Instant MVP

Turn a rough idea into a scoped, tested first version, one question at a time. By Eric Fondren

What it does

You bring an idea for an app, product or service. Your AI assistant asks you plain questions, one at a time, and turns your answers into a plan for the smallest first version worth building. Then it helps you get that version built and checks it with you. You do not need to know product terms or the order of the work. The assistant leads; you answer in your own words and approve each part.

When you finish, you have your idea written down the way you said it, a clear first version with what is left out, a read on whether the problem is real, a definition of success, the words to explain it, a build spec, a test checklist, and a working first version checked against that list.

An example. Priya walks dogs in her neighborhood and takes bookings by text message. Her idea: "An app where my clients book a walk and pay in one step." Instant MVP asks who books, how they do it today, and what the one step is. It settles on a first version where a client picks a time slot and pays, and Priya sees the day's walks. Reminders, reviews and a second walker go on the "later" list. The research plan points her to reviews of booking apps that pet owners already use. The build ends with a short checklist she runs herself: "Book a walk for Tuesday at 10. Does it show on your list?"

What you need

- **An AI assistant.** It works in Claude Code, Codex and Claude.ai as an installed skill. It also works in ChatGPT and other chat assistants through the all-in-one file.
- **Your idea and some time.** The planning stages take one or two hours across a few sittings. The build takes longer and depends on who builds it.
- **For the build, one of these:** an assistant that can write and run code (Claude Code or Codex can build it for you), an AI app builder, or a developer. The assistant asks which

one you have.

No accounts or fees come with the skill itself.

How to start

1. Download the skill zip from ericfondren.com.
2. Give it to your assistant and say: "Set up this skill using its README." The assistant installs it and tells you how to start it.
3. Start it. In Claude Code, type `/instant-mvp`. In Codex, mention `$instant-mvp`. Anywhere else, say "Help me turn my idea into a first version."

In Claude.ai, upload the zip in the Skills section of your settings. **In ChatGPT or another chat assistant,** download the all-in-one file instead, paste or attach it to a new chat, and say what your idea is.

What to expect

The assistant opens by saying it will ask a few short questions and that you approve each part as you go. Then it asks about your idea one question at a time: what it is, what problem it solves, whether you have the problem yourself, who would use it, how they handle it today, whether you already do it by hand, the one feature you would build first, and the simplest version (one thing a person does, one thing they get back).

If your idea is large, it says once that it will cost more and take longer, and asks whether the first version could start with one kind of user and one feature. You decide.

The nine stages

After the opening questions, it runs nine stages in order. It never asks you to pick one.

1. **Idea snapshot.** Your idea written down as you said it, with no polishing, plus a list of missing details. It then asks about each gap, one at a time.
2. **First-version scope.** What is in, what is out, the one flow from start to finish, the data it keeps and the screens it needs.
3. **Research brief.** A map of what to find out about your customers and where to look.
4. **Market check.** Competitors, other ways people solve the problem today, signs of demand, price ranges and risks. It does not change your scope.
5. **Success criteria.** A check that the scope still matches what you meant, and a short, concrete definition of success for the first 6 to 12 weeks.

6. **Messaging kit.** A one-line promise, answers to likely objections, ways to position it, a homepage outline and a pitch deck outline.
7. **Build spec.** One exact description of what to build and a build prompt any builder can follow. If something is unclear, it stops and asks you instead of guessing.
8. **Acceptance tests.** Detailed tests for the builder and a plain checklist for you.
9. **Guided build.** The build itself, one piece at a time, then the checklist with you, item by item, until every item passes.

Approving each stage

Each stage ends with a short summary in plain words, then the full saved record. Some records are in a structured format called JSON; that is the saved copy of the same thing, and you do not need to read it. The assistant asks whether the summary matches what you mean and waits. Corrections rerun that stage. Approved stages stay fixed. If you change your mind about one, it goes back to that stage and redoes every stage after it.

The research pause

After you approve the research brief, you choose: pause, gather real evidence, and come back with it; or continue with what you have. Evidence means reviews of competing products, forum posts and conversations with people who might use it. If your assistant can search the web, it offers to look for public reviews and posts and shows where each one came from. It never presents a guess as evidence. The market check is stronger with real quotes.

Before the build

When the planning is done, the assistant asks whether you want to build now or stop with the plan. It does not ask you to choose technology you do not know; if you have no preference, common defaults apply. Then it asks who will build it:

- **It builds it.** If it can write and run code, it builds one piece at a time and shows you how to try each piece.
- **An AI app builder.** It gives you the build prompt to paste in, then compares what the builder made against the spec and writes fix requests for anything missing or extra.
- **A developer.** It gives you a handoff package and a short cover note.

In every case it walks you through the checklist one item at a time and records what passed and what failed. New ideas go on a "later" list. At the end it reads back your success definition and says what to watch over the coming weeks.

Tips and limits

Good at: keeping a first version small, keeping your idea in your own words, catching gaps before you pay to build, and giving a builder a spec they can follow without guessing.

It won't: fill in answers for you, polish or change your idea, add features to the build, mark a test as passed without running it, or treat a guess as research. If your answers show no real person with the problem, it says so and suggests talking to a few possible users first. You decide whether to continue.

Tips

- Describe one real person who would use it. "Busy parents" is vague; "a parent of two who packs lunches at 6 a.m." is clear.
- Short answers are fine. If an answer is vague, it asks once more with an example.
- Save each approved result. Claude Code and Codex save them as files; in a chat assistant, copy each one into a document before you move on.
- Start a new folder for each idea, so the saved files stay together.

If it stalls or loses track

- Long chats can lose earlier details. Start a new chat, start the skill again, paste the results you saved, and it picks up at the first unfinished stage.
- If it skips ahead or asks several questions at once, say "One question at a time, and wait for my approval before the next stage."
- If the build spec stops with open questions, answer them one at a time. It reruns the spec with your answers.
- If you want the planning stages without an installed skill, the Instant MVP prompt pack PDF has the first six stages as copy-and-paste prompts.