

The App Idea Research Prompt Guide

Screenshot Any App, Get a Full Business Case: The Claude Reverse-Engineering Prompt + a Worked Example

01

Step 1: Screenshot the Target App

Pick a category you already understand — fitness, productivity, finance, habit tracking — and open the top-ranked app in that category on the App Store or Google Play. Screenshot the listing page: the icon, the name, the short description, and the star rating with review count. That's enough signal for Claude to work with.

QUICK TIPS

- A second screenshot of the main app screen adds detail but isn't required.
- Pick a paid or freemium app — free ad-supported apps rarely have a clean revenue model to analyze.

02

Step 2: The Reverse-Engineer Prompt

Drop your screenshot into Claude with this exact prompt. It's built around five outputs — ICP, pain point, gap, revenue model, and blue ocean angle — so Claude reasons about the business, not just the feature list.

PASTE THIS

I'm sharing a screenshot of [App Name] from the App Store.

Reverse engineer this app and give me:

1. The ideal customer profile (ICP) — be specific, not "busy people"
2. The core pain point this app solves for that ICP
3. The gap: what this app still does badly or ignores entirely
4. 2–3 plausible revenue models this app is likely running
5. A "blue ocean" angle — a version of this that serves the same ICP without competing head-on on the same feature set

Step 3: Worked Example — Reverse-Engineering a Top Habit Tracker

Here's the prompt run against a real top-5 habit-tracking app (4.8 stars, 40,000+ reviews, \$4.99/month). Claude's output: ICP — people who have tried and abandoned 3+ habit apps already and blame the apps, not themselves, for the failure. Pain point — the app tracks habits but does nothing when a streak breaks, so the user just quits. Gap — zero recovery flow after a missed day; the app treats a broken streak as failure instead of data. Revenue model — \$4.99/month subscription plus a one-time lifetime unlock, likely a 70/30 split between the two. Blue ocean angle — a habit app that's built entirely around streak-recovery, not streak-building: it asks 'what got in the way?' after every miss and adjusts the habit instead of just resetting the counter to zero.

💡 QUICK TIPS

- Notice the blue ocean angle doesn't out-feature the original app — it reframes the same problem.
- Run this on 2-3 competing apps in the category to see which gap keeps repeating; that's your strongest signal.

Step 4: Turn It Into a Build Prompt

Once you have the analysis, this second prompt compresses it into a scoped v1 you can paste straight into Claude Code as your first message on a new project.

PASTE THIS

Using the reverse-engineering analysis above, write a build prompt for Claude Code that scopes a v1 around the blue ocean angle.

Include:

- The one core feature that proves the blue ocean angle works
- The screens needed for that feature only (max 3)
- Suggested tech stack and why
- What to explicitly leave out of v1

Format it as a single prompt I can paste directly into Claude Code.

💡 QUICK TIPS

- Keep v1 to one feature. The point is to prove the blue ocean angle works, not to match the incumbent app screen-for-screen.

Step 5: What to Do With the Output

Paste the build prompt into a fresh Claude Code session and let it scaffold the project against the spec. Treat the ICP, pain point, and gap as your north star for every later feature decision — if a feature doesn't serve the ICP or close the gap, it's scope creep, not v1.

💡 QUICK TIPS

- Save the full reverse-engineering analysis as a NOTES.md file in your project root so future Claude Code sessions can reference the original research.
- Re-run Step 2 on a second competing app before you commit — one data point is a hunch, two is a pattern.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>