



AI TOOL DROP

The App Builder Prompt Pack

Spec, build, debug, and launch a real product without developers.

aitooldrop.com

Tested, not listed. · © 2026 AI Tool Drop · Free resource, share the link, not the file.

The App Builder Prompt Pack

Ship a real web product without hiring developers. This is the prompt chain for the Lovable + Claude + Supabase stack: spec it, build it, debug it, wire the database, take payment. Run them in order.

1 · The Product Spec

Run this in Claude BEFORE you touch Lovable. A vague prompt builds a vague app; this turns your idea into a spec a builder can execute.

I want to build: [one sentence, e.g. "a tool that turns podcast episodes into show notes"].

Who pays for it: [who + why they care].

Turn this into a build spec:

- 1) The ONE core action a user takes on day one (ruthlessly cut everything else)
 - 2) The 3 screens that action needs, with what is on each screen
 - 3) The data we store, as a simple table list
 - 4) What we charge and where the paywall sits
 - 5) Everything we are NOT building in v1, listed explicitly
- Keep it under a page. I will paste this into an AI app builder.

TIP, The "NOT building" list is the most valuable section. Scope creep kills more first apps than bad code does.

2 · The Lovable Build Prompt

Paste this into Lovable, filling from your spec.

Build a web app: [core action from spec].

Screens: [the 3 screens].

Design: clean, modern, [your accent color], generous white space, mobile-first.

Data: connect Supabase with tables for [table list].

Auth: email and password signup, required before [the core action].

Start with the [first screen] and make the core flow work end to end before any polish.

TIP, One flow end to end beats five half-built screens. Ask for polish only after the core action works.

3 · The Debugger

When something breaks, and it will, paste this into Claude.

My app built in [Lovable/React] throws this error:
[paste the FULL error, not a summary]

Here is the code around it:
[paste the component or function]

Explain what is wrong in one paragraph a non-developer can follow, then give me the exact corrected code to paste back, marking what changed with comments.

4 · The Schema Designer

Before your data outgrows the prototype, get the database right.

My app does: [core action]. Current data: [what you store].
Design the Supabase schema: tables, columns with types, relationships, and which columns need indexes.
Then write the SQL to create it, plus row level security policies so users can only read and write their own rows. Explain each policy in one line.

TIP, Ask for row level security every single time. It is the difference between a demo and a product you can charge for.

5 · The Launch Wiring

My app is working. Walk me through wiring Stripe checkout for [one-time purchase / monthly subscription] at \$[price]:
1) The exact Stripe products and prices to create
2) Where the paywall check goes in my app
3) The webhook I need and what it updates in Supabase
4) A test checklist before I share the link publicly
Assume I have never used Stripe. Name every button I need to click.

Ship It

The full stack with costs and the build order: aitooldrop.com/stacks, and the App Builders category in the directory reviews every tool in this pack.