From Zero to Public

Fifty internet properties, built in the open.
Every launch, every number, every lesson.

GERARDO · CODEAUSTRAL

Typeset in Source Serif and Fraunces · zerotopublic.com

Written, designed, and typeset with the help of AI agents — which is, in itself, half the point of this book.

Contents

PART II

Fifty Properties

01

foodphoto.ai

AI food photography for restaurants — and the only property in this book with confirmed revenue.

What it is

FoodPhoto takes the phone photo a restaurant owner snaps of their burrito under bad fluorescent light and turns it into something that looks like it came out of a studio shoot — lit, styled, color-graded, ready for a delivery app listing or an Instagram grid. Under the hood it's a Next.js frontend talking to a Python FastAPI backend, with Postgres and Redis behind it and a smart router that picks the best image model for each request from a handful of providers. You buy credits through Stripe, you upload, you download. There are about thirty presets and thirty-five styles, and yes, I have had to verify that both of those numbers were actually true on the marketing pages, because at one point they weren't consistent.

Why I built it

I come from the restaurant world adjacent — several of my properties orbit food and gastronomy — and the pain is obvious the moment you scroll any delivery app: the food that sells best is the food that photographs best, and most independent restaurants can't afford a photographer. Image models had just gotten good enough that "make my food look professionally shot" went from gimmick to genuinely useful. It was the rare idea where the customer, the pain, and the willingness to pay all lined up in one sentence.

The build

This is the property I have invested the most agent-hours into, by far. At one point I had six named AI agents working it in parallel — one owned the frontend build, one owned the backend, others did SEO, copy, audits — with explicit lane rules so they wouldn't trample each other, because earlier two agents building the same Next.js repo at once took the site down for forty-five minutes. A later sprint ran eight agents and shipped sixty programmatic SEO pages, twenty-five blog posts, thirty translated pages, and an alerting stack in a day. The site now carries close to three thousand static SEO pages — cuisines, dishes, cities — plus free demo tools on dish pages, rate-limited to two uses per IP per day so the demo doesn't bankrupt me.

The launch

There was no launch day. FoodPhoto accreted in public: pages went up, the checkout went live, the free tools started catching search traffic, and orders began arriving. I made a deliberate decision not to buy ads — growth here is SEO, free tools, and partnerships, full stop. That choice keeps the economics honest: every sale is someone who found the thing and wanted it, not someone I paid to click. The launch-adjacent work that actually mattered was trust repair. At one point the site carried phantom offers — a three-dollar tier here, a nineteen-dollar package there — that didn't match the canonical pricing in the backend, the kind of drift that happens when many agents touch many pages over many sprints. A buyer who notices the price on the pricing page disagrees with the price at checkout doesn't email you about it; they just leave. So one full day went to nothing but reconciling every public number against the single source of truth, and "canonical SKUs only" joined the rulebook.

The numbers

This is the one property where I can say it plainly: real strangers have paid real money, repeatedly. I won't dress it up with a vanity revenue figure, but FoodPhoto is the confirmed cash engine of the portfolio — which is exactly why its failures hurt the most. One morning a single Python import error crashed the backend and paid orders failed silently for most of a day before I caught it. Another time a concurrent agent session deleted fifty-seven route directories from the app — most of the site — and because I don't run git ceremony on these projects, recovery meant snapshots and reconstruction, plus a new rule: snapshot before every build, verify the route tree before every deploy. And for two days the app silently dropped every transactional email because a deleted token file made the send function return a polite "success" without sending anything. Each incident produced a written rule. The rules are the real asset.

The lesson

The property that makes money is the property where sloppiness becomes expensive. On a parked experiment, a broken checkout is trivia; on FoodPhoto it's an outage with a dollar cost and an angry customer attached. So this is where I learned the operational discipline the rest of the portfolio borrows: one builder per repo, snapshots before builds, verify the money path end-to-end after every change, and never trust a function that returns success without checking what it actually did. Earn revenue first; the revenue will then teach you everything else.

That was chapter one of fifty.

The other forty-nine — courses, calculators, AI tools, games, shells, and the dead domains too — are in the full book, alongside Part I: the operating system that runs them all on one server with AI agents.

187 pages · $10 · zerotopublic.com