

● ON AIR • CASE STUDY

# Orosei.ai

## How I Built Orosei.ai

*A non-technical journalist ships a website with AI. The full workflow, the decisions, and the missteps.*

GABRIEL ELIZONDO

Where Journalism meets intelligence

# Phase 1 — Setup

**What I did.** Before any building, I made the decisions that shaped everything after. The first was almost accidental, and it became my first lesson.

**The mistake I made early.** I started building inside an existing project's chat, mixing this work in with unrelated conversations. It got cluttered fast. Later I gave the website build its own dedicated project and labeled it Orosei.ai, a clean and separate workspace, and immediately wished I'd done it on day one. Lesson: give a real project its own room from the start.

**The decision that mattered most: the name.** I came in with about ten candidates. Bouncing them off Claude, I narrowed ten to four, then four to one, and stress-tested the top three before committing: how each sounded, what it implied, whether the .ai paired cleanly. But the real test was simpler than any of that. The name Orosei brought a smile to my face every time I said it. That was the whole point. Pick a name that resonates, that gives you a little joy. Don't talk yourself into a clever one that leaves you cold. I locked in the name early and never looked back.

**The other big fork: how to build it.** Two paths. The easy one (drag-and-drop: upload your files to a host and you're done), or the harder one (build through Claude Code, the developer's workflow, with version history and proper deployment). For me it was an easy call. I chose Code, because I wanted to practice and learn the real workflow, not just get a site online. That choice cost me time up front, and again during deployment, but post-launch it has paid off every time, making updates quick and painless.

## Compared to where we've been

| Name          | Pronounceability | Distinctiveness | Story strength | Domain neighborhood  | Scales beyond journalism | Personal connection |
|---------------|------------------|-----------------|----------------|----------------------|--------------------------|---------------------|
| Orosei        | B+               | A               | A              | A                    | A                        | A+                  |
| Seulo Labs    | B-               | A+              | A+             | A                    | A                        | A-                  |
| Sardo Labs    | A+               | B+              | B+             | A-                   | A                        | B                   |
| Sardinia Labs | A+               | B-              | B              | C (crowded category) | A                        | B+                  |
| Cala Labs     | A+               | B               | B+             | C (crowded)          | A                        | A                   |



Stress-testing the finalists. Note the earlier 'Labs' framing I later dropped.

## Phase 2 — Building

---

**What I did.** First I secured the foundation. There are a million pieces of software out there, and the hardest early work was just figuring out which ones fit my project. I researched it, with Claude's help, and landed on two: Porkbun, a domain registrar, to buy orosei.ai (declining a parade of upsells for hosting and email I didn't need), and Netlify, a free service, to host the site and put it online. Neither was an obvious pick to me at the start; both came from asking questions, comparing options, and deciding what suited my product rather than grabbing the first name I recognized. Then came the real work: the pages. I didn't set out to build seven. I started with one, and as the idea took shape it grew. One became three, three eventually became seven. Don't lock yourself into a blueprint on day one. Build the house, then see how many rooms you actually need.

**The decision that mattered most: a design system.** Early on I locked in rules and refused to break them: four colors, two fonts, a broadcast-style “ON AIR” bar across the top. Every page had to obey. It sounds rigid, but the constraint is what makes the site feel like one thing instead of seven. When I was tempted to add a fifth color or a flashier font, the rule won. Discipline read as design.

**Judgment, again and again.** Here's the thing I keep coming back to. Claude helped enormously, but at every fork in the road, the call was mine. Which name. Which photo. What to cut. When to keep a rule and when to bend it. The AI is a brilliant collaborator, but it doesn't have your taste or know your audience. The judgment has to be yours. And it turns out a journalist's instincts for what's true, what's clear, and what matters transfer pretty well to building a website.

**A note on the tools themselves.** I built most of this on Claude Opus 4.8. Partway through, Anthropic released a more powerful model, Fable 5, and I switched to it, mostly out of curiosity to test it, not because the work demanded it. Three days later Fable was pulled from public use under a government order, and I was back on Opus 4.8 to finish. Two lessons in that small detour: you can change models mid-project without drama, and newer isn't always necessary. Match the tool to the task. Don't chase the shiny new thing just because it's there. For what I was building, the workhorse model was always enough.

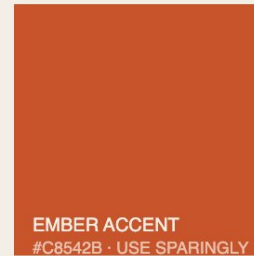
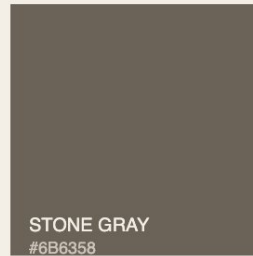
**What tripped me up, and the real lesson of this phase.** Iteration. So much iteration. The page explaining my site's name went through more than a dozen versions in a single sitting. The About page, nearly as many. I swapped photos, rewrote lines, moved buttons, then moved them back. The lesson: don't rush it. Make a change, make more, sleep on it, come back. It's like buying a car. You don't take the first one on the lot that catches your eye; you keep looking until you're sure you've got the one you'll be happy with. The tools make rewriting cheap, so use that freedom.

**The other lesson: know when to stop.** Here's the balance, though. Endless revision becomes its own trap. The hard part isn't making changes; it's knowing when a page is done and refusing to reopen it. When do you keep iterating, and when do you lock it in? There's no formula. It's judgment, again, the same call I kept making at every fork. At some point I started marking pages finished and moving on, because otherwise nothing would ever ship.

# Orosei

AI STRATEGY · TRAINING · ADVISORY

## 01 — PALETTE



An early color direction. I went a completely different way for the final site.



The US government, citing national security authorities, has issued an export control directive to suspend all access to Fable 5 and Mythos 5 by any foreign national, whether inside or outside the United States, including foreign national Anthropic employees.

The net effect of this order is that we must abruptly disable Fable 5 and Mythos 5 for all our customers to ensure compliance.

Access to all other Claude models is not affected.

We apologize for this disruption to our customers. We believe this is a misunderstanding and are working to restore access as soon as possible.

Anthropic's announcement that Fable 5 was being suspended, mid-build.

## Phase 3 — Deployment

---

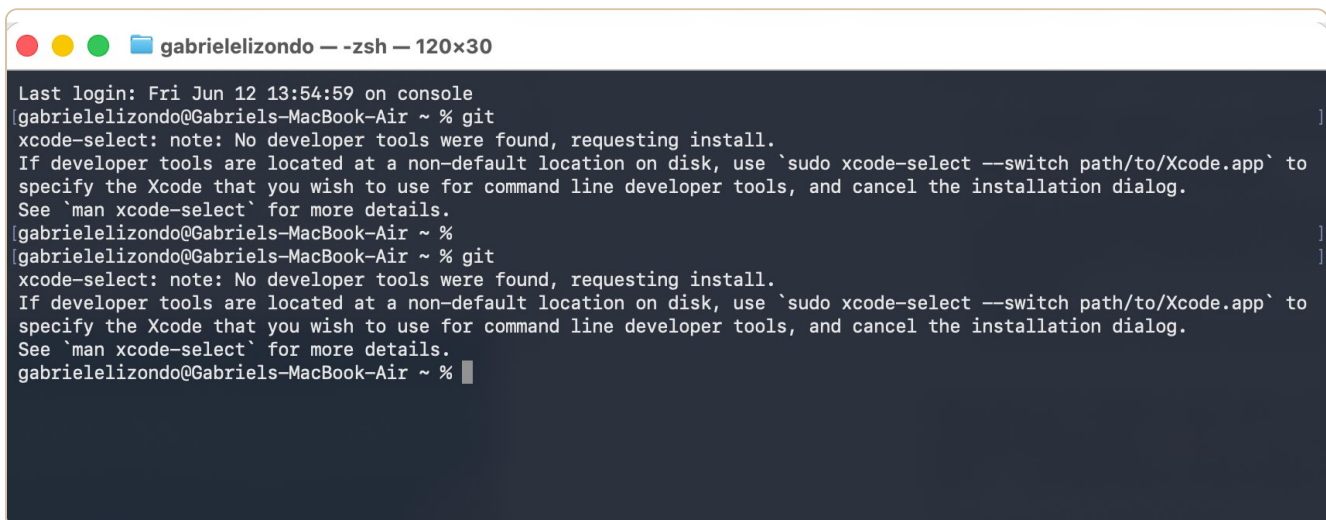
**What I did.** This was the day the site went from files on my laptop to something the world could reach at orosei.ai. The path had a lot of steps: install the tools, put my project under version control with Git (the system that tracks every change), push it up to GitHub (an online home for the code), connect that to Netlify (my host), wire up the contact form, and finally point my domain at it through its DNS settings. None of it was conceptually hard. All of it was unfamiliar, and that combination is exactly where a beginner sweats.

**How I actually worked.** Not from one screen, but three. I had the Claude Code desktop app open to do the building, Claude on the web for thinking through problems and getting my bearings, and the Claude app on my phone for checking things on the side. I toggled between all three the entire way through, using each for what it was best at. It looked chaotic. It worked.

**What tripped me up (all of it, honestly).** The snags came in a steady drip. Git wasn't installed on my new laptop, and the install prompt hid behind another window, so I sat there confused before finding it. Once installed, Claude Code couldn't see it until I fully quit and reopened the app. My contact form didn't work on the first try and needed a second deployment before Netlify would detect it. And in the middle of repointing my domain's nameservers, I hit a page showing a cartoon pig telling me orosei.ai was “parked,” and I was sure I'd broken something. I hadn't. I'd just not finished the last step.

**The decision that mattered most: push through, don't panic.** Every one of those snags looked, in the moment, like the thing that would stop me. Not one of them did. The pattern I learned: read the actual message, do the next small step, and when I was stuck, paste the exact error to Claude and ask what it meant. Nothing was ever truly broken. Everything was recoverable. That mindset, more than any technical skill, is what got the site live.

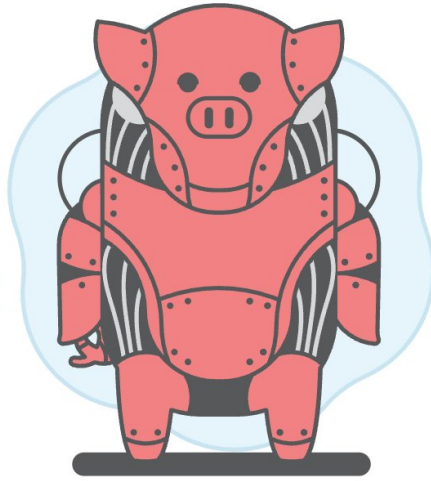
**And then it was live.** I swapped the domain's nameservers over to Netlify, waited for the internet to catch up (DNS changes take time to spread worldwide), and refreshed. The cartoon pig was gone. There was my globe at orosei.ai, its signal dot tracing the same path it always had. An idea I'd typed into a chat box weeks earlier was now a real website anyone could visit.

A screenshot of a terminal window titled "gabrielelizondo — zsh — 120x30". The terminal shows the user running the command "git", which results in an "xcodeselect" error message. The message states: "xcodeselect: note: No developer tools were found, requesting install. If developer tools are located at a non-default location on disk, use 'sudo xcodeselect --switch path/to/Xcode.app' to specify the Xcode that you wish to use for command line developer tools, and cancel the installation dialog. See 'man xcodeselect' for more details." This message is repeated twice. The terminal background is dark blue, and the text is white. The window has standard macOS window controls (red, yellow, green buttons) at the top left.

```
gabrielelizondo@Gabriels-MacBook-Air ~ % git
xcodeselect: note: No developer tools were found, requesting install.
If developer tools are located at a non-default location on disk, use `sudo xcodeselect --switch path/to/Xcode.app` to
specify the Xcode that you wish to use for command line developer tools, and cancel the installation dialog.
See `man xcodeselect` for more details.
gabrielelizondo@Gabriels-MacBook-Air ~ % git
xcodeselect: note: No developer tools were found, requesting install.
If developer tools are located at a non-default location on disk, use `sudo xcodeselect --switch path/to/Xcode.app` to
specify the Xcode that you wish to use for command line developer tools, and cancel the installation dialog.
See `man xcodeselect` for more details.
gabrielelizondo@Gabriels-MacBook-Air ~ %
```

The first snag: Git wasn't installed yet.

# Parked on the Bun!



**orosei.ai** has been registered at [Porkbun](#) but the owner has not put up a site yet. Visit again soon to see what amazing website they decide to build.

**Find your own great domain:**

The 'parked' page that made me think I'd broken everything.

---

**The takeaway.** It's worth sitting with this for a second. A few weeks earlier, this was a sentence in a chat box. Now it was a live website with my name on it, a working contact form, and a domain I pointed across the internet myself. I'm a journalist. I don't write code. And I built and shipped this with AI as my collaborator and my own judgment at every fork. That's the whole point of this page: not that I'm special, but that the barrier between an idea and a real, working thing is lower than most people think. If you've been wondering whether you could do something like this, that's your answer.