy: strict-origin-when-cross-origin** — sane default
- **.well-known/security.txt** — a tiny RFC 9116 file telling researchers how to report bugs

PHASE 9

Legal pages and consent

Three pages everyone needs, especially if you run analytics or take signups:

- **Privacy Policy** — what data you collect, how you use it, who you share it with. List your analytics tools by name. Mention rights under GDPR and CCPA.
- **Terms of Use** — the rules for using your site. Disclaim liability for external links and user content.
- **Cookie Notice** — what cookies are set, by whom, and how to opt out. Pair this with a small consent banner on first visit.

Claude can draft all three for you in a single pass — feed it your site URL, your analytics list, and your jurisdiction. Have a real lawyer review if you take payments or handle sensitive data.

Link them in your footer

Every page in your site footer:

Privacy · Terms · Cookie Notice · Cookie Settings (re-open consent)

PHASE 10

Email and audience

Leaving Wix means leaving its email tool. You need a new home for your list.

My choice: Brevo

Brevo (formerly Sendinblue) gives you a generous free tier (300 emails/day, unlimited contacts), drag-and-drop campaigns, transactional sending, and an embeddable signup form. Mailchimp, ConvertKit, MailerLite, and Beehiiv are all good alternatives — pick what matches your style.

DNS records (boring but critical)

For your emails to actually arrive in inboxes (not spam), add three records to your domain's DNS:

- **SPF** — TXT record authorizing Brevo to send on your behalf
- **DKIM** — TXT record proving the message wasn't tampered with
- **DMARC** — TXT record telling receivers what to do if SPF/DKIM fail

Brevo (and most platforms) generates these for you. Copy and paste into your registrar's DNS panel. Verify in Brevo. Done.

Import your contacts

Export from Wix as CSV. Import to Brevo. Tag by source so you remember where each contact came from.

PHASE 11

Real-time visitor pings

One Wix feature I missed: a little ping when someone visited the site. Recreating it took two minutes and got *better*.

Albacross → **Slack**

1. Create a free Slack workspace for your business (mine is the ToyBox workspace).
2. In Albacross, go to Integrations → Slack. Authorize and pick a channel (I use **#site-visitors**).
3. When a known company visits your site, Albacross pings the channel with the company name, industry, size, and the pages they viewed.

Pair it with Clarity highlights

Clarity also offers a Slack integration for high-friction sessions (rage clicks, dead clicks). Two pings, two views: who visited (Albacross) and how they struggled (Clarity).

CLOSING

What I'd do differently, what's next

What I'd do differently

- **Plan the architecture on paper first.** I changed brand names mid-build. Costs a day of grep-and-replace each time.
- **Buy all your domains up front.** Even ones you're not ready to launch. They're \$15/year. Sitting on a name beats losing it.
- **Set up GSC + Bing on day one.** You want them watching the moment you launch — not three weeks later.
- **Write the legal pages early.** Don't ship analytics without them.

What's next on my list

- Migrate primary Google Workspace domain to the new email home
- Tax-prep / 1099 dashboard for ToyBox contractors
- Government agreements tracker for reporting deadlines
- Convert the busiest pages to artifacts that pull live data on each visit

The real takeaway

The biggest lesson: the platforms we pay for are mostly renting us things we could own. Hosting is free. Search engines are free. Analytics are free. Email gets cheap fast. What you're paying for, mostly, is the comfort of not having to think about it. Claude makes the thinking part lighter — so the savings are real.