

PROMPT PACK

IDEA TO FIRST USERS: 20 AI PROMPTS FOR BUILDERS

A copy-and-paste prompt pack for validating ideas, finding first users, shaping an offer, shipping faster, and learning from feedback.

26 Oct 2026 · 5 min read

F4F · @USF4F ON X

AI is useful when you give it a clear job. These 20 prompts are built for the messy early stage: testing an idea, finding people who care, making the offer sharper, and shipping before you feel ready.

They are starting points, not answers. Give the model your real context, ask it to show its assumptions, and check anything important with a human.

HOW TO USE THIS PACK

- Replace every [bracket] with specific details about your project.
- Paste in real notes, messages, or feedback instead of describing them from memory.
- Ask for options first, then ask the model to challenge its own favourite.
- Keep the output short enough to act on today.

VALIDATE THE PROBLEM

1. PRESSURE-TEST THE IDEA

Act as a skeptical customer. I am building [product] for [audience]. List the 10 assumptions my idea depends on, rank them by how badly the project fails if they are wrong, and suggest one cheap way to test each assumption this week.

2. WRITE A BETTER CUSTOMER INTERVIEW

Turn this idea into a customer interview script: [idea]. Ask about past behaviour, not hypotheticals. Include 12 open questions, three follow-up probes, and three questions I should avoid because they lead the witness.

3. FIND THE STRONGEST PAIN POINT

Here are notes from [number] conversations with potential users: [notes]. Cluster the pain points, quote the strongest evidence for each, and tell me which problem appears most urgent. Flag anything that sounds like polite encouragement rather than real demand.

4. DEFINE A NARROW FIRST AUDIENCE

Build a simple ideal customer profile for [idea]. Give me three narrow segments, the trigger that makes each one look for a solution, where they spend time online, and what they currently use instead.

FIND YOUR FIRST USERS

5. CHOOSE WHERE TO SHOW UP

I need the first 10 users for [product]. Suggest seven specific places I could find them, including communities, events, search terms, and people already gathering the audience. For each, give me one useful way to contribute before asking for anything.

6. WRITE A NON-SPAMMY FIRST MESSAGE

Write three versions of a short message to [person or group] about [project]. Each must be under 80 words, reference something specific about their work, offer one small piece of value, and end with a low-pressure question. No fake compliments.

7. PLAN A SEVEN-DAY LAUNCH

Create a seven-day launch plan for [project] on X. Include one post per day, the goal of each post, a concrete hook, and a simple way to measure whether it worked. Assume I have limited time and no existing audience.

8. OUTLINE A LANDING PAGE THAT TELLS THE TRUTH

Draft a landing page outline for [product]. Include three headline options, the problem, the promised outcome, proof I can realistically gather before launch, the main objections, and one clear call to action. Do not invent testimonials or metrics.

SHARPEN THE OFFER

9. TURN SKILLS INTO A CLEAR SERVICE

Turn my skills [list] into three service offers for [audience]. For each, define the outcome, deliverables, timeline, a fair starting price, and what is explicitly out of scope. Make the smallest offer useful on its own.

10. COMPARE THE ALTERNATIVES

Compare my offer [offer] with five alternatives a customer could choose, including doing nothing and hiring someone more established. Explain where my offer is genuinely different and where I need to improve the offer instead of the messaging.

11. WRITE A LESS GENERIC VALUE PROPOSITION

Write a one-sentence value proposition for [product] using this structure: helps [audience] achieve [outcome] without [pain]. Give me 10 options, then pick the three least generic ones and explain why.

12. STRESS-TEST THE PRICE

I am considering pricing [product] at [price]. Ask me eight questions that would reveal whether the price is too low, too high, or badly packaged. Then suggest three pricing experiments I can run with early customers.

BUILD AND SHIP

13. CUT THE MVP DOWN

Break [project] into a two-week MVP. Only include what is needed to test [core assumption]. List what to cut, what I can fake manually, what I should buy instead of build, and how I will know whether it worked.

14. REVIEW A PRODUCT SPEC

Review this product spec: [spec]. Find gaps, contradictory requirements, and hidden dependencies. Rewrite it as a build checklist ordered by risk, with the smallest testable version first.

15. WRITE AN HONEST BUILD UPDATE

Help me write a build-in-public update about [what I shipped]. Include what I built, one thing that broke, what I learned, and what is next. Keep it under 150 words and do not make it sound more successful than it is.

16. GET A SIMPLER TECHNICAL PLAN

Act as a friendly senior engineer. Review this plan for [feature]. Point out the simplest implementation, the likely failure modes, and one thing I should test before adding more code or infrastructure.

LEARN AND IMPROVE

17. TURN FEEDBACK INTO DECISIONS

Summarise these user feedback notes: [notes]. Separate bugs, feature requests, confusion, and nice-to-haves. Recommend the next three changes and cite the evidence behind each recommendation.

18. RUN A PRE-MORTEM

Run a pre-mortem on [project]. Imagine it failed 90 days from now. List the 10 most likely reasons, rank them by probability and impact, and give one preventive action for the top three.

19. FOLLOW UP LIKE A HUMAN

Draft a follow-up message after meeting [person] at [event]. Reference one specific thing we discussed, offer one useful next step, and keep it under 90 words. Do not overstate the connection.

20. MAKE THE WEEK REALISTIC

I have [time] per week and these priorities: [list]. Build a realistic weekly plan that protects deep work, includes one customer conversation, and leaves slack for the unexpected. Tell me what to drop if the week goes wrong.

A NOTE ON JUDGMENT

AI can help you ask better questions, but it cannot tell you whether someone will use what you build. Treat every output as a draft, keep the customer conversation in the real world, and never share private information about another person without consent.

If one of these prompts helps you move forward, tell us which one. We will keep adding resources that are useful, specific, and honest.