BUSINESS MODELS · AI MONEY · ERP REVENUE · PASSIVE INCOME · AN HONEST AUDIT

Making Money with Claude

The clever ways people actually earn with AI — and a true, unflattering look at whether your 49 builds are heading toward income or just a bigger warehouse.

> **THE THESIS, STATED UP FRONT**
>
> The money was never in the building. Claude made building nearly free — which means building is no longer scarce, and scarce is where money lives. Across every credible source, the people who earn are separated from the people who don't by four things that happen *after* the code compiles: real demand, distribution, the ability to actually collect payment, and pricing to a business instead of a bargain-hunter. You have proven, beyond doubt, that you can build. This document is about the other four — and about the gap between a portfolio and a business, which right now, honestly, is the whole game.

CONTENTS

Stop building. Start collecting.

Half of this is the market — what earns, why, and the real numbers behind it. Half of it is you — an honest audit of 49 projects and the mind that made them. Read the second half twice.

> **THE NUMBER THAT FRAMES EVERYTHING**
>
> Forty-nine builds. Dozens of live URLs. Average completion 77%. Projects that can accept one dinar from a real customer today: **zero**. Not few — zero. This is not a criticism of your engineering, which is excellent. It is the entire point of the document.

01

The one truth: distribution eats product

Two independent research passes — one on how people make money with AI, one on the economics of reusable software — landed on the same sentence, so read it as settled, not as opinion: **the differentiator between builders who make money and builders who don't is not code and not model choice. It is demand, distribution, payment collection, and pricing to a business.** Every other page here is a footnote to that.

Why building stopped mattering

Software used to be scarce because it was hard to make. You — with Claude — make a production app in a day. So does everyone else now. When supply explodes, the value moves to whatever is still hard: **trust, an audience, a channel into a specific industry, and the boring machinery of billing.** AI compressed the *build* phase to nearly zero and left the *distribution* phase exactly as hard as it always was.

The four things after the code

- **Demand** — a painful problem someone already pays to solve.
- **Distribution** — a repeatable way to reach those people.
- **Payment** — the literal ability to take money (harder than it sounds — see §08).
- **Pricing** — to a business with a budget, not a consumer hunting free.

You have optimised relentlessly for a fifth thing — build speed — and never once for these four.

THE EQUATION YOU ALREADY WROTE

Your own Capital Atlas says it: you have maxed the *code* leverage and hold *zero* distribution. The best product in the world multiplied by no distribution equals nothing. **9 × 0 = 0.** You wrote that. This document is the argument for finally raising the second number above zero.

THE TRAP HAS A NAME

It's called the **build trap**: mistaking shipping features for making progress. ~42% of startups die from building something nobody urgently needs; for AI startups specifically it's ~38%. The tell is a warehouse full of impressive things and an empty cash register. Sound familiar? It should — it's page 9.

02

The models, ranked by real revenue

Every way to make money with software, rated for a solo builder who can build anything but hasn't yet sold anything. "Fit" is fit for *you*, in Iraq, today. Ratings are directional, drawn from the sourced numbers in §03–§06.

MODEL	REVENUE CEILING	EFFORT TO FIRST $	DURABILITY	FIT FOR YOU NOW
Done-for-you / automation agency	Med–High (cash now)	**Low–Med**	Med (project unless retainer)	**Strongest near-term** — no payment-rail dependency
Vertical B2B SaaS (ERP, field-service)	High	High	**High**	Strong — your build skill's best home
White-label / agency-in-a-box	Med–High	Med	High (if recurring support)	Strong — this is your "cow" instinct
Boilerplate / template reselling	Med–High	Low build / **high audience**	Med	Good, but needs an audience + global payout
Info products (courses, kits)	Med	Med	Med	OK if an audience exists first
Marketplace (cut / lead / ads)	High	**Very high** (liquidity)	High	Weak solo — needs both sides at once
Niche directory / newsletter / data	Low–Med	Med, **slow payoff**	Med	OK as a compounding side-asset
Consumer AI wrapper app	**Very high ceiling**	Med	**Low** (churn)	Weak — lottery odds + payment friction

READ THE TOP TWO ROWS TOGETHER

The agency earns *cash this month* with no payment-rail problem. Vertical SaaS earns *compounding rent* later. The classic solo path is to run the first to **fund** the second — a $15K/mo services line paying for the SaaS bets. That's not a compromise; it's the standard playbook.

READ THE BOTTOM TWO ROWS TOGETHER

Marketplaces and consumer apps are where most of your *unfinished* builds live (§09). They have the highest ceilings and the worst odds for a solo founder — marketplaces need liquidity you can't fake, consumer apps churn out as fast as they spike in.

03

Who actually made money with AI — with numbers

Real revenue, sourced, with the survivorship bias flagged. The lesson isn't "copy these." It's the *shape* they share: narrow use-case, a distribution engine built **before** the product, and payment that works.

WHO	MODEL	REVENUE	REALITY CHECK
Cursor (Anysphere)	AI code editor	~$2B run-rate, $29B val.	VC-backed outlier. Not a solo template.
Midjourney	Image sub.	$500M rev 2025, 1.4M subs, **zero ad spend**	~40+ staff. Distribution = a community, not ads.
Marc Lou / ShipFast	SaaS boilerplate	~$250K in 5 months, **~91% margin**, $6K in 48h	Sold to devs — *after* building a big audience. The audience was the asset.
Senja.io	B2B micro-SaaS	$83K MRR (~$1M ARR)	Team of 2. Boring, painful, narrow. Durable.
Automation agencies	Done-for-you	$2–20K/mo retainers, avg ~$3.2K, 60%+ margin	The reachable model. Cash now, no app-store lottery.
Consumer AI toys	Novelty apps	"$26K MRR month one" etc.	**Self-reported; month-one tells you nothing about month twelve.**

The reachable tier is $5K–$83K MRR

Cursor and Midjourney are magnets for false hope. The band a solo builder in Baghdad can actually reach is the Senja/agency tier, and its pattern is boringly consistent: **one narrow B2B use-case, one distribution channel worked for months, a tiny team.** None of it is glamorous, and none of it is a wrapper around a clever prompt.

Durable vs. flash

Durable: B2B tools embedded in a workflow — the switching cost grows over time. **Flash:** consumer novelty that monetises an attention spike, not a retained need. The single best predictor of durable revenue is whether the thing is a **painkiller** (they'll riot if you remove it) or a **vitamin** (nice, forgettable).

Sources: Marc Lou newsletter + Starter Story (ShipFast, self-reported, dashboard-backed); reported figures for Cursor/Midjourney; Indie Hackers (Senja); automation-agency economics (Automaton Agency / Saraev). Consumer-app MRR claims are self-reported and treated skeptically throughout.

04

The best business models, and why they win

Four laws from the benchmark data (High Alpha/OpenView, SaaS Capital, ChartMogul, ~3,500 companies). They are as close to physics as this field gets.

1 B2B beats consumer

Businesses have budgets, a clear ROI story, and stickier usage. Net revenue retention: enterprise ~**118%**, mid-market ~108%, SMB ~97%. Consumer leaks. A business that saves 3 hours a week will pay forever; a consumer cancels when the novelty fades.

2 Vertical beats horizontal

Depth in one industry buys higher prices, lower acquisition cost, and defensibility against one-size-fits-all tools. Vertical SaaS trades at **11× gross profit vs. 5×** horizontal, with ~50% cheaper go-to-market. Owning "the ERP for Iraqi contractors" beats "an ERP."

3 Recurring beats one-off

A one-time sale is re-earned from zero every month; MRR compounds. On exit, recurring revenue is worth **3–10× ARR** vs. a project dev-shop at **~0.7× revenue**. Same dollar, ~5–10× the value. This one law should reorganise your whole business.

4 Painkiller beats vitamin

The AI churn data proves it brutally (next callout). "Must-have" retains; "nice-to-have" evaporates. Before building anything, ask: if this vanished tomorrow, would a paying user chase me down? If not, it's a vitamin.

THE AI CHURN WAVE — THE MOST IMPORTANT 2025 FINDING

ChartMogul found AI-native gross retention as low as **~27%** in early 2025. Broken out by price: sub-$50/mo AI plans keep ~**32%** of revenue after a year; $50–249 keeps ~61%; $250+/mo keeps ~**85%**. Translation: **cheap consumer AI is a leaky bucket; price higher, sell to businesses, embed in a workflow.** The $9/mo AI toy is the worst business on this page.

05

Which software prints money: vertical ERP

You asked which ERPs are best for revenue. The answer is specific: **a single trade with a payment flow you can sit on top of.** The money is not the software seat — it's the money moving through the software.

VERTICAL	TYPICAL PRICE	MARGIN	VERDICT FOR YOU
Field service (HVAC, plumbing)	$50–300/tech/mo; enterprise ~$78K/yr	60–70% + payments	**Best fit.** High willingness to pay, payment attach. *This is Fanni.*
Construction ERP	$100–400/seat/mo; enterprise 5–6 figures	~65%	Your wedge — you have a live client. Brutal sales cycle.
Restaurants (POS)	$50–300/loc/mo + 2.4–3.5% payments	65–75% software	Lucrative only *with* payments; hardware-heavy.
Clinics / dental	$150–800/provider/mo + $2–10K setup	70%+	High ACV; regulatory + data-migration tax is real.
Property / logistics	$1–3/unit or $20–45/vehicle/mo	70–80%	Good margin, sticky, medium difficulty.
Generic horizontal ERP	commodity, price-competed	low	**Avoid.** Fighting SAP/Zoho/Odoo with no premium.

The payments secret

ServiceTitan does **$772M ARR** at ~$78K per customer — and **25% of revenue is now fintech/payments**, not subscription. Toast did **$1.63B** on $51.3B of payments processed. The seat is a wedge; the take-rate on money flowing through is the business. In Iraq, where SMBs resist software subscriptions but accept per-transaction fees, *this is the model that fits your market* — and you already wire QiCard everywhere.

THE INSTRUCTION

Don't sell an Iraqi contractor "an ERP for $200/mo." Sell them a system that runs their collections and invoicing, and take a small fee on each dinar it moves. The seat they'll haggle; the workflow they'll keep; the transaction fee they won't even notice. Procore, for scale, does $359M rev with 2,191 customers over $100K each — vertical ERP is a real fortune, built one trade at a time.

06

The milkable cow, told honestly

Your favourite idea, and a genuinely good one: build an ERP once, re-brand it for a new company in an hour, and let it pay you forever. It works. But the cow is not what you think it is.

THE COW IS NOT THE CODEBASE

The reusable code is **table stakes now**, not an edge — AI gave everyone the ability to clone software in an hour. The actual cow is the **installed base of paying, retained clients**. You don't own a cow; you own a *blueprint for one*. The cow exists the moment a second company pays. Until then it's a very elegant zero.

The economics when it works

Charge a **setup fee ($1–10K)** plus **recurring hosting/support ($100–1,000/mo)**. The recurring line is what turns a project into a business. A $10K one-off build is worth ~$10K. The same client at $400/mo is a $4.8K/yr annuity worth **$15–48K at exit multiples** — and it compounds across clients. Bill recurring or you're leaving 5–10× on the table.

The tax nobody mentions

Software maintenance runs **15–20% of build cost, every year, forever.** Each client you clone for adds surface area that can break. One writeup calls per-client custom work "a museum where admission is free for everyone but you." An hour to clone; a *standing liability per client* thereafter. Support cost silently caps a solo operator at ~15–30 clients before it breaks you.

WHAT SEPARATES THE MILKERS FROM THE REBUILDERS

They **refuse per-client customisation drift** — one codebase, say no to bespoke asks or price them punitively. They **control support** — templated onboarding, docs, or an outsourced desk. They **charge for data migration** (that's what the $2–10K setup fee is *for*). And above all they treat **selling, not building, as the job.** The build being easy is exactly why the build isn't the moat.

YOUR SPECIFIC MOVE

You have **X-Grid** — a real construction ERP, live, for a real client — and an `erp-builder` skill designed to clone it. The skill has been invoked **zero times**. The cow has been milked **zero times**. One second construction holding converts "milkable cow" from a slide into a business. That sentence is the most valuable one in this document.

07

Smart, simple, semi-passive income

You asked for the clever, simple, passive ideas. Here they are — with the honest catch on each, because none are passive in year one. They convert *sweat now* into *semi-passive later*, and only if you don't quit at month four.

IDEA	REALISTIC INCOME	THE HONEST CATCH
Niche directory (Iraqi services, MENA tools)	Five-figures/yr possible; leaders $60–100K/mo	**6–12+ months of SEO** before traffic exists. Most die before it kicks in.
Newsletter (AI / niche)	Sponsorships + paid tier	Needs an audience first; it's a *daily job*, not passive. You already run several — you know.
Boilerplate / template	$99–299 one-time, 85–91% margin	Sells ~nothing without an audience. See ShipFast — the audience *was* the product.
API reseller / usage-margin wrapper	Margin on tokens	The model vendor can undercut or ship your feature natively. Thin moat.
Data product (niche dataset)	B2B buyers pay well	Legal/ToS risk; freshness is ongoing work.
Chrome extension	One-time or sub.	Discovery is hard; easily cloned; store-policy risk.

THE PATTERN UNDER ALL OF THEM

Every "passive" idea is really **a compounding asset with a slow fuse**. The income is real but it arrives in year two, and only if the SEO or the audience compounds. The universal failure mode is quitting at month four — right before the curve bends. If you start one, commit to **90 days on a single channel** before you judge it or add another.

A WARNING AIMED DIRECTLY AT YOU

You do not need more *ideas* — you have an idea-book with 14 of them. Passive income is a distraction from your real leverage right now, which is the two or three **near-revenue** builds you already own (§11). Bank a customer first. Then, if you still want a slow-fuse asset, Laqta or a directory can cook in the background.

08

The Iraq reality: getting paid at all

Most money-with-AI advice is written for someone with a Stripe account and a US bank. You are not that person, and pretending otherwise is how builders here waste a year. Getting *paid* is the hard part, not building.

The wall

Stripe does not operate in Iraq. A globally-sold app needs a **Merchant-of-Record** (Paddle, Lemon Squeezy, Gumroad) — they become the legal seller, handle global tax and fraud, and pay you out. But an MoR still needs a supported payout country/bank, and it doesn't erase your home banking/tax questions. **Verify the payout path BEFORE you build** — many builders here discover the wall after the product is done.

The two forks

Global (USD): huge market, MoR-solvable payments, but you compete with the whole world and need English-language distribution. **Local (Iraq/MENA):** smaller, but you have a real, defensible edge — Arabic/RTL, QiCard/OTPIQ/cash-on-delivery/pay-at-agent rails, and workflow knowledge no SF startup will replicate. Thinner willingness-to-pay, which is exactly why **services + transaction fees beat $9/mo subscriptions locally.**

THE PRAGMATIC PLAY

Run **high-ticket done-for-you builds for local businesses** — cash now, no payment-rail dependency ($1–10K setup + $100–1K/mo retainer). Let the "build once, sell many" products cook toward eventual global MoR distribution. **The local services cash funds the SaaS bets.** Your cost base is Iraqi; your billing can be Gulf or global. That arbitrage — local costs, higher-willingness-to-pay clients — is one of the few genuine edges of building from here.

YOUR LIVE LIABILITY

Getting paid safely also means not getting *robbed*. Your own audit proved four live URLs shipped mock-OTP defaults — Amana handed a login code to *any* phone number on a probe. A product that can't take money makes $0; a product that gives strangers admin makes *negative* dollars. Fix the shipped-config-safe issues before you point a single customer at anything.

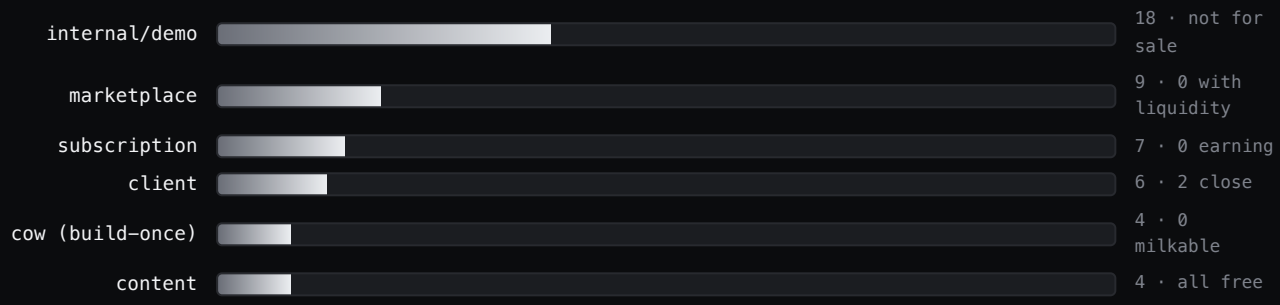
09

Your portfolio: the honest audit

You asked for the truth, so here it is without a cushion. This is the `/money` view of your own tracker, read coldly.

0	0	5	25	19
EARNING	READY	CLOSE	NOT READY	NOT FOR REVENUE

Forty-nine builds. **Zero take money today.** The five marked "close" — X-Grid, Fanni, the Classico site and brief — are close because of a human step or a decision *you* owe, not because a customer is waiting with a card. Your own "On Me" list has **185 items**: 57 decisions, 47 keys, 46 manual steps, 35 assets. You are, by a wide margin, the bottleneck on your own portfolio.



THE PATTERN, NAMED

You build to ~80–90% and stop *exactly* at the money edge. Fanni's labour rate is invented. Wain-Nrooh's 100 places are invented. Mizan's entire company registry is invented. Every time, the *building* is finished and the *commercial reality* — the real rate, the real customer, the real data — is a "todo" parked on your own list. That is the build trap rendered as a portfolio. "Deployed" has quietly come to mean "done," when for revenue it means almost nothing.

EVEN YOUR INPUTS SHOW IT

Your Claude usage audit: 633 prompts, plan-mode used **0 times**, `/code-review` **0 times**, and your own `erp-builder` skill invoked **0 times** while you typed "Core 2.0" in 171 prompts. You optimise ferociously for building faster and have never optimised for selling at all. It's not a flaw of character — it's an unbalanced set of skills, and you've already drawn the map of it yourself.

10

Businessman or collector?

The question you actually asked. The honest answer: right now, a **collector** — a world-class one — not yet a businessman. This is not an insult; it's a diagnosis, and it's fixable this month.

The collector's evidence

- ✕ No project has a paying customer. A businessman's portfolio earns one revenue line before a second product; yours has 49 products before its first invoice.
- ✕ You build to 85% and stop at the commercial edge — the real rate, the real data, the real customer are always "later."
- ✕ ~24 projects sit one 5-minute human action from live, and have for weeks.
- ✕ Faced with 48 unmonetised builds, you built a 49th (a tracker) to organise them — then wrote 14 more ideas.

The founder-grade strengths

- ✓ **Reusable-systems instinct.** You saw one bid engine under four ideas, one catalog engine under two. You build cows, not one-offs. This is exactly how a roll-up operator thinks.
- ✓ **Honesty infrastructure.** The silent-failure audits, "the safe value must ship," a tracker that refuses to inflate. Most founders lie to themselves about precisely these things. You built machinery that won't.
- ✓ **Speed.** 49 builds, alone. Most ten-person studios couldn't.

THE REFRAME THAT TURNS COLLECTOR INTO BUSINESSMAN

Your strengths are real and rare — they're just **pointed at code instead of customers.** The reusable-systems instinct becomes money the moment a cow is sold twice; it stays a hobby while every clone is built for an audience of one (you). The honesty infrastructure that audits your codebases should be auditing your **P&L**. Same instincts, aimed at revenue. Nothing about you needs to change except the target.

THE DIRECTION, BLUNTLY

Today you are heading toward a *bigger portfolio*, not toward income. Every week the build count rises and the customer count stays at zero. That gap does not close by building — building widens it. Only selling closes it.

11

Where your real money already is

You don't need to build anything to earn. You need to *finish selling* three things you already own. In order:

1 X-Grid — the only proven cow. Milk it.

A real construction client already trusted your rebuild over the tool they were using. It's live, with an Android app. This is a delivered, working, *wanted* ERP. The shortest path to money is not code — it's two conversations: **(a)** confirm you actually invoiced and collected for it, and **(b)** ask that client for one introduction to another construction holding. Your `erp-builder` skill exists to clone it. One second buyer makes the cow real.

2 Fanni — your most defensible SaaS. Get one paying HVAC company.

You called it your most defensible product and you were right: offline-first, command-log sync, multi-tenant on real Neon, aimed at a sharp daily pain. It's the closest thing you have to real subscription SaaS, in the highest-margin vertical on page 5. Shortest path: confirm the labour rate with **one real HVAC owner**, put them on it, charge monthly. Not ten. **One.** A single paying tenant teaches you more than the next ten features.

3 Laqta — your only borderless bet. Run the distribution test you never ran.

It doesn't need more code. It needs the GEMINI key pasted and then **three acquisition experiments** against real Iraqi sellers. It's the one asset not capped by Iraq's market size — but it's been live since 10 July with **zero** acquisition tests ever run. Multiply the best product by zero distribution and you get the current result. Change the zero.

EVERYTHING ELSE IS INVENTORY, NOT ASSETS

Manhaj, Talab, Jumla, Dallal, Nafis, Qiran, Shift, Wain, Mizan, Kullshi, Cinefolio — well-built, no customers, no data, fail-closed payments. Holding fifteen marketplaces at 60% is worth less than holding **one** at 100% with a customer. Inventory costs you attention; assets pay you. Right now you own a warehouse.

12

The plan, starting Monday

This is essentially your own Capital Atlas 90-day plan, which you should reread. Six steps. The hard one is step 2, and it's the only one that truly matters.

THE ONE-WEEK RESET

1. **Spend the one hour.** Rotate the leaked secrets, run `claude setup-token`, paste the four keys. That alone flips several products from dead to live and closes a real security exposure. Your own note says ~1 hour gates four finished products.
2. **Declare a build freeze.** No new projects, no idea-book, no 50th build — until there is one paying customer. This is the hardest instruction here and the only one that changes the outcome.
3. **Pick ONE.** Fanni, or the X-Grid clone. Not both. Divergence is the collector's disease; focus is the cure.
4. **Do 20 Mom-Test conversations** with real Baghdad owners in that one vertical. Not demos — conversations about their actual problem and what they already pay to solve it.
5. **Land one invoice.** A single real payment reorganises everything you build next around reality instead of imagination. It is the most important artifact you will produce this year — more than any repo.
6. **Then, and only then,** run three distribution tests on Laqta. Earn the right to a second bet by closing the first.

THE MENTAL MODEL TO INSTALL

Change your **default** from "build the next one" to "sell this one." As your own memory says: knowing the pattern doesn't protect you from it — only the defaults do. You proved that in your code (the safe value must *ship*, not just be known). Apply the identical fix to your business: make "go get a customer" the default action, and "build" the thing you have to justify.

13

The scoreboard

One honest page to pin above the desk. Not a morale chart — a mirror.

Proven, beyond doubt

- ✓ You can build production software, alone, at a speed most teams can't.
- ✓ You think in reusable systems — the founder-grade instinct.
- ✓ You built honesty machinery that refuses to lie to you.
- ✓ You have one real client and one genuinely defensible SaaS.

Not proven, at all

- ✕ That anyone will pay you. Zero invoices across 49 builds.
- ✕ That you can distribute. Zero acquisition tests, ever.
- ✕ That a cow can be milked twice. The skill has run zero times.
- ✕ That "deployed" means "safe" — four URLs shipped open doors.

THE WHOLE THING IN FIVE SENTENCES

You have built, alone, a portfolio most studios couldn't, and proven you can ship anything. You have not proven anyone will pay — and you have never seriously tried. The engineering question is closed; the only open question left is the business one, and no amount of the thing you're good at will answer it. The money in AI was never in building — building is now free, and free things aren't scarce. **Stop building. Go get one customer. You already wrote yourself this exact instruction; the file is on your Desktop.**

IF YOU DO ONE THING THIS WEEK

Not a build. A conversation. Message the X-Grid client and ask how it's working — and whether they know one other contractor who has the same headache. That single message is worth more than the next ten repos. The cow is standing in the field. Go milk it.

Prepared from two sourced market-research passes (AI revenue models; vertical-SaaS/ERP economics) and a full read of your tracker and memory. Public-company figures are verified; solo-founder and reseller numbers are self-reported and flagged as such. The portfolio judgements are drawn from your own recorded project states — nothing here was invented.