THE PLAYBOOK FOR THE PART NOBODY TEACHES

Build Then What

You already did the hard part. You built the app.
This is the system for getting strangers to find it, understand it, and buy it.

Built on a real app that had 584 visitors and zero customers —
and every fix we actually made to it.

What's inside

> Every phase ends with a **Do This Now** walkthrough — the literal steps to take this week, plus a box showing those same steps run on the real case-study app.

You Built It. Nobody Came.

Build Then What

The true story of an app with real traffic, a real product, and zero customers — and the exact leak that was killing it.

The setup

Here's a situation you might recognize.

You have an idea. You use one of the new AI build tools — Lovable, Bolt, Cursor, v0, whatever — and for the first time in your life, the thing in your head becomes a real, working app. Real database. Real login. Real features. Months ago this would have meant hiring a developer or learning to code for a year. Now it's a weekend.

You ship it. You buy the domain. You wire up payments. You post it.

And then… nothing.

Not "slow." Nothing. No signups. No trials. No sales. Just a dashboard that says "0 current visitors" more often than you'd like to admit.

This is the wall that almost every indie builder hits, and it's the reason most AI-built apps quietly die. Not because the app was bad. Because **building the app was never the hard part.** The hard part is the thing nobody hands you a tool for: getting a stranger to find it, understand it, and pay for it.

This playbook is about that part. And before we get into the system, I want to show you it working — on a real app, with real numbers, warts and all.

Meet the patient: a sports card app with a pulse and no heartbeat

The app is **Scout AI** — an AI-powered tool that gives sports card collectors and flippers exact buy prices, sell targets, and timing on rookie cards. Built on Lovable and Supabase, later migrated to Cloudflare and GitHub. A genuinely useful product for three kinds of people at once: serious collectors, active flippers, and beginners looking to turn card-flipping into a side hustle.

It had been live for months. Here's what the numbers looked like over a 90-day stretch:

- **584 visitors**
- **60% bounce rate**
- Average time on site: **4 minutes** (people who stayed were *interested*)
- Trials started: **0**
- Paying customers: **0**

Stop on those last two lines. This is the part that breaks people's spirit, because every instinct says the problem must be traffic. "If I just had more visitors..." But look again: 584 people showed up, and the ones who stuck around gave it four minutes. That's not a traffic problem. **Something was converting interested visitors into departing visitors,** and no amount of extra traffic would fix a leak like that — it would just pour more water through the same hole.

So the first move wasn't "get more traffic." It was **find the leak.**

The diagnosis

When you actually walk the path a visitor takes — click by click, the way a stranger experiences it, not the way the builder remembers it — two problems show up almost immediately.

Leak #1: The "free" promise that turned into a bill

Every call-to-action on the site said the same friendly thing: **"Get Started Free."** The hero button. The nav bar. The pricing section even showed a real, honest Free tier — "$0/mo, no credit card required."

But click any of those "Get Started Free" buttons and you landed on a signup page where the button said:

> ***Create Account — $9.99/mo***

...with fine print underneath: "*$9.99/month. Cancel anytime.*"

Read that as a stranger. You were promised free. You clicked *free*. And the very next screen asks for $9.99 a month. It doesn't matter that a free tier existed somewhere — at the exact moment of highest anxiety, the signup screen, the site appeared to bait-and-switch you. So you did what anyone does when they feel tricked: you closed the tab.

The cruel part? The app **wasn't** actually charging at signup. The flow just created a free account. The "$9.99/mo" was leftover label text that never got fixed. A lie the app was telling on its own behalf, for free.

Leak #2: The app couldn't decide what it was called

Follow the same two clicks and watch the name change under your feet:

- The domain: **scoutairookie.com**
- The homepage brand: **Scout AI**
- The signup page logo: **"RC"**, with text reading **"Create your RookieCard Guide account"**

Three names — Scout AI, RC, RookieCard Guide — in the span of two clicks. Each switch is a tiny "wait... am I still in the right place?" And a signup page is the worst possible place to make someone hesitate. Hesitation there doesn't convert to a question. It converts to a closed tab.

The fix

Neither leak needed a redesign. Neither needed more features, more traffic, or more money. They needed *honesty and consistency* — which took about twenty minutes.

Fix 1 — make the free promise true. The signup button changed from "Create Account — $9.99/mo" to **"Create My Free Account."** The fine print changed to **"Free to start. No credit card required. Upgrade only when you're ready."** Now the screen matches the button that got them there.

Fix 2 — one name, everywhere. Every "RC" and every "RookieCard Guide" across the signup, login, and profile pages became **Scout AI** — the name on the door. No more identity flip mid-journey.

That's it. No new code, no new features. Two changes that made the app stop sabotaging itself.

The lesson this whole playbook is built on

Here's what the Scout AI story teaches, and why it opens this kit:

The gap between "built" and "bought" is almost never where you think it is.

The builder's instinct is to assume the problem is *the product* (add features) or *the traffic* (get more visitors). Both are usually wrong. The real leak is almost always in the unglamorous space between a stranger arriving and a stranger deciding — the promise your page makes, the clarity of what you do, the friction at the moment of signup, the reason someone should care *today*.

That space is invisible to you, because you built the thing. You know it's free. You know what it's called. You know why it's good. The stranger knows none of that and gives you about five seconds to prove it before they're gone.

You built it. So — build then what?

That question is the whole reason this exists. This playbook is a system for seeing your app the way a stranger sees it — and closing every leak between *arrived* and *paid*, in the right order. We'll do it in four phases:

1. **Ideate** — make sure you're solving a problem a real person will pay to fix, and can say why in one sentence.
2. **Build** — shape the product and the page so the value is obvious in five seconds.
3. **Launch** — get it in front of the *right* strangers and give them a reason to act now.
4. **Optimize** — measure what actually matters (hint: not bounce rate), find the next leak, and fix it.

Scout AI is the case study we'll carry through all four. You'll watch the same app get taken apart and put back together, phase by phase, so every principle in here has a real example attached to it — not theory, not a hypothetical SaaS, an actual app that had 584 visitors and zero customers.

Let's find your leak.

Next: Phase 1 — Ideate. Before you fix a single button, you make sure you're building something a stranger will pay for — and that you can say who it's for and why it matters in one sentence. Most "nobody came" problems start here.

PART ONE

Fix the App

Before anyone can buy it, the thing itself has to make sense to a stranger.

Phase 1 — Ideate make sure a specific person will pay

Phase 2 — Build make the value obvious in five seconds

Phase 1 — Ideate

Make sure you're building something a stranger will pay to fix — and that you can say who it's for and why it matters in one sentence. Most "nobody came" problems are actually born here, long before the code.

Read this even if you already built your app

You might be thinking: *I already have an app. I'm past ideation. Skip to the part where you fix my signups.*

Don't skip. Here's why.

"Ideate" isn't only "pick an idea." It's the phase where you answer two questions that every later phase depends on:

1. **Is there a real person with a real problem who would pay to solve it?**
2. **Can I say who that person is and why my app matters — in one sentence?**

If you already built something, you're not choosing an idea anymore — you're *auditing* the one you bet on. And most builders who hit the "nobody came" wall discover the crack was here all along: they built something clever that no specific person was actually waiting for, or they built something genuinely useful but can't explain it to a stranger in one breath. Scout AI, our case study, is the second kind — a genuinely useful app with a positioning problem, not a product problem.

So whether you're staring at a blank screen or a live app with zero customers, Phase 1 does the same job. It just runs forward for one and backward for the other.

The trap: "cool" is not "wanted"

AI build tools made it dangerously easy to build things nobody asked for. When building took months, the sheer cost forced you to think hard about whether anyone wanted it. Now that building takes a weekend, that filter is gone. You can build the wrong thing faster than ever.

The AI tools have the same blind spot on the *idea* side. Ask an AI "give me a profitable app idea" and it will cheerfully hand you something that sounds plausible, uses the right buzzwords, and has absolutely no one behind it who'd pay. Plausible is not the same as wanted. **The whole skill of this phase is separating the two.**

That's why the ideation engine below has two halves: one to *generate* candidates, and a second — the part people skip — to *validate* them against reality before you write a line of code.

The engine, part 1: generate real candidates

This is the first of the kit's core prompts. Paste it into Claude (or your AI tool of choice) exactly as written. The specificity is deliberate — vague prompts get vague ideas.

> ***PROMPT 1 — Find a Profitable Niche***
> *Act as a Micro-SaaS strategist. Suggest 3 web app ideas with **real, current demand** and recurring-revenue potential. Avoid generic ideas — favor specific, underserved niches.*
> *For each idea, give me: - **Target audience** (be specific — not "small businesses," but "solo real-estate agents in their first two years") - **The problem it solves** (stated the way the customer would say it, in their words) - **The core MVP feature** (the one thing it must do) - **The unique advantage** (why me, not a spreadsheet or an existing tool) - **Monthly pricing** (a realistic number this audience would actually pay) - **Where these people already gather** (subreddits, Discords, Facebook groups, forums — name them)*
> *Then rank the three from most to least "painful problem," and tell me which to build first and why.*

That last addition — *where these people already gather* — isn't in most versions of this prompt, and it's the most important line. An idea you can't find an audience for is not an idea, it's a hobby. Forcing the AI to name the watering holes turns an abstract "target audience" into actual places you can go stand.

The engine, part 2: validate before you trust it

Now the half everyone skips. The AI just gave you three confident-sounding ideas. Confidence is free. Before you believe any of them, run this:

> ***PROMPT 1B — Pressure-test the idea***
> *I'm considering building [idea] for [target audience]. Play the skeptic, not the cheerleader.*
>
> - *What are the 3 strongest reasons this would **fail** to get paying customers?*
> - *Who **already** solves this problem for that audience (including free tools, spreadsheets, and "they just live with it")?*
> - *What would have to be true about this audience for them to actually pay monthly — and how could I check that this week, before building?*
> - *If you had to bet, would this audience pay, or just say "neat"?*

Then — and this is the part that separates builders who sell from builders who don't — **go verify it with humans, not the AI.** The prompt tells you where they gather. Go there. Read what they complain about. Post the problem (not the product) and see if anyone leans in. Five real replies from the target audience is worth more than fifty AI-generated reassurances.

You are looking for one specific signal: **people describing this problem in their own words, unprompted, with frustration.** That frustration is the fuel your whole business runs on. No frustration, no purchase — no matter how elegant the app.

The one-sentence test

Before you leave Phase 1, you must be able to fill in this sentence about your app without hedging:

> ***For [specific person], [app name] is the [category] that [specific outcome] — unlike [the alternative], it [key difference].***

If you can't fill that in cleanly, you don't have an idea problem — you have a *clarity* problem, and it will follow you into every later phase. The landing page you can't write, the ad you can't target, the cold email you can't send — they all trace back to a fuzzy version of this sentence.

Say it out loud. If a stranger wouldn't instantly get it, keep sharpening until they would.

Worked example: running Scout AI through Phase 1, backward

Scout AI was already built when we got to it, so we run Phase 1 as an audit. Here's what that looked like.

Who's the specific person? This was the first crack. "Sports card people" is not a person. Pushed for specifics, there are really three: the **serious collector** (wants to protect and grow a collection), the **active flipper** (wants to buy low and sell high, fast), and the **beginner** (heard you can make money flipping cards and wants a system that removes the guesswork). Three very different people with three very different pains — and the app was talking to all of them at once, which means it was landing squarely on none of them.

What's the problem, in their words? For the flipper and the beginner — the ones most likely to pay monthly — it's some version of: "*I don't know which cards are actually worth buying right now, or what to sell them for, and I'm scared of overpaying.*" That's a real, frightened, wallet-open problem. Good sign.

The unique advantage? *Unlike a spreadsheet or endless eBay sold-listings scrolling, Scout AI gives you exact buy prices, sell targets, and timing — powered by AI, no experience needed.* That's a legitimate edge, and notice it was already sitting right there in the app's own meta description. The idea was sound. The product was real. What was broken was everything *downstream* of a clear one-sentence promise — which is exactly why traffic came and nobody converted.

The one-sentence test, filled in:

> *For a beginner who wants to flip sports cards without overpaying, Scout AI is the AI scout that tells you exactly what to buy, what to sell it for, and when — unlike scrolling eBay for hours, it just hands you the answer.*

That sentence passes. Which tells us something important: Scout AI's "nobody came" problem was **never an idea problem.** It was a positioning-and-friction problem living in Phases 2 and 3 — the brand that flipped three times, the "free" button that asked for $9.99. Phase 1 didn't fix Scout AI. But it *proved the foundation was solid*, so we knew the fixes downstream were worth making instead of scrapping the whole thing.

That's the other gift of Phase 1 for an app you already built: it tells you whether to fix or to walk away.

Do This Now

Goal of this session: walk away with one sentence you'd stake money on, and proof a real human has this problem. Budget: about 90 minutes.

1. **Write your one-sentence test, badly, right now.** Fill in: "*For [specific person], [app] is the [category] that [outcome] — unlike [alternative], it [difference].*" Don't polish. Just get a first version on paper.
2. **Attack the word "everyone."** Wherever your sentence says a broad group ("collectors," "agents," "parents"), force it narrower until it names a person you could picture. "Beginners who want to flip cards without overpaying" beats "card people."
3. **Run Prompt 1B** (the skeptic prompt) on that sentence. Read the three failure reasons out loud. If any of them is obviously true, your sentence isn't done — fix it and rerun.
4. **Go find five real humans with this problem.** Use the "where they gather" list Prompt 1 gave you. Don't pitch. Just read threads and look for people describing your problem *in their own words, with frustration.* Screenshot three of them.
5. **Decide: fix or walk away.** If you found the frustration and your sentence survives the skeptic — this app is worth the next three phases. If you found silence, that's not failure, that's the cheapest lesson you'll ever get. Better to learn it now.

You'll know it worked when: you can say your one sentence to a stranger and they instantly know who it's for — and you have three screenshots of real people wanting it.

> ***Here's me doing it — Scout AI*** *My bad first sentence: "Scout AI helps sports card people make money." Too vague — "people," "make money." Narrowed: "For a beginner who wants to flip cards without overpaying, Scout AI tells you exactly what to buy, what to sell it for, and when — unlike scrolling eBay for hours, it just hands you the answer." Skeptic prompt flagged: "beginners may not trust an AI's prices." Real objection — noted for the landing page (add proof/sources). Where they gather: r/sportscards, r/baseballcards, card-flipping Discords, TikTok #sportscards. Action: pull three real "I overpaid / how do I know what it's worth" posts as proof. Verdict:* ***fix, don't scrap.*** *The idea is sound; the leaks are downstream.*

Phase 1 exit checklist

Don't move to Phase 2 until you can check every box:

- [ ] I can name my customer as a **specific person**, not a category.
- [ ] I can state their problem **in their own words**, and I've seen a real human express that frustration unprompted.
- [ ] I know **where these people gather** and could reach them this week if I had to.
- [ ] I can name **who/what already solves this** (including "a spreadsheet" and "they live with it") and why I'm better.
- [ ] I can fill in the **one-sentence test** without hedging.
- [ ] If I already built the app: I've decided this foundation is worth **fixing, not scrapping.**

If any box is empty, that's not a reason to feel behind — it's the highest-value work you can do right now. Every hour here saves ten in the phases ahead.

Next: Phase 2 — Build. Now that you know who it's for and why it matters, we make that value obvious in the first five seconds — in the product and on the page. This is where Scout AI's real leaks were hiding, and where we'll fix them on the page in front of you.

Phase 2 — Build

Make the value obvious in five seconds — in the product and on the page. This is where most "nobody came" apps are actually losing people, and it's where Scout AI's real leaks were hiding.

"Build" doesn't mean "write more code"

If you're reading this, you can already build. That's the strange gift of this moment — the code is the easy part now. So Phase 2 is not about building *more*. It's about building the **thin layer between your app and a stranger** so that the value you already created becomes visible to someone who has no idea what they're looking at.

Here's the mental model that matters: **you are cursed with knowledge.** You know what your app does, who it's for, why it's good, and that the free tier exists. A stranger knows none of that. They land on your page carrying zero context and about five seconds of patience. Phase 2 is the discipline of building for *that* person — the one who knows nothing and is one small confusion away from closing the tab.

Two surfaces decide whether they stay:

1. **The page** — can they understand what this is and why they'd want it, fast?
2. **The path** — from "interested" to "signed up," does anything make them hesitate, distrust, or stall?

Scout AI had a good product and lost people on *both*. Let's use it.

The five-second test

Before anything else, run this on your own landing page. Show it to someone who's never seen it — a friend, a spouse, a stranger in a Discord — for **five seconds**, then hide it and ask:

- What does this do?
- Who is it for?
- What would you do next?

If they can't answer all three, you don't have a traffic problem or a product problem. You have a **five-second problem**, and it's the highest-leverage thing you will fix in this entire playbook.

Your homepage's job is not to explain everything. It's to make one promise so clearly that the right person thinks "*that's for me*" and the wrong person leaves (which is fine — they were never going to pay). Clarity beats persuasion. A clear page that says one true thing outperforms a clever page that says five.

The anatomy of a page that passes

Top to bottom, a landing page that survives the five-second test usually has:

1. **Headline** — the one-sentence promise from Phase 1, in the customer's words. Outcome, not feature.
2. **Subhead** — who it's for + how it works, in one line. Removes the "is this for me?" doubt.
3. **Primary CTA** — one button, one action, repeated. Not five choices. One.
4. **The "how it works" in 3 steps** — because a stranger's real fear is "*will I understand how to use this?*"
5. **Proof** — a number, a screenshot of the actual product, a real quote. Something that says "this is real."
6. **The CTA again** — they scrolled, they're interested, don't make them scroll back up.

Notice what's *not* required: a feature list, a founder's story, an FAQ, a wall of text. Those can exist lower down. None of them belong in the first screen.

Scout AI on the operating table

Scout AI's homepage copy was genuinely good — this surprised us. The hero made a clear promise ("exact buy prices, sell targets, and timing for sports card flips"), the "how it works" was three simple steps, the trust strip was fine. On the five-second test, the *page* mostly passed.

So why 60% bounce and zero signups? Because the leak wasn't on the page a visitor read. It was in the **path** a visitor walked. And the path is exactly the surface builders forget to test, because they never walk it as a stranger — they're always already logged in.

Here's what walking Scout AI's path as a stranger revealed.

Leak #1 — the promise broke at the worst possible moment

Every call-to-action said **"Get Started Free."** The hero button. The nav. The pricing section even listed a real Free tier — $0, no card required. So a stranger clicks "Get Started Free," fully expecting *free*, and lands on the signup page — the single highest-anxiety moment in the entire journey, the moment they're deciding whether to trust you with an account.

And the signup button said:

> ***Create Account — $9.99/mo***

...with fine print: "*$9.99/month. Cancel anytime.*"

Read it cold. You were promised free three times. You clicked *free*. The very next screen asks for ten dollars a month. It does not matter that a free tier technically existed — at the decisive moment, the app appeared to bait-and-switch. So the stranger did the rational thing and left.

The bitter irony: **the app wasn't even charging.** The signup flow created a free account. No card was collected. The "$9.99/mo" was dead label text that never got updated — a lie the app was telling *against its own interest*, for free, to every visitor who made it that far.

The fix (about fifteen minutes): - Button: "Create Account — $9.99/mo" → **"Create My Free Account"** - Fine print: "$9.99/month. Cancel anytime." → **"Free to start. No credit card required. Upgrade only when you're ready."**

That's it. The screen now keeps the promise the button made to get them there.

The lesson for your app: walk your own signup path as a stranger, today. Not the homepage — the *signup screen* and every screen between "I'm interested" and "I have an account." Read every word as someone who was promised something on the previous click. The gap between what you promised and what the next screen says is where your money is leaking. It's almost always somewhere in there, and it's almost always invisible to you because you never walk it cold.

Leak #2 — the app changed its name mid-handshake

Follow the same two clicks and watch the brand shape-shift:

- The domain said **scoutairookie.com**
- The homepage said **Scout AI**
- The signup page logo said **"RC"** and the text read **"Create your RookieCard Guide account"**

Three identities — Scout AI, RC, RookieCard Guide — inside two clicks. Each switch is a micro-flinch: *wait, am I still in the right place? Did I click the wrong thing?* On a signup page, that flinch doesn't turn into a question you can answer. It turns into a closed tab.

This happens constantly to apps that evolve — the product gets renamed, but the old name survives in the corners the builder doesn't revisit: auth pages, emails, the profile screen. You stop seeing it because you know they're all the same thing. A stranger doesn't.

The fix: one name, everywhere. Every "RC" and "RookieCard Guide" across signup, login, and profile became **Scout AI** — the name on the front door. The handshake stays consistent from first click to account created.

The lesson for your app: audit every user-facing surface for name, logo, and voice consistency — especially the ones you never look at because you're always logged in. Signup, login, password reset, confirmation emails, the profile page. Pick one name and make every corner of the app agree with the domain.

Why these count as "Build"

You might notice neither Scout AI fix was a marketing tactic. No new traffic, no ad copy, no funnel. They were changes to the *product itself* — the button text, the brand consistency, the honesty of the signup screen. That's the point of Phase 2: **a huge share of "nobody's buying" is a build problem wearing a marketing costume.** Before you spend a dollar driving traffic, you make sure the thing the traffic lands on doesn't sabotage itself.

Driving traffic to an app with these leaks is pouring water through a bucket with holes. Phase 3 is where we fill the bucket. Phase 2 is where we patch the holes, so every visitor Phase 3 earns actually has a chance to convert.

Using the build prompts (Claude Prompts 2–5)

Phase 1 gave you the ideation prompt. Phase 2 is where the build prompts earn their keep — but with a crucial reframe: **use them to build the stranger-facing layer, not just the features.**

- **Prompt 2 (Interface):** Point it at your landing page and signup path, not just the app's internal screens. "*Design this so a new user understands the value within 30 seconds of landing*" — that line in the prompt is the five-second test in disguise.
- **Prompt 3 (Core Logic):** Fine as written for features — but the highest-value logic to get right is the *signup and trial flow*, because that's the path where Scout AI was bleeding.
- **Prompt 4 (Database):** When it designs your schema, make sure "captured but not yet signed-up" emails have somewhere to live (more on this in Phase 3 — the 60% who won't sign up today are not worth *nothing*).
- **Prompt 5 (Interactive Features):** Ends with "what to test before launch." Add one item to that test list it won't think of: *walk the entire path from landing page to created account as a brand-new stranger, reading every word cold.*

The prompts are in the appendix, formatted to copy and paste. But a prompt is only as good as what you point it at. Point them at the stranger's experience, not just the feature set.

Do This Now

Goal of this session: find and fix every place your app breaks its own promise on the way to signup. Budget: about 2 hours.

1. **Open your app in a private/incognito window.** This is non-negotiable — you must arrive as a stranger, logged out, with no memory of how it works.
2. **Screenshot every screen from landing to account-created.** Homepage, every button you click, the signup screen, the confirmation. Lay the screenshots side by side.
3. **Read every word as someone who was promised something on the previous click.** On each screen ask: does this keep the promise the last screen made? The classic break: "Get Started Free" → a screen that mentions a price.

4. **Run the five-second test on your homepage.** Show it to one real person for five seconds, hide it, ask: what is this, who's it for, what would you do next? Write down what they actually say.
5. **Fix the leaks in priority order:** (a) any broken promise (free-that-isn't, etc.), (b) any place the brand name/logo changes, (c) any moment of "wait, what do I do here?" Ship the fixes. If you build in Lovable/Cursor, use Prompt 2 and Prompt 5 to make the changes cleanly.
6. **Walk the path again, still incognito, to confirm.** The promise holds start to finish.

You'll know it worked when: a stranger can go from your homepage to a created account without a single "wait, huh?" moment — and every "free" you promised is still true at signup.

> ***Here's me doing it — Scout AI*** *Walked scoutairookie.com logged out. Found it: "Get Started Free" everywhere → signup button said* ***"Create Account — $9.99/mo."*** *Promised free, asked for money. (It wasn't even charging — dead label text.) Also found: brand flipped Scout AI → "RC" → "RookieCard Guide" across two clicks. Fixed: button →* ***"Create My Free Account"****, free-to-start subtext; unified every page to* ***Scout AI****. Shipped. Confirmed live. Still on my list: add the price-trust proof the Phase 1 skeptic flagged; add a no-account "see this week's free pick" so logged-out visitors get value before committing.*

Phase 2 exit checklist

Don't move to Phase 3 (driving traffic) until:

- [ ] My landing page passes the **five-second test** with a real person who's never seen it.
- [ ] My headline is the **outcome** in the customer's words, not a feature.
- [ ] There is **one** primary CTA, repeated — not a menu of choices.
- [ ] I have walked the **entire path** from landing page to created account as a stranger, reading every word cold.
- [ ] Every promise on the page is **still true** on the next screen (no "free" that turns into a bill).

- [ ] **One name, one logo, one voice** across homepage, signup, login, profile, and emails.
- [ ] There's at least one piece of **real proof** on the page — a screenshot of the actual product, a number, or a genuine quote.

Every unchecked box is a hole in the bucket. Patch them before you spend a cent on traffic — because in Phase 3, we start pouring.

Next: Phase 3 — Launch. The bucket's patched. Now we get the right strangers to it — where they already gather, with a reason to act today — and give them a low-friction way to raise their hand even if they're not ready to sign up yet. This is where the 60% who bounce stop being a total loss.

PART TWO

Get It Seen

The app is sound. Now the real work — getting the right strangers in front of it.

Phase 3 — Launch

The bucket's patched. Now you get the right strangers to it. This is the phase where most builders work the hardest and get the least — so we're going to look at exactly why, using a real account with 460 posts and nothing to show for it.

The most honest section in this book

I need to tell you something before we start, because it's the whole reason this section is trustworthy.

The case study app in this book — Scout AI — was promoted. Hard. Not "I posted a few times and gave up." **460 posts** across TikTok, Instagram, and Facebook. Real hooks, real effort, months of it. And it produced **zero customers.**

Most launch advice is written by people selling you the idea that you just haven't *tried* enough — post more, be consistent, hustle harder. This section is written from the other side of that advice: someone who *did* post 460 times and still got nothing, then went back through the analytics screenshot by screenshot to find out why. What we found wasn't a lack of effort. It was five specific, mechanical mistakes — and every one of them is the kind of thing you cannot see from inside your own launch.

If you've been posting and hearing crickets, you are probably making some of these five right now. Let's find them.

First, the myth we're killing: "your hooks are weak"

When a launch fails, the first thing everyone attacks is the hooks — the opening line of each post. "Make them punchier. Add more curiosity. Study viral formats."

So let's look at Scout AI's actual hooks:

- "*Rookie card market is moving fast — here's what's trending.*"
- "*How to spot the next breakout rookie card.*"
- "*3 rookie card mistakes collectors make (and how to avoid them).*"
- "*My card collection is worth $28,865. Most collectors have no idea...*"

- "*This $5 card is now worth $50.*"
- "*Stop guessing which rookie cards to buy — let AI do the research.*"

Those are *good hooks.* Curiosity gaps, specific numbers, mistake-avoidance, a contrarian angle. If you paid a copywriter, these are what you'd get back. And they still produced nothing.

Sit with that, because it's the most important lesson in this phase: **good hooks are necessary but nowhere near sufficient.** A perfect hook shown to the wrong room, or no room, or a room that can't hear you, converts exactly as well as a terrible one. If your launch is failing, the hook is the *last* place to look, not the first. Here's where to actually look.

Leak #1: You're posting into a vacuum

Scout AI's 460 posts went out from an account with no established audience of *card people.* No following of collectors. No community that showed up for card content. Just post after post launched into a void.

Here's what almost no one tells first-time builders: **on every modern platform, reach is earned from an existing audience, not granted to a good post.** The algorithm shows your post to a small test batch first — mostly your existing followers plus a sliver of strangers. If that batch engages, it expands. If you have no relevant followers, the test batch is tiny and random, the engagement is weak, and the post dies in an hour. Volume doesn't fix this. 460 posts into a vacuum is just 460 small deaths.

The fix: before you expect a launch to convert, you build the room. Pick one platform, one specific person (your Phase 1 customer), and spend real effort *becoming a place that person follows* — not just broadcasting at them. Follow, comment, reply, show up in the niche. Fifty engaged followers who are your exact customer are worth more than 5,000 random impressions. Audience first, launch second.

Leak #2: The account didn't match the app

This one is subtle and it was quietly killing Scout AI the whole time.

The card content wasn't even going out from a card account. It was being posted from an account branded around **faceless digital-business education** — a "make money online" style brand. Think about who follows an account like that: people who want side-hustle and online-business content. And they were being served… rookie card flipping.

That's an audience mismatch *at the account level* — upstream of the hook, upstream of the algorithm. Even the few people who saw it were the wrong people. A brilliant card hook shown to a room of "make money online" followers is a wasted brilliant hook. The account itself was pre-filtering for exactly the wrong audience.

The fix: one app, one correctly-branded account, per platform. If you sell a card tool, the account is a *card* account — its name, its bio, its whole vibe signals "this is for card people," so the followers it attracts are the people who might actually buy. Don't promote your app from whatever account you happen to have. The account's brand *is* audience targeting, for free, before you post a word.

Leak #3: You split your effort until none of it compounded

Scout AI wasn't promoted from one account. It was scattered across three or four different handles — one TikTok, two different Instagram accounts, a Facebook page — none of which was the same identity.

Even if each of those had been correctly branded, this splits your momentum into pieces too small to matter. Audiences compound: the 100th follower makes the 101st easier, engagement builds on engagement, the algorithm starts trusting an account that consistently holds attention. Spread across four accounts, no single one ever reaches the threshold where that compounding kicks in. Four accounts with 25 followers each is not "100 followers." It's four accounts that all look dead to the algorithm.

The fix: consolidate. One primary account per platform per app, and pour everything into it. Depth beats spread. A single account that becomes *the* place for your niche is worth infinitely more than four half-built ones.

Leak #4: There was no engagement loop, so nothing ever caught fire

Across all those Scout AI posts: **comments were essentially zero.** Instagram over 90 days showed 2 total interactions. Facebook posts sat at 0 engagement.

Remember Leak #1 — the algorithm expands what gets early engagement. A view is passive; the platform barely counts it. A *comment* or a *save* or a *share* is the signal that tells the algorithm "push this to more people." Scout AI's posts gave viewers no reason to do any of those. They stated something and ended. Nothing asked, nothing to answer, nothing to save for later. So the algorithm's verdict was: nobody's reacting, stop showing it. Every post capped out at its tiny test batch and stopped.

The fix: engineer one engagement trigger into every post. End on a real question ("which rookie are you holding?"). Take a stance people want to argue with. Use the "comment [word] and I'll send you the list" mechanic to manufacture comments *and* capture leads at once. You're not begging for engagement — you're giving the algorithm the signal it needs to do its job for you.

Leak #5: The small mechanical stuff that quietly bled reach

Two more, quick but real:

Some posts were never public. Going back through the TikTok data, a batch of posts were set to **"Only me"** — private. Not a strategy choice; a settings glitch. Real work, done, uploaded, and invisible to the entire world. Before you conclude a launch "isn't working," verify the posts actually went out public. It sounds too dumb to matter. It cost real reach.

The content didn't connect to the app. Here's the quiet gut-punch: Scout AI's own website analytics showed its traffic was almost entirely *direct* — the social platforms barely registered as referrers. So even the views that did happen weren't clicking through to the app. The content lived in one world and the product lived in another, with no bridge between them. Every post needs an obvious, friction-free path to the thing you're selling — and you measure *click-throughs to the app,* not just views on the post.

Now — organic vs. paid, and the thing nobody admits

Everything above is about *organic* reach: free posting, hoping the right person finds you. There's a hard truth buried in Scout AI's 460 posts that only becomes visible once you see all five leaks together:

With organic-only promotion, you never control who sees your content. The algorithm decides. So when a launch fails organically, you genuinely cannot tell whether the problem was the message or the audience — because you never held the audience still. Scout AI's failure was tangled: bad audience (wrong account), no audience (no following), and no amplification (no engagement), all at once. Impossible to isolate.

This is what paid ads are actually *for* at this stage — and it's not what most people think. A small paid test (even $5–10/day) isn't about buying customers. It's about buying **control of the audience variable.** You choose exactly who sees your best piece — the specific person from Phase 1 — and now if it still doesn't convert, you've learned something real, because you controlled who was in the room. Paid is the clean experiment organic can never be.

The sequence that matters:

1. **Build a small, correct, engaged organic audience first** (Leaks 1–4). This proves your content can hold *someone.*
2. **Fix the mechanics** (Leak 5) so nothing leaks.
3. **Then run a small paid test** to a controlled audience, to isolate whether your offer converts when the right person definitely sees it.

Do it in that order. Running paid traffic before the bucket is patched and the content works just means paying to fill a leaky bucket faster.

The Audience Playbook

How to actually find the right people, say the right thing, and put it in front of them on purpose

The uncomfortable truth about "find your audience"

Every launch guide says it. Almost none of them tell you *how*. So here's the honest version:

Your audience already exists, already has the problem, and is already gathered somewhere talking about it. You don't create an audience. You go find the room they're already in, become a useful member of it, and eventually mention that you built something. That's it. That's the whole game.

The reason most builders fail here isn't laziness — it's that they skip straight to broadcasting. They post *at* a room they've never entered. Below is the sequence that actually works, in order, with the exact work each step takes.

Step 1 — Build your audience profile (30 minutes)

Before you can find them, you have to be able to describe them precisely enough to recognize them. Vague audiences produce vague marketing that reaches nobody.

Fill this in. Every line needs a *specific* answer, not a category:

Field	Your answer
Who they are (a person, not a group)	
The problem in their words (how *they'd* say it)	
What they're doing about it now (the current bad solution)	
Where they already gather (name 5 actual places)	
What they're afraid of (the fear behind the problem)	
What they'd search for (exact words they'd type)	
Who they already follow (5 accounts/creators in this space)	
What they'd never care about (helps you cut noise)	

Two lines matter most. **"What they'd search for"** becomes your content topics and your SEO. **"Who they already follow"** becomes your paid targeting *and* your map of where to go be useful.

> ***Here's me doing it — Scout AI Who:*** *a beginner/hobbyist flipper, 30–60, who buys a few cards a month and doesn't want to get burned.* ***Problem in their words:*** *"How do I know what this is actually worth before I buy it?"* ***Doing now:*** *scrolling eBay sold listings for an hour, guessing, asking in Facebook groups.* ***Where they gather:*** *r/sportscards, r/baseballcards, card-flipping Discords, TikTok #sportscards / #cardtok, Facebook card-buying groups.* ***Afraid of:*** *overpaying and looking dumb; missing a card that pops later.* ***Would search:*** *"is this rookie card worth it," "how to spot a fake card," "what rookie cards to buy 2026."* ***Already follow:*** *the big card-breaker/flipper accounts on TikTok and YouTube.* ***Never care about:*** *SaaS features, AI architecture, my tech stack.*

Step 2 — Go *into* the rooms (10 days, 20 min/day)

You listed five places. Now you show up in them — not as a marketer, as a member. This is the step everyone skips and it's the one that makes everything after it work.

The daily 20 minutes, split three ways:

1. **Listen (5 min).** Read the top threads. Write down the exact phrases people use to describe the problem. This is free copywriting — their words beat yours every time.
2. **Help (10 min).** Answer 2–3 questions with genuinely useful answers and *no link*. Not "check out my app." Just help. You're building the right to be heard later.
3. **Note (5 min).** Log every recurring question. After 10 days you'll have a list of the 10 questions your audience asks constantly — that list *is* your content calendar. You will never wonder what to post again.

Why no links for 10 days: every community can smell a drive-by promoter, and one bad first impression closes the room. Ten days of being useful buys you a permanent welcome. It's the cheapest marketing investment you'll ever make.

> ***Here's me doing it — Scout AI*** *20 min/day for 10 days in r/sportscards, card Discords, and #cardtok comments. Answer "what's this worth?" questions honestly, no Scout AI link. Log every repeated question — that becomes my post list for the next month.*

Step 3 — Turn their questions into content (the only content plan you need)

You now have a list of real questions from real people. That's your content engine. Every post is one question, answered.

The four post types that actually attract buyers:

1. **The direct answer** — take one logged question and answer it fully and honestly. ("How do you know if a rookie card is overpriced?") This attracts exactly the person with that problem.
2. **The mistake** — name a specific expensive mistake and how to avoid it. Fear of loss pulls harder than promise of gain.
3. **The receipt** — show a real number, a real result, a real screenshot. ("Here's what this $5 card sold for.") Proof beats claims.
4. **The contrarian take** — a defensible opinion the niche argues about. This is your comment engine; arguments are engagement.

The rule that makes it work: every post answers the question *first*, completely, for free. The app gets mentioned as the shortcut, not the gate. "Here's how to do it manually — takes about an hour. I built a tool that does it in a minute if you'd rather." That framing converts; "buy my app" doesn't.

Cadence that's sustainable: 4–5 posts a week on ONE platform beats 20 posts across four. Pick the platform where your audience profile says they gather most, and go deep.

> ***Here's me doing it — Scout AI*** *Every post = one real logged question from the card communities, answered fully. Rotate the four types. All of it through* ***CardVaultCarl*** *(correctly card-branded), TikTok primary + Instagram secondary. 4–5/week. Always end with the engagement trigger ("what rookie are you holding?"), always verify PUBLIC.*

Step 4 — Set up paid targeting properly (your first controlled test)

Only after your organic content shows it can hold a small correct audience. Then paid stops being a gamble and becomes an experiment.

What you're actually buying: control of *who* sees your message. That's it. You're not buying customers yet — you're buying an answer to "does this convert when the right person definitely sees it?"

The setup, step by step:

1. **Pick your winner.** Use your best-performing organic post — the one that got real comments. Never test with untested creative; you'll confound two variables.
2. **Budget: $10/day for 5 days.** ~$50 total. That's enough for a signal, small enough that a wrong answer doesn't hurt. Do not scale before you have a signal.
3. **Build the audience from your profile.** Use the "who they already follow" line — target interests around those creators/brands, plus the obvious niche interests. Keep the audience *narrow* (this is the whole point). If the platform says your audience is 8 million people, you've targeted too broadly.
4. **One objective: traffic or conversions to your patched landing page.** Not "engagement," not "views." You want clicks that land on the app.
5. **Send them to the patched page** — the one where "free" is actually free. Never run ads to an unpatched funnel; you'll pay to prove your bucket leaks.

6. **Track one number: signup rate.** Visitors from the ad ÷ signups. That's the answer you paid for.

How to read the result:

- **Clicks are cheap but nobody signs up** → your *offer or landing page* is the problem, not your audience. (You've now isolated it — that's worth $50.)
- **Nobody even clicks** → your creative/hook doesn't work for this audience, or the targeting missed. Try one more audience before blaming the creative.
- **Clicks and signups both happen** → you found product-audience fit. Now you scale carefully and move to Phase 4.

> ***Here's me doing it — Scout AI (first paid test ever)*** *Winner post: best-performing rookie-card hook from CardVaultCarl. Budget: $10/day × 5 days = $50. Audience: narrow — sports-card-collector interests + the big card creators my profile says they follow. Keep it tight, not millions. Destination: patched Scout AI signup (the honest free one). The one number: signups ÷ ad visitors. First time in this whole journey I'll actually control who sees it — which finally makes the result mean something.*

Step 5 — Know when the right audience has arrived

You're looking for **quality signals, not volume signals.** Here's how to tell the difference:

Right audience showing up: - Comments that ask *specific* questions about the problem ("what about graded cards?") - Saves and shares (people keeping it for later = real value) - Profile visits and follows *after* a post (they want more of this) - Clicks through to the app, not just views - DMs asking if it does X

Wrong audience (or none): - Views with no comments (the Scout AI pattern — 460 posts, ~0 comments) - Generic praise with no specifics ("nice!" / bot-ish comments) - Follows from accounts unrelated to your niche - Traffic that bounces in seconds

The honest benchmark: 50 followers who are genuinely your customer and comment on your posts is worth more than 5,000 random followers. If you have to choose between growing the number and tightening the fit — tighten the fit, every time. A small correct audience converts. A large wrong one never will, and it'll cost you months finding that out.

The order matters more than the effort

If you take one thing from this section: **profile → enter the rooms → answer their questions → then and only then, pay to target.** Every step earns the right to the next one.

Scout AI's 460 posts were step 3 without steps 1 and 2 — broadcasting content into rooms nobody had entered, for an audience never defined, from an account that signalled the wrong thing. All that work, executed out of order, produced nothing.

Same effort, right order, is a completely different outcome. That's not a motivational line — it's the mechanical difference between a post the algorithm buries and a post that finds the person who's been looking for exactly what you built.

The Content Engine

How to actually produce and distribute content without burning out or hiring anyone

This is the second half of Phase 3. The Audience Playbook told you who to reach and where. This tells you what to make, how to make it at volume without losing your life to it, and where each piece goes. It's the part that turns "post consistently" from a guilt trip into a system.

The problem with "just post consistently"

Every guide says post consistently. None of them tell you how to generate enough content to be consistent without it eating forty hours a week. So you post hard for two weeks, burn out, go quiet, and the algorithm forgets you exist.

The fix isn't discipline. It's a **system that turns one idea into many pieces of content across many platforms** — mostly faceless, mostly automated, so the machine keeps running whether you feel inspired that day or not.

Here's the whole system, in the order you'd build it.

Part 1 — The content-format menu

You don't need to do all of these. You need to pick the two or three that fit your app and your energy, and run them consistently. Here's what each one is good for.

Faceless carousels (the workhorse)

Static multi-slide posts — a hook slide, a few value slides, a call-to-action slide. Text over clean branded backgrounds. No camera, no voice, no face.

Why it's the workhorse: it's the highest value-per-effort format there is. Carousels get saved and shared (the two signals that make the algorithm push you), they teach genuinely useful things, and — critically — they can be **produced on an assembly line.** One template, swap the text, export, done. This is the format to automate first.

Best for: teaching, listicles ("5 rookie cards to watch"), mistakes-to-avoid, step-by-steps, before/afters. Instagram and LinkedIn especially; TikTok now too.

Short-form video, faceless (the reach engine)

15–40 second vertical videos with text on screen, b-roll or screen-recording underneath, trending audio, no face. The app's own screen recording is gold here — show it doing the thing.

Why it matters: short video is still where raw reach lives. The algorithm hands new accounts more distribution on video than on anything else. Faceless keeps it repeatable.

Best for: the hook-first pieces — "this $5 card is now worth $50," a fast screen-record of your app solving the problem, quick tips. TikTok, Reels, Shorts.

Talking-head / mascot video (the trust builder)

A face (yours) or a consistent mascot character talking to camera. Higher trust, higher connection — people buy from people (or from characters they recognize).

Why it's worth the extra effort: faceless scales reach but is weak on trust. A recurring face or mascot is what converts a viewer into a follower into a buyer, because they feel like they know you. You don't need many — one a week does the relationship work your faceless content can't.

Best for: the story, the honest take, the "here's what I learned" piece, answering a real question. If you won't go on camera, a consistent AI-generated or illustrated mascot with a voiceover does the same job — recognizability is what matters, not your actual face.

The pick-two rule

Don't run all three from day one — you'll drown. **Run faceless carousels + short video first** (reach + saves), add the talking-head/mascot layer once the machine is humming. One trust-building piece a week is plenty to start.

Part 2 — The production system (make it an assembly line)

This is the part that makes consistency possible: you stop making content one piece at a time and start running a pipeline. The model below is a real one — adapt the tools to whatever you already use.

Step 1: Batch the ideas, not the posts

Once a month, sit down with your logged audience questions (from the Audience Playbook) and turn them into a list of 20–30 content ideas in one sitting. Ideas are cheap and fast in bulk; the mistake is generating them one day at a time. One idea session feeds a month.

Step 2: Templatize the format

Build **one** reusable design template per format. A carousel template with your fonts, colors, and slide structure locked in — so producing a new one means swapping text, not designing from scratch. This is the single biggest time-saver in the whole system. A good template turns a 45-minute design job into a 5-minute fill-in.

Step 3: Automate the assembly

This is where it goes from "batch" to "engine." A design tool like Canva (with its bulk-create feature) can take a spreadsheet of your hooks and text and **generate a whole month of carousels in one pass** — one row per post, auto-filled into your template. Feed it 20 rows of text, get 20 finished designs. Pair that with an AI step that writes the slide text from your idea list, and you've got a near-hands-free carousel factory: idea list → AI writes the slides → bulk-create fills the template → export the batch.

For video, the same logic applies: a script template, an AI voice or a mascot generator, and a repeatable edit style mean you're assembling from parts, not starting blank each time.

The principle underneath all of it: never make one thing at a time. Set up the template once, then run batches through it. The setup is the work; after that it's throughput.

Step 4: Schedule it and walk away

Load the batch into a scheduler and let it drip out on cadence. The whole month goes out whether you're busy, traveling, or uninspired. This is what "consistent" actually looks like mechanically — not willpower, a queue.

Part 3 — One idea, many pieces (the repurposing multiplier)

Here's the leverage that makes the whole thing sustainable: **one good idea is not one post. It's a week of content across every platform.**

Take a single logged question — say, "how do I know if a rookie card is overpriced?" One idea becomes:

- A **faceless carousel** walking through the 4 checks (Instagram, LinkedIn)
- A **short video** of the app doing it in ten seconds (TikTok, Reels, Shorts)
- A **talking-head/mascot** clip telling the story of a time you overpaid (all video platforms)
- A **text post** with the single best tip (X, Threads, a community reply)
- A line in your **email** to the list

That's five to seven pieces from one idea, each native to where it lives. Do this and your "20 ideas a month" becomes 100+ pieces of content — without 100 units of work, because they all trace back to 20 thoughts. The repurposing *is* the productivity.

The rule: never let a good idea be a single post. Every idea gets run through every format it fits.

Part 4 — Where to post, and how much

You don't post everywhere equally. You anchor on one platform and repurpose outward. Here's the honest per-platform read for an app builder.

Platform	Best format	Why it's here	Realistic cadence
TikTok	Short faceless video	Highest organic reach for a new account; best place to be discovered cold	1/day if video is your anchor
Instagram	Carousels + Reels	Saves and shares compound; carousels punch above their weight	4–5/week
YouTube Shorts	Repurposed short video	Massive reach, low competition, evergreen discovery	Repurpose your TikToks, 3–5/week
LinkedIn	Carousels + text	Underrated for B2B/tool builders; less noise, real reach	3/week
X / Threads	Text + short video	Fast feedback, good for the contrarian takes and building in public	Daily is easy since text is cheap
Pinterest	Carousels/pins	Sleeper for evergreen search traffic to a landing page	Repurpose carousels, low effort

The anchor-and-echo model: pick ONE platform as your anchor (where your audience profile says they gather most, usually TikTok or Instagram for a consumer app, LinkedIn for a B2B tool). Put your best daily effort there. Every other platform gets the *repurposed echo* of that anchor content — near-zero extra work, extra reach. Don't try to be excellent on six platforms. Be excellent on one and present on the rest.

On "how much": consistency beats volume beats perfection. A sustainable 4–5 posts a week you can keep up for six months crushes 30 posts in two weeks followed by silence. The algorithm rewards the streak, not the sprint. Set the cadence at the level you can *automate and maintain*, not the level you can white-knuckle for a fortnight.

Part 5 — The marketing levers beyond social

Social content is the engine, but it's not the whole vehicle. These are the other levers, roughly in order of value-per-effort for a solo app builder.

Build in public

Share the actual journey — the numbers, the fixes, the failures. "584 visitors, zero signups, here's what I found" is more compelling than any polished ad, because it's real and rare. It builds an audience that's rooting for you, and it turns your development work into content for free. This whole playbook is proof the approach works.

Email — the one channel you own

Every follower on every platform is rented; the algorithm can cut your reach tomorrow. An email list is the only audience you own outright. Capture emails from day one (the "comment X and I'll send you the guide" mechanic, a free lead magnet, a waitlist), and send a simple weekly note. A small owned list out-converts a large rented following every time.

SEO / a simple blog

The app's site should answer the exact things your audience searches. One honest post per week targeting a real search phrase ("how to spot a fake rookie card") compounds quietly for years, pulling in buyer-intent traffic while you sleep. Slow to start, but it's an asset that doesn't stop working.

Communities (give first)

The Reddit threads, Discords, and Facebook groups from your Audience Playbook aren't just for research — they're distribution. Be genuinely useful there for weeks, and when you mention your app it lands as a recommendation from a known member, not a spam drop. Highest-trust traffic there is.

Launch platforms

Product Hunt, relevant subreddits, indie-maker directories, and AI-tool listing sites give you a discovery spike and backlinks. Not a long-term engine, but a good punctuation mark for a launch or a big update. Line them up and fire them together for one concentrated push.

Collaborations

One shoutout from an established creator in your niche beats a month of your own posting, because you borrow their built audience. Find the mid-size accounts your people already follow (from your audience profile), and offer them genuine value — a free lifetime account, an affiliate cut, a co-made piece of content. Their trust transfers to you in one post.

The Content Engine, in one picture

Put it together and the machine looks like this:

One idea session (20–30 ideas from real questions) → **AI drafts the pieces** → **templates + bulk-create assemble them** → **scheduler drips them out on your anchor platform** → **repurposed echoes hit every other platform** → **email captures the ones who aren't ready to buy** → **communities and collabs amplify the best pieces** → and the whole thing feeds itself, because the comments and questions you get back become next month's idea session.

You built the app. This is the engine that finally gets it seen — running mostly without you, so you can keep building.

Content Engine → Do This Now

Goal: stand up a repeatable content machine this week — not a viral post, a *system*. Budget: one afternoon to set up, then a few hours a month.

1. **Run one idea session.** Open your logged audience questions and write 20 content ideas in one sitting. Don't design anything yet — just the ideas.
2. **Pick your two formats.** Faceless carousels + short video is the default starting pair. Commit to those two; ignore the rest for now.
3. **Build one template per format.** One carousel template with your brand locked in; one video structure (hook → value → CTA) you can refill. This is the afternoon's real work.
4. **Batch-produce week one.** Turn 5 of your ideas into 5 carousels using the template (or bulk-create if your tool supports it). You now have a week of content in an hour.
5. **Pick your anchor platform** (where your audience profile says they gather) and **schedule the batch** there. Repurpose the same pieces to one echo platform.
6. **Set the cadence you can maintain** — 4–5/week beats a doomed daily sprint. Put it in the scheduler so it runs without you.

You'll know it worked when: a week of content goes out without you touching it daily, and producing next week's batch takes an hour, not a weekend.

> ***Here's me doing it — Scout AI*** *Idea session: pull the repeated "what's it worth / how do I not overpay" questions from the card communities → 20 ideas. Formats: faceless carousels (my automated Canva pipeline) + short screen-record videos of Scout AI pricing a card. Add one mascot/talking-head a week later. Template: my existing card-carousel template, brand-locked. Bulk-create fills a month from a spreadsheet of hooks. Anchor: CardVaultCarl on TikTok; echo the same pieces to Instagram + YouTube Shorts. Cadence: 5/week scheduled, dripping out whether I'm heads-down building or not. Email: every carousel ends with "comment SCOUT and I'll send you the free rookie-watch list" — captures the not-ready-to-buy into a list I own.*

The Paid Ads Playbook

Your first $200, spent to learn something instead of to disappear

The last piece of Phase 3. Ads are not how you rescue a launch that isn't working — they're how you scale one that is, and how you buy a clean answer when organic can't give you one. This section is about spending small, learning fast, and not lighting money on fire the way most first-timers do.

Before you spend a dollar: the four prerequisites

Ads amplify whatever is already there. If what's there is a leak, ads just make you lose money faster. Do not run traffic until all four of these are true:

1. **Your funnel is patched.** Phase 2 done. The signup path works, the "free" promise holds, the brand is consistent. Paying to send strangers into a broken funnel is the single most expensive mistake in this book.
2. **You have proven creative.** At least one organic post that got real engagement — comments, saves, shares. You are going to *promote that*, not invent something new. Testing unproven creative on paid traffic means you can't tell whether the ad failed or the targeting did.
3. **You can track a conversion.** You need to see signups attributed to the ad, not just clicks. That means a pixel installed and a signup event firing (more on this below). Without it you're flying blind and every number you look at will lie to you.

4. **You know your specific person.** The audience profile from the Audience Playbook. "Everyone who likes sports" is not targeting; it's donating.

If any of those is missing, fix it first. The ads will still be there next week.

What ads are actually for at your stage

Reframe this now, because it saves you from the most common disappointment: **your first ad campaign is not a revenue campaign. It's an experiment.**

Organic reach never lets you control who sees your message, so when nothing converts you can't tell whether the problem is the offer, the message, or the audience. Paid ads let you hold the audience still. You are buying a **clean answer to one question**: *does this convert when the right person definitely sees it?*

Budget accordingly. A first test that returns "no, and here's which part is broken" for $150 is a *successful* test. That's cheaper than another six months of guessing.

Which platform, and why

Pick **one**. Running two platforms on a small budget means neither gets enough data to teach you anything.

Meta (Facebook + Instagram) — the default for most consumer apps and the best place for a beginner to start. Deepest targeting, cheapest reliable clicks, best learning tools, and it self-optimizes well once your pixel has data. If your customer is over 30, this is almost certainly your platform.

TikTok Ads — best if your organic winner is video and your audience skews younger. Cheap reach, high volume, but creative burns out fast and you'll need fresh video regularly. Strong if you're already producing short video from the Content Engine.

Google Search Ads — different animal entirely, and often the highest-intent traffic you can buy. Instead of interrupting people, you appear when someone *searches the problem*. If your audience profile's "what they'd search for" line has real search phrases in it ("how to price a rookie card"), this is worth testing — the traffic is warmer and converts better, though clicks cost more.

Reddit Ads — underrated for niche tools. You can target *specific subreddits* where your people already gather, which maps perfectly onto your Audience Playbook research.

Cheap, less competitive, but lower volume and the audience is ad-skeptical, so the creative must be honest and non-salesy.

YouTube Ads — best once you have a strong talking-head/demo video and a bigger budget. Skip for a first test.

The beginner's answer: Meta if your customer is a consumer, Google Search if they actively search for the problem, Reddit if they live in a specific community. One of those three.

Campaign setup: the settings that actually matter

Most first-time ad money is wasted on three settings. Get these right and you're ahead of most people spending ten times more.

1. Choose the right objective — this is the big one

Ad platforms ask what you want, then optimize hard for it. Choose **Conversions** (or "Sales" / "Leads" depending on the platform's wording), targeting your signup event.

Do not choose "Engagement," "Traffic," "Video Views," or "Reach." These are where beginners lose money, because the platform will faithfully deliver exactly what you asked for: cheap clicks from people who click things and never buy. Traffic objectives get you the cheapest, least valuable humans on the internet. If your pixel doesn't have enough conversion data yet, "Traffic" is a temporary fallback — but know it's a compromise, and switch as soon as you can.

2. Install the pixel and define the conversion event

Put the platform's pixel/tag on your site, and fire a specific event on successful signup. This does two things: it lets you measure the only number that matters (cost per signup), and it teaches the platform's algorithm what a good customer looks like so it can go find more of them. **The algorithm is your best targeter — but only once you feed it conversion data.**

3. Set the budget and then leave it alone

- **$10–20/day** for a first test. Enough for signal, small enough to be tuition.
- **Run it 7–14 days minimum**, and here's the part people get wrong: **do not touch it for the first 3–5 days.** Every platform has a learning phase where it's exploring

audiences and its numbers look terrible. Editing the campaign resets that phase and burns your budget starting over. Set it, walk away, judge it after the learning period.

- **Total first commitment: about $150–250.** That's the price of a real answer.

Targeting: narrow, then let the machine widen

Two schools of thought, and the honest answer depends on your data:

Start narrow when you have no pixel data. Build the audience straight from your audience profile — the interests, and especially the **"who they already follow"** line. Stack 3–8 related interests. If the platform tells you your audience is 5 million+ people, you're too broad for a small budget; you want something in the low hundreds of thousands to a couple million.

Go broader once the pixel has conversions. Modern ad algorithms out-target humans once they've seen 30–50 conversions. At that point, loosening the targeting and letting the machine find your people usually beats your manual guesses.

Lookalikes come later. Once you have a real customer list or 100+ signups, a 1% lookalike audience is typically the best-performing audience you'll ever build. You can't do this on day one — it's a reward for having data.

Exclude your existing traffic so you're not paying to reach people who already signed up.

Creative: what to actually run

Promote your proven organic winner. The post that already got comments and saves. It has demonstrated it can hold a stranger's attention — that's the hardest thing to know in advance, and organic already told you for free.

Run 3–5 creative variants, not one. Same offer, different hooks/openings/thumbnails. Let the platform find the winner; it's much better at that than you are. Kill the losers after a week and put the budget behind what's working.

What works in app ads specifically: - **Show the product doing the thing.** A ten-second screen recording of your app solving the problem beats any amount of copy about benefits. - **Lead with the problem, not the app.** "Tired of guessing what a card's worth?" outperforms "Introducing Scout AI." - **Native-looking beats polished.** Ads that look like organic posts get more engagement than ads that look like ads. Resist the urge to make it slick. - **Honest specifics beat superlatives.** "Prices any rookie card in 10 seconds" beats "The ultimate card tool."

Where you send them matters as much as the ad. Send traffic to your patched landing page or a purpose-built page that matches the ad's promise — never to a generic homepage. If the ad says "price any card in 10 seconds," the page had better say that too. Message-match is one of the biggest levers on conversion rate.

How to read the results (the diagnostic ladder)

This is the most valuable part of the section, because it turns a confusing dashboard into a specific diagnosis. Work down the ladder in order — each rung tells you where the break is.

Rung 1 — Is the ad being shown? (CPM / impressions) If your cost per thousand impressions is wildly high, your audience is too narrow, too competitive, or your account is new. Widen the audience slightly.

Rung 2 — Is the ad interesting? (CTR — click-through rate) Low CTR means the **creative or the hook is the problem**, not the targeting. People saw it and scrolled past. Fix: new hooks, new opening frames, more native-looking creative. As a rough rule of thumb, CTR under about 1% on social is a creative problem; there's a wide range by platform and niche, so treat it as a signal, not a law.

Rung 3 — Do clicks become signups? (conversion rate) Good CTR but nobody signs up means the **landing page or the offer is the problem** — the ad wrote a check the page didn't cash. Fix: message-match, simplify the page, check the signup path again. *This is the exact leak Phase 2 was about, and it's why the prerequisites matter.*

Rung 4 — Is the math sustainable? (cost per signup, then cost per customer) Now you're comparing what a signup costs to what a customer is worth. If a customer is worth $120 over their life and a signup costs you $4 with 1-in-10 converting to paid, you're paying $40 to earn $120 — that works, scale it. If the numbers don't work, you have a pricing/retention question, not an ads question.

The rule underneath the ladder: each rung isolates a different failure. Don't change three things at once — you'll learn nothing. Find the rung that's broken, fix only that, re-measure.

Retargeting: the highest-ROI ad you'll ever run

Most people who visit your site don't convert on the first visit. Retargeting shows ads specifically to people who already visited — and it consistently outperforms cold traffic by a wide margin, because these people already know who you are.

Set up a simple retargeting audience of "visited the site in the last 30 days but didn't sign up." Show them a different angle: a testimonial, a specific feature, an objection-handler, a limited offer. It costs little because the audience is small, and it's usually the cheapest conversions you'll buy.

Do this as soon as you have enough traffic to build the audience. Many builders run *only* retargeting profitably for months before they ever scale cold traffic.

Scaling (and when not to)

Don't scale until the math works. If your cost per customer is above what a customer's worth, more budget just loses money faster.

When it works, scale slowly. Increase budget roughly 20–30% at a time, every few days. Big jumps throw the campaign back into the learning phase and performance craters.

Watch for creative fatigue. Performance decays as your audience sees the same ad repeatedly — especially on TikTok, where it can happen in a week or two. This is exactly why the Content Engine matters: your organic pipeline is also your ad-creative pipeline. Fresh creative is the fuel that keeps a working campaign working.

The five ways first-timers waste money

1. **Running ads to a broken funnel.** Fix Phase 2 first. Always.
2. **Choosing a Traffic or Engagement objective** and celebrating cheap clicks that never convert.
3. **Touching the campaign daily** during the learning phase, resetting it over and over.
4. **Testing everything at once** — new creative, new audience, new page — so no result can be attributed to anything.
5. **Quitting after two days.** Day-two numbers are noise. Judge after the learning period and a full week of stable delivery.

Paid Ads → Do This Now

Goal: run one honest $150–250 experiment that returns a real answer. Timeline: ~2 weeks.

1. **Confirm the four prerequisites** — patched funnel, a proven organic winner, conversion tracking installed, audience profile in hand. Don't skip this gate.
2. **Install the pixel and fire a signup event.** Verify it actually registers a test signup before spending anything.
3. **Pick ONE platform** — Meta if consumer, Google Search if they search the problem, Reddit if they live in a specific community.
4. **Build the audience from your profile** — stack the interests and the "who they already follow" accounts. Keep it in the low hundreds of thousands to low millions, not tens of millions.
5. **Set objective = Conversions**, budget $10–20/day, and schedule 7–14 days.
6. **Upload 3–5 variants** of your proven organic winner — same offer, different hooks.
7. **Point them at the patched page** with message-match to the ad.
8. **Launch, then do not touch it for 3–5 days.** Let it learn.
9. **After the learning phase, walk the diagnostic ladder** — CPM, then CTR, then conversion rate, then cost per signup. Identify the single broken rung.
10. **Fix that one rung and re-run.** Or, if the math works, scale 20–30% at a time and start a retargeting campaign.

You'll know it worked when: you can say a specific sentence like "people click but don't sign up, so my landing page is the problem" — instead of "ads didn't work." That sentence is what you paid for.

> ***Note on this section:*** *unlike the rest of this book, I can't show you my own before/after here yet — this is the one play I hadn't run when I wrote this. What you're reading is the plan I built for my own first test, using the same discipline as everything else: small budget, one variable, real tracking, honest reading of the result. When I run it, you'll get the numbers, good or bad.*

Do This Now

Goal: stop posting into a vacuum. Build ONE correct account, get your first 50 real followers of the specific person, wire an engagement loop, then run one small controlled test. This is the phase that takes weeks, not hours — so here's the order.

1. **Pick ONE app and ONE platform.** Not three apps across four accounts. One. (Choose the platform where your customer actually hangs out.)
2. **Create — or claim — the correctly-branded account.** The account name and bio must signal your exact customer in a glance. If someone lands on the profile, they should think "this is for me." No posting your app from an unrelated brand's account.
3. **Set the bio link to a single clear destination** — your patched landing page (or a Beacons page whose top link is the app, not a pile of categories).
4. **Before broadcasting, spend 20 min/day for 10 days being a member of the niche:** follow 10 relevant accounts, leave 5 genuine comments, answer questions in the community. You're becoming a familiar face, not a billboard. This is how the first 50 *right* followers show up.
5. **Post with one engagement trigger every time.** End on a real question, take a stance, or use "comment [word] and I'll send you the [thing]" — which manufactures comments AND captures leads. Confirm every post is actually PUBLIC (check the privacy setting — this bit me).
6. **Put a friction-free path to the app in every post** and watch click-throughs, not views. Views feel good; clicks pay.
7. **Only after the account can hold a small audience, run one paid test.** $5–10/day for 5 days, audience tightly targeted to your specific person, pointing at your best-performing organic piece. You're not buying customers — you're buying a clean answer to "does this convert when the right person definitely sees it?"

You'll know it worked when: you have 50+ followers who are genuinely your customer, posts are getting real comments (not just views), and you can see clicks landing on the app — before you spend a dollar on ads.

> ***Here's me doing it — Scout AI (my real plan)*** *One app:* ***Scout AI.*** *One hub:* ***CardVaultCarl*** *— it's already correctly card-branded, which fixes my old mistake of posting card content from Simple Digital Systems and Retired Chaos. Consolidate: stop scattering across RookieCardGuideAI + SDS + Retired Chaos. Everything card goes through CardVaultCarl on TikTok + Instagram. Bio link → Beacons, top link = Scout AI (the patched free signup), not a category list. 10-day warm-up: comment on r/sportscards-adjacent TikToks, answer "what's this worth" questions, follow active collectors/flippers. Every post: end on "what rookie are you holding right now?" + confirm PUBLIC (I had a batch stuck on "Only me" — never again). Then: first-ever paid test, $10/day × 5 days, targeted to sports-card-collector interests, pointing at my best rookie-card hook. First time I'll actually control who sees it.*

Phase 3 exit checklist

Don't call your launch "not working" until every box is checked:

- [ ] I'm promoting **one app** from **one correctly-branded account per platform** — the account's name and bio signal my exact customer.
- [ ] I've spent real effort **building a following of the specific person**, not just broadcasting at strangers.
- [ ] Every post has **one engagement trigger** — a question, a stance, or a comment-to-get mechanic.
- [ ] Every post has a **clear, friction-free path to the app**, and I'm measuring click-throughs, not just views.
- [ ] I've **verified my posts are actually public** (no accidental "Only me").
- [ ] Before spending on ads, my organic content has proven it can **hold and engage** at least a small correct audience.
- [ ] When I do test paid, I'm using it to **control the audience** and isolate offer-fit — not to rescue content that isn't working yet.

If you've been posting into silence, I'd bet the failure is in the first three boxes, not your hooks. That's good news: those are the most fixable things on the list.

Next: Phase 4 — Optimize. You've patched the bucket, built the right small audience, and connected it to the app. Now we measure what actually matters (hint: it was never "views"), find the next leak, and turn a trickle of the right people into paying customers who stay.

PART THREE

Optimize

Find the one leak that matters, fix it, measure, repeat — forever.

Phase 4 — Optimize

chase the funnel, not the vanity metric

Phase 4 — Optimize

You've patched the bucket, built a small correct audience, and connected it to the app. Now you find the next leak — and turn a trickle of the right people into customers who pay and stay. This is the phase that never ends, and that's the point.

The number that lied to you

Let's go back to where this whole book started.

Scout AI had **584 visitors and a 60% bounce rate.** For months, that bounce rate was the number that got stared at, worried over, and blamed. It felt like the problem. It wasn't. It was a *distraction* — a number that looked like a diagnosis but was actually just noise.

Here's the trap of Phase 4, and it's the trap that keeps builders busy for months while getting nowhere: **you optimize the number that's easy to see instead of the number that matters.** Bounce rate is easy to see. Views are easy to see. Follower count is easy to see. And all three can move in the right direction while your bank balance stays at exactly zero.

The entire skill of this phase is knowing *which* number to chase — and having the discipline to ignore the ones that feel like progress but aren't.

The only funnel that matters

Every app, no matter what it does, has the same five-stage funnel. Money leaks out between stages. Your job in Phase 4 is to find the *one* stage where the biggest leak is, fix only that, and then look again.

1. **Visitors** — people who land on your app at all.
2. **Signups** — visitors who create an account (free or paid).
3. **Activated** — signups who actually *use* the thing once, and get the "oh, I get it" moment.
4. **Paying** — activated users who put money down.
5. **Retained** — paying users who are still paying next month.

Here's the rule that will save you months: **fix these in order, and only fix the earliest one that's broken.** There is no point optimizing your pricing page (stage 4) if nobody gets past signup (stage 2). There's no point driving more visitors (stage 1) if the ones you have all bounce. Find the first stage where people fall off a cliff, fix *that*, and re-measure. One leak at a time, earliest first.

Watch how this plays out on the real app.

Scout AI, by the funnel

When we actually put Scout AI's numbers into these five stages, the "60% bounce rate" panic evaporates and the real problem snaps into focus:

- **Visitors:** yes — 584 of them. Stage 1 was *working.* Traffic was never the core problem.
- **Signups: zero.** Everyone died here.

We didn't need to look at stages 3, 4, or 5. The funnel broke completely at stage 2, and everything downstream was irrelevant until that was fixed. And when we looked *closely* at stage 2 — walked the path as a stranger, the way Phase 2 teaches — we found the reason: the "Get Started Free" button led to a signup screen demanding $9.99/mo, and the brand changed names three times on the way. Visitors arrived, got confused and distrustful at the exact moment of signup, and left.

So the highest-value optimization in the entire app wasn't the bounce rate, the hooks, the pricing, or the traffic. It was **fixing the one screen where the funnel snapped.** That's the whole lesson: the bounce rate was a number that *looked* like the problem; the signup cliff was the number that *was* the problem. Chase the funnel, not the vanity metric.

What to measure (and what to ignore)

Measure these — they map to real money:

- **Signup rate** (signups ÷ visitors) — is stage 2 working?
- **Activation rate** (used-it-once ÷ signups) — do people reach the "aha"?
- **Paid conversion** (paying ÷ activated) — does value turn into money?

- **Retention / churn** (still paying next month) — does the money *stay*?

Ignore these — they feel like progress and aren't:

- Raw view counts and impressions (Scout AI had plenty; they meant nothing).
- Bounce rate *in isolation* (it's a symptom, not a diagnosis — it tells you people left, not why).
- Follower count as a goal in itself (followers who never sign up are an audience you're renting, not a business).
- Likes (the softest signal there is — remember Scout AI's posts had a few likes and zero customers).

The test for any metric: **if this number doubled, would I make more money?** If you can't draw a straight line from the metric to a dollar, it's a vanity metric. Track it if you must, but never *optimize* for it.

Using the optimize prompts (Claude Prompts 7 & 8)

The last two prompts in the kit are your Phase 4 engine. They only work once you have *some* real data — even a handful of signups — so this is where they finally earn their place.

Prompt 8 — the monthly review (run this first, once you have ~30 days of any real numbers):

> *Act as a SaaS performance analyst. Here are my last 30 days: Visitors [X], Signups [X], Activated [X], Paying [X], Churn [X], MRR [X]. Walk my funnel stage by stage, tell me the ONE stage where I'm losing the most people, and give me the single highest-impact change to fix that stage next month. Ignore vanity metrics.*

That "ignore vanity metrics" line matters — it keeps the AI from cheerleading your view count. The output tells you which *one* leak to attack.

Prompt 7 — the churn diagnostic (run this once you have paying users leaving):

> *Act as a SaaS retention expert. My app is [app] at [$/month]. Here's what I know about users who cancelled in the first 30 days: [whatever you know]. Give me the 3 most likely reasons they left, and for each, one specific product change that would prevent it. Rank by likely impact.*

Churn is the last leak because it's the most expensive: acquiring a customer is hard, and losing them in month one means you paid the full acquisition cost for one month of revenue. A small retention fix is often worth more than a big acquisition push — but only once you *have* paying users to retain. Don't run this one until stage 4 is real.

The mindset that separates finishers from quitters

Here's the thing about Phase 4: **it doesn't end.** There is no moment where the app is "optimized" and you're done. There's always a next leak — a signup step that's one field too long, an onboarding that loses people on day 3, a pricing tier nobody picks. The builders who win aren't the ones who find the perfect launch. They're the ones who treat their app as a permanent series of "find the next leak, fix it, measure, repeat."

That sounds exhausting. It's actually freeing — because it means **you never have to get it right all at once.** Scout AI didn't need a perfect relaunch. It needed one honest look at the funnel, one fix to the stage that was broken, and then the discipline to look again. That's a thing you can do this week. And next week. And the week after.

The apps that die are the ones whose builders concluded, after the silence, that the whole idea was wrong. Usually the idea was fine. There was just a leak they couldn't see — and they quit one fix short of finding it.

Do This Now

Goal: set up the five-stage funnel so you can see the ONE leak that matters, and build the monthly habit of finding the next one. Budget: about 1 hour to set up, then 30 min/month.

1. **Write down your five numbers**, even if four of them are zero: Visitors, Signups, Activated (used it once), Paying, Retained (still paying next month).

2. **Find the first cliff** — the earliest stage where the number craters versus the one before it. That's your only priority. Ignore every later stage until this one's fixed.
3. **Make sure you can actually measure the cliff.** You need signup events and activation events tracked (most Lovable/Supabase apps can log these; your web analytics shows visitors). If you can't see a stage, you can't fix it — instrument it first.
4. **Fix only that one stage.** Resist the urge to redesign everything. One leak, one fix, then re-measure.
5. **Run Prompt 8 once a month** with your real numbers. It tells you where the biggest leak moved to. Fix that. Repeat.
6. **The day a paying user cancels, run Prompt 7.** Not before — churn tools on a zero-customer app is theater.

You'll know it worked when: you can name your single biggest leak on any given week, and you're fixing that instead of polishing things that don't move money.

> ***Here's me doing it — Scout AI*** *My five numbers today: Visitors 584, Signups 0, Activated 0, Paying 0, Retained 0. First cliff:* ***Visitors → Signups.*** *Everything died at signup — which is exactly what the Phase 2 fix targeted. So my #1 job isn't traffic or pricing; it's watching whether the signup fix moves that second number off zero. Instrument: confirm Supabase is logging signup + first-use events so I can see stage 2 and 3 for real. Next monthly review: once CardVaultCarl traffic is flowing, run Prompt 8 and see if the cliff has moved from "signups" to "activation" — then chase whatever's newly biggest.*

Phase 4 exit checklist

You're not "done" (you never are), but you're operating correctly when:

- [ ] I've mapped my real numbers onto the **five funnel stages.**
- [ ] I know the **single earliest stage** where I'm losing the most people.
- [ ] I'm optimizing **that stage only** — not polishing a later stage while an earlier one leaks.
- [ ] I'm tracking **money-linked metrics** (signup / activation / paid / retention), not vanity metrics.
- [ ] I'm running **Prompt 8** monthly to re-find the biggest leak.

- [ ] I'll run **Prompt 7** the moment I have paying users starting to churn.
- [ ] I've internalized that this **repeats** — find leak, fix, measure, again — and I'm not waiting to "finish."

So — you built it. Then what?

You went back and read the whole thing, so let's close the loop the title opened.

You built it, and nobody came. That sentence has broken a lot of good builders, because it *feels* like a verdict on the idea, or on you. It's neither. It's just a signal that there's a leak between *built* and *bought* — and now you have the four-phase system to find it:

- **Ideate** — make sure a specific person will pay, and you can say why in one sentence.
- **Build** — make the value obvious in five seconds, and stop the app sabotaging itself.
- **Launch** — get the right small audience to it, from the right account, with a reason to act.
- **Optimize** — chase the funnel, not the vanity metric, one leak at a time, forever.

Scout AI is a real app that had 584 visitors and zero customers. Every fix in this book is one we actually made to it — the honest signup, the unified brand, the promotion autopsy, the funnel diagnosis. Not theory. Receipts.

You already did the hard part. You built the thing most people only talk about. This was never about building more. It was about the part nobody hands you a tool for — until now.

Go find your leak.

Build Then What — the playbook for the part nobody teaches. Proof, not hype.

The Full Case Study — Scout AI, Start to Finish

Every principle in Build Then What came from one real app. Here's the whole story in one place — what was broken, what we changed, and what's still being measured. Nothing here is hypothetical. These are the receipts.

The app

Scout AI is an AI-powered tool for sports card collectors and flippers — it gives you exact buy prices, sell targets, and timing on rookie cards, aimed at three kinds of people at once: serious collectors, active flippers, and beginners who heard you can make money flipping cards and want a system that removes the guesswork. It was built on Lovable and Supabase, then migrated to Cloudflare and GitHub. A genuinely useful product, live for months.

And it had made zero dollars.

The numbers that started it

Over one 90-day stretch:

- **584 visitors**
- **60% bounce rate**
- Average time on site: about **4 minutes** (the people who stayed were genuinely interested)
- Trials started: **0**
- Paying customers: **0**

Plus a promotion effort that was anything but lazy: **460 posts** across TikTok, Instagram, and Facebook.

Real product. Real traffic. Real effort. Zero customers. This is the exact "you built it, nobody came" wall — and the point of the case study is that the reasons were findable and fixable, once we stopped guessing and started walking the app the way a stranger does.

Phase 1 audit — was the idea even sound?

Before fixing anything, we checked the foundation. Running Scout AI backward through the ideation questions:

Who's the specific person? "Sports card people" isn't a person. There were really three — collector, flipper, beginner — and the app was talking to all of them at once, landing squarely on none. But the flipper and beginner share a real, wallet-open problem: "*I don't know which cards are worth buying right now or what to sell them for, and I'm scared of overpaying.*"

The one-sentence test:

> *For a beginner who wants to flip sports cards without overpaying, Scout AI is the AI scout that tells you exactly what to buy, what to sell it for, and when — unlike scrolling eBay for hours, it just hands you the answer.*

That passed. Which delivered the single most important verdict of the whole project: **Scout AI's problem was never the idea.** The foundation was sound. That told us the leaks were downstream — in how the app presented itself and how it was promoted — and therefore worth fixing rather than scrapping.

Phase 2 fixes — the app was sabotaging itself

The landing page copy was actually good. The leak wasn't on the page a visitor *read* — it was in the path a visitor *walked*. Walking it as a stranger revealed two things.

The "free" promise broke at signup. Every call-to-action said "Get Started Free." There was a real free tier. But clicking "Get Started Free" led to a signup page whose button read **"Create Account — $9.99/mo"** with fine print "$9.99/month. Cancel anytime." Promised free three times, asked for money at the decisive moment. The bitter part: the app wasn't even charging at signup — it created a free account. The "$9.99/mo" was dead label text that never got updated. A lie the app told against its own interest.

Fixed: button → **"Create My Free Account"**; fine print → **"Free to start. No credit card required. Upgrade only when you're ready."** About fifteen minutes of work.

The brand changed names three times in two clicks. Domain said *scoutairookie.com*, homepage said *Scout AI*, signup page showed an "*RC*" logo and "*Create your RookieCard Guide account.*" Three identities before you'd even made an account — and every switch is a "wait, am I in the right place?" flinch at the worst possible moment.

Fixed: one name everywhere. Every "RC" and "RookieCard Guide" across signup, login, and profile became **Scout AI** — matching the front door.

Both fixes shipped. Confirmed live: free signup works, Scout AI shows consistently across all pages. **This is the first real "before → after" in the kit** — the app stopped sabotaging itself at the exact moment of decision.

Phase 3 autopsy — 460 posts, five reasons they failed

This is where the effort was, and where the honesty matters most. The 460 posts produced almost nothing, and the analytics showed exactly why — five mechanical failures, none of them about talent:

1. **The hooks were good.** "How to spot the next breakout rookie card," "3 rookie card mistakes collectors make," "This $5 card is now worth $50." Textbook problem/result hooks. So the failure was *not* the writing.

2. **Posting into a vacuum.** The account had no established following of card people, so the algorithm had no one to show the posts to. Views sat at 20–129 with essentially zero comments — the signature of volume with no audience.

3. **Account/brand mismatch.** Scout AI's card content was posted largely from an account branded around faceless digital-business education — the wrong room by construction. Card hooks shown to "make money online" followers.

4. **Fragmentation.** The effort was split across three or four different handles per app, so no single account ever built the critical mass where the algorithm starts working for you.

5. **Mechanical bleed.** A batch of posts were accidentally set to **"Only me"** — private, invisible, work done for nothing. And the app's own analytics showed traffic was

almost entirely *direct*: the social platforms barely registered as referrers, meaning even the views that happened weren't clicking through to the app.

The lesson that fell out of this is the most valuable in the kit precisely because it was earned the hard way: **good hooks + huge effort + wrong/no audience + no engagement loop + no targeting = zero.** The fix isn't "post more" or "write better." It's one app, one correctly-branded account per platform, an audience built before the ask, an engagement trigger on every post, and only then a small paid test to control who sees it.

Phase 4 diagnosis — the number that lied

The "60% bounce rate" that started the panic turned out to be a distraction. Mapped onto the real funnel:

- **Visitors:** 584 — stage 1 was working.
- **Signups:** 0 — everyone died here.

No need to look further down. The funnel snapped completely at stage 2, and the Phase 2 fixes were aimed at exactly that break. The lesson: chase the funnel, not the vanity metric. Bounce rate *looked* like the problem; the signup cliff *was* the problem.

Where it stands right now (the honest part)

Here's what's true as of this writing, stated plainly because the whole brand is "proof, not hype":

- **Idea validated** as sound (Phase 1).
- **Signup bug fixed and live** — the app no longer promises free then asks for money (Phase 2).
- **Brand unified** to Scout AI across every page (Phase 2).
- **Promotion failure fully diagnosed** — five specific, named causes (Phase 3).
- **Signups:** still being measured post-fix. The bug is gone; the numbers need time and traffic.

- **The "redo it right" launch** — consolidating promotion onto a single correctly-branded card account (CardVaultCarl) — is the next real-world step. That's the test of whether the Phase 3 lessons produce the "after."

That last part is deliberate. This case study doesn't claim a happy ending it hasn't earned yet. It claims something more useful and more honest: **a real app, a real diagnosis, real fixes shipped, and a clear measurement in progress.** The day the signups move, this section gets its final paragraph — and it'll be true.

That's the difference between this and a course sold by someone who'll never show you their own numbers. You're not being sold a fantasy. You're watching the actual work.

Build Then What — proof, not hype.

Appendix — The 8 Prompts

Every prompt referenced in Build Then What, collected in one place and ready to copy. Paste them into Claude, ChatGPT, or whatever AI tool you use. Replace anything in [brackets] with your own details. The more specific you are, the better the output — vague prompts get vague answers.

How to use this appendix: the prompts are grouped by phase, in the order you'll actually need them. Don't run them all at once. Run each one when you reach that point in your build, using what you learned from the previous phase as the input to the next.

Phase 1 — Ideate

Prompt 1 — Find a Profitable Niche

Use this to generate validated app ideas, or to check whether the app you already built is aimed at a real, reachable audience.

```
Act as a Micro-SaaS strategist. Suggest 3 web app ideas with real,
current demand and recurring-revenue potential. Avoid generic ideas —
favor specific, underserved niches.

For each idea, give me:
- Target audience (be specific — not "small businesses," but
  "solo real-estate agents in their first two years")
- The problem it solves (stated the way the customer would say it,
  in their words)
- The core MVP feature (the one thing it must do)
- The unique advantage (why me, not a spreadsheet or an existing tool)
- Monthly pricing (a realistic number this audience would actually pay)
- Where these people already gather (subreddits, Discords, Facebook
  groups, forums — name them)

Then rank the three from most to least "painful problem," and tell me
which to build first and why.
```

Prompt 1B — Pressure-test the idea

Run this on your top idea before you build anything. It's the step that separates builders who sell from builders who don't.

```
I'm considering building [idea] for [target audience]. Play the
skeptic, not the cheerleader.

- What are the 3 strongest reasons this would fail to get paying
  customers?
- Who already solves this problem for that audience (including free
  tools, spreadsheets, and "they just live with it")?
- What would have to be true about this audience for them to actually
  pay monthly — and how could I check that this week, before building?
- If you had to bet, would this audience pay, or just say "neat"?
```

Phase 2 — Build

Point these at the stranger-facing layer — your landing page and signup path — not just the app's internal features. That's where the money leaks.

Prompt 2 — Plan the Interface

```
Act as a UX designer. For my [app idea], design a clean interface
with the key screens, user flow, and essential UI elements. Make sure
a brand-new user understands the app's value within 30 seconds of
landing. Point out anywhere a first-time visitor might get confused
about what to do next.
```

Prompt 3 — Write the Core Logic

```
Act as a senior software engineer. For my [app idea], write clean,
well-structured code for the core feature [main function]. Then do
the same for the signup and trial flow specifically — this is the
path where most apps lose people, so make it simple and trustworthy.
Explain each section clearly so it's easy to maintain and update later.
```

Prompt 4 — Design the Database

```
Act as a backend architect. For my [app idea], design a simple,
scalable database schema with the required tables, fields, user data,
usage tracking, and subscriptions. Include a place to store captured
emails from people who are interested but haven't signed up yet.
Also highlight the 3 most common database mistakes to avoid.
```

Prompt 5 — Build the Interactive Features

```
Act as a frontend developer. For my [app idea], build the interactive
features [list features] with clean code. Explain how each connects to
the backend and what to test before launch. Add this to the test list:
walk the entire path from landing page to created account as a
brand-new stranger, reading every word cold, and flag anything that
breaks a promise or causes hesitation.
```

Phase 3 — Launch

Prompt 6 — Build the Subscription Flow

```
Act as a SaaS monetization expert. Design a high-converting
subscription flow for [app] with a free trial or free tier, a clear
upgrade path, and a checkout page that maximizes conversions while
reducing friction. Make sure every promise on the way in (especially
the word "free") is still true at every later step — no surprises at
signup or checkout.
```

Phase 4 — Optimize

These two only work once you have real data. Don't run them on an app with no users — you'll be optimizing theater.

Prompt 8 — Review Month One

Run this first, once you have ~30 days of any real numbers.

```
Act as a SaaS performance analyst. Here are my last 30 days:
Visitors [X], Signups [X], Activated [X], Paying [X], Churn [X],
MRR [X].

Walk my funnel stage by stage, tell me the ONE stage where I'm losing
the most people, and give me the single highest-impact change to fix
that stage next month. Ignore vanity metrics like raw views and likes.
```

Prompt 7 — Reduce Churn

Run this once you have paying users starting to cancel.

```
Act as a SaaS retention expert. My app is [app] at [$/month]. Here's
what I know about users who cancelled in the first 30 days: [whatever
you know].

Give me the 3 most likely reasons they left, and for each, one
specific product change that would prevent it. Rank by likely impact.
```

The one rule for all eight

A prompt is only as good as what you point it at. These aren't magic — they're leverage. The builder who feeds them specific, honest inputs (a real audience, a real number, a real problem stated in the customer's words) gets output worth acting on. The builder who feeds them "make me a profitable app" gets a plausible-sounding answer that helps no one.

Be specific. Tell the truth in the brackets. Then go do the un-promptable part: talk to real people, ship the fix, watch the numbers.

Build Then What — proof, not hype.