

Read reviews like a **product** designer.

The exact Claude prompts I use to analyze competitor app reviews — no Python, no setup, no data background needed. Just your browser and a few reviews.

free

takes about an hour

STEP ONE

Get the reviews

manually — no code required

You need raw material before Claude can do anything. Here's how to collect 50–100 reviews per competitor **without writing a single line of code**.

1

Find your competitors on Google Play

Search for 3–5 apps in your space. You want apps solving the same problem your product solves — or the closest thing to it.

2

Filter by rating

On each app's Play Store page, filter reviews to show 1-star and 2-star first. These surface the real pain points. Then switch to 3-star — those often hide feature requests in the middle of complaints.

4 and 5 stars are useful too, but start with the problems.

3

Copy into a doc

Open a Google Doc or Notes app. For each app, paste 20–30 reviews. Include the star rating next to each one. Label which app each group came from.

Format doesn't need to be perfect. Just readable.

4

Paste into Claude

Open claude.ai. Start a new conversation. Paste all your reviews in one message with the priming prompt below — and you're ready to analyze.

STEP TWO

The prompts

use these in order — each one builds on the last

PROMPT 01 — PRIMING

Tell Claude what it's looking at

Send this first with your reviews pasted below it. This sets the context so every follow-up prompt works properly.

Below are real app store reviews I collected from [X] competitor apps in the [your market] space. Each group is labeled with the app name and star rating.

Read through all of them and confirm when you're ready. I'll be asking you several questions to help me understand this competitive landscape.

PROMPT 02 — PAIN POINTS

Find what keeps showing up

This is the core analysis. You're looking for patterns, not individual complaints.

For each app, identify the top 3-5 complaint categories. Group similar complaints together under one theme — don't just list individual reviews.

For each category:

- Give it a short label (e.g. "sync issues", "missing features", "confusing UI")
- Estimate roughly how many reviews mention it
- Quote 1-2 real review lines that represent this complaint well

Format it as a simple list per app.

Find what users keep asking for

Pain points tell you what's broken. Feature requests tell you what nobody has built yet.

```
Now look across all the apps together. What features or improvements do users keep requesting that none of these apps seem to have built well?
```

COPY

```
List the top 5 feature gaps. For each one:
```

- What users are asking for (in their own words)
- Which apps get this complaint most often
- Whether any app partially addresses it already

See who's winning on what

This is where the real intelligence starts — understanding the competitive dynamic, not just individual apps.

```
Based on the reviews, give me a brief competitive comparison:
```

COPY

- Which app has the most loyal users (and why, based on review sentiment)?
- Which app seems to be losing users most actively?
- What's one thing each app does that users genuinely appreciate?
- If you had to describe the "white space" in this market — the thing users want that nobody offers well — what would it be?

```
Keep it concise. One short paragraph per question is enough.
```

Turn it into something you can act on

This is the output that matters. Ask Claude to synthesize everything into a strategic brief you can actually use.

```
Now write me a short competitive intelligence brief based on everything we've discussed. Structure it as three sections:
```

COPY

```
BUILD PRIORITY — the one problem I should focus on if I'm building in this space, based on what users keep asking for
```

```
COMPETITIVE THREAT — which competitor is doing something right that others aren't, and why it matters
```

```
WATCH CLOSELY — a trend or pattern in the reviews that might signal where this market is heading
```

Keep it direct. Write it like you're briefing a product team, not writing a report.

TIPS

Get better results

small adjustments that make a big difference



More reviews = better patterns

20 reviews gives you impressions. 50–80 reviews per app gives you actual patterns. The more you paste in, the more confident Claude's analysis becomes.



Add context if it helps

If your market has specific nuances — regulatory stuff, a specific user type, a particular use case — mention it. "These are all menstrual tracking apps used primarily by women 18–35" changes the quality of the analysis.



Push back if the output feels generic

If Claude's response could apply to any app in any market, it's too generic. Just say: "Be more specific — reference actual review text and give me something I couldn't have guessed without the data."



Save your briefs

Copy the final brief into a doc. Do this every month and you'll start to see how sentiment shifts over time. That's when it becomes a real intelligence system.

a pattern over months is more valuable than a snapshot.

WANT TO GO FURTHER?

The \$9 guide

for when you're ready to build a real system

This free guide gets you real insights from a handful of manually collected reviews. The \$9 guide takes that same methodology and scales it — scraping 500 reviews per competitor automatically, then building a live dashboard that updates every time you run it.



Python scraping script

Pulls 500 reviews per app from Google Play automatically. Run it in Google Colab — no local setup needed.



Full prompt pack for large datasets

Upgraded prompts designed for CSV-scale data — not 50 reviews, but 2,500. The patterns are much clearer at volume.



Live intelligence dashboard

Step-by-step walkthrough for building a Notion dashboard that surfaces your analysis in a format you can bring to meetings.



Beginner Python setup guide

Never run a Python script before? The guide walks through every step — install, run, troubleshoot — for both Mac and Windows.

GET THE FULL SYSTEM

2 weeks of work. *One afternoon* for you.

Step by step, even if you've never touched Python before. The scraping script, the prompts, and the dashboard — everything packaged so you can build it once and run it every month.

\$9 · link in bio *beacons.ai/kateintheloops*

@kateintheloops