

AI Design Prompt Pack

Reusable prompt patterns for research synthesis, moodboards, content variants and QA — the ones I actually run to move faster without outsourcing judgment.

Mariana Ordonez

Agentic Product & AI UX Designer

How to use it — Copy a prompt, replace every **[bracket]** with your own context, and paste in your material. Treat the output as a fast first draft to interrogate and edit — never a final answer. Keep your voice, your ethics and your design judgment in the loop, and ask the model to cite or flag anything it inferred.

Research & synthesis · make sense of raw input

Cluster interviews into themes

Turn messy notes into structured insight.

You are a UX researcher. Below are **[N]** raw interview notes/transcripts.

- 1) Cluster findings into 5–8 themes.
- 2) Per theme: a one-line insight, 2–3 exact verbatim quotes, and how many participants mentioned it.
- 3) Flag contradictions and surprises.
- 4) End with the top 3 design opportunities.

Use only what's in the data; mark anything inferred.
Notes: **[paste]**

Tip — Paste it raw and messy. Pre-summarizing biases the clusters.

Draft JTBD & lightweight personas

Convert themes into something designable.

From the themes below, write **[2–3]** jobs-to-be-done as: "When [situation], I want to [motivation], so I can [outcome]."

Then one lightweight persona per job: goals, frustrations, context of use.

Ground every attribute in a cited theme; label inferences as assumptions.
Themes: **[paste]**

Tip — The "label assumptions" clause doubles as a validation backlog.

Problem framing · design the right thing

Sharpen the problem statement

Pressure-test before any solutioning.

Act as a skeptical product lead. My problem statement: "**[draft]**"

- 1) Restate it as a testable problem.
- 2) List the hidden assumptions.
- 3) Give 3 sharper reframings at different altitudes (user, business, system).
- 4) Name the single metric that proves it's solved.

Tip — Run this first — it stops you from solving the wrong problem beautifully.

Ideation · diverge with intent

Diverge on distinct concepts

Spread, not polish.

Give me **[8]** genuinely different UX concepts for **[problem/flow]**, ranging safe → bold.

Per concept: a name, the core idea in one sentence, the main trade-off, and who it's best for.

Avoid variations on one idea — maximize spread.

Tip — Requiring the trade-off up front lets you kill weak ideas fast.

Moodboards & visual direction · set the look

Write a visual-direction brief

Language before pixels.

You are an art director. For a **[product type]** aimed at **[audience]** with a **[3 adjectives]** personality, propose 3 distinct visual directions.

Per direction: mood keywords, colour logic, type-pairing rationale, imagery style, 2 reference brands.

No hex codes yet — describe intent.

Tip — Keep the direction names; they become your comparison labels downstream.

Moodboard image generation

For Nano Banana / Gemini / any image model.

Generate a **[aspect ratio]** moodboard image: **[subject]**, **[lighting]**, **[colour palette]**, **[texture/material]**, **[mood]**, shot on **[lens/style]**. No text or logos.

Style reference: **[direction name]**.

Tip — Change one variable per run to learn what each term actually controls.

UX writing & content · variants with a voice

Microcopy variants on-voice

On-brand, not generic SaaS.

Voice: **[3 traits, e.g. warm, precise, unfussy]**.

Write **[5]** variants for this **[button / empty state / error]**: "**[context]**".

Constraints: ≤ **[N]** chars, no jargon, action-led.

Return a table with a one-line rationale each.

Tip — Feed it your voice chart; without it every option sounds the same.

Pressure-test copy for clarity

Find drop-off, not typos.

Read this **[PDP / landing / onboarding]** copy as a distracted first-time mobile user.
Where do I get confused, doubtful, or bored enough to leave?
Mark each spot, say why, and rewrite it tighter.
Copy: **[paste]**

Tip — The "distracted mobile user" role surfaces real friction, not grammar nits.

Spec a component for handoff

Developer-ready in one pass.

Write a dev-ready spec for this **[component]**:
anatomy, variants, states
(default/hover/focus/active/disabled/loading),
tokens used, responsive behaviour, content rules, edge cases.
Format as a concise table + notes.
Component: **[describe / attach]**

Tip — Feed the spec straight into Claude Code to scaffold the component.

Critique & QA · a second set of eyes

Heuristic critique of a screen

Structured, severity-ranked.

Critique this screen against Nielsen's 10 heuristics + WCAG AA basics.
Per issue: heuristic, severity (1–4), the specific element, a concrete fix.
Then rank the top 3 to fix first.
Screen: **[attach image / describe]**

Tip — Attach the real frame — described-only critiques miss layout and contrast.

Prototype with Claude Code

From spec to working preview.

In this repo, build a working prototype of **[flow]** using the existing design tokens and components. Keep it accessible and responsive.
Show me the diff and a preview, then wait for feedback before any styling polish.

Tip — Constrain it to existing tokens/components so the prototype matches your system.

Accessibility pass

Catch it before dev does.

Audit this component for accessibility: contrast, focus order, keyboard operation, semantics/ARIA, target size, reduced-motion.
List each violation with its WCAG criterion and a fix.
Component: **[code / describe]**

Tip — Pair with the accessibility section of my CRO checklist for a full sweep.