

The Full-App Builder Playbook

The 4-step system to build a deployable app with Claude Code —
auth, payments and dashboard, no developers needed.

01

Have Claude Chat Write Your CLAUDE.md

Don't prompt a coding agent off the top of your head — make Claude Chat build the spec first. Describe your app idea in a few sentences, then paste the prompt below. It interviews you one question at a time, then outputs a complete CLAUDE.md master file: product overview, tech stack, data model, v1 feature list, and build rules.

PASTE THIS INTO CLAUDE CHAT

```
Act as a senior product engineer. Interview me about the app I just described – target users, core features, data, payments, admin needs – one question at a time. Then output a CLAUDE.md file for Claude Code with: 1) product overview, 2) tech stack with one-line justifications, 3) data model, 4) features split into v1 vs v2, 5) auth + payment requirements, 6) build rules: no mock data, verify each feature by running the app, smallest diff per step, stop for review after each phase.
```

💡 QUICK TIPS

- The build rules section is what separates a deployable product from a demo — never cut it.
- Save the output as CLAUDE.md in an empty project folder; Claude Code reads it automatically every session.

Build in Phases, Never in One Shot

Open Claude Code in the folder with your CLAUDE.md and start the phased build. One giant 'build my app' request produces something that looks finished and doesn't run — phases with review stops produce working software.

PASTE THIS INTO CLAUDE CODE

```
Read CLAUDE.md. Build v1 in phases, stopping for my review after each: Phase 1 – repo scaffold, database schema, user authentication. Phase 2 – the core feature end to end with real data. Phase 3 – payment processing with a test-mode checkout I can click. Phase 4 – admin dashboard and analytics. Follow the build rules in CLAUDE.md at every step.
```

💡 QUICK TIPS

- Don't accept 'done' until you've watched the flow run — make the agent prove each phase.
- Run /clear between phases and let Claude Code re-read CLAUDE.md; long sessions drift.

Run the Error-Fix Loop

When something breaks, paste the full error message back into Claude Code verbatim and let it self-correct. The agent reads the stack trace, finds the file, fixes the cause, and re-runs the app. This loop is the entire debugging workflow — no developer required.

💡 QUICK TIPS

- Paste the actual traceback, never a summary — 'it's broken' gets a guess, the real error gets a fix.
- If the same error survives two fixes, ask the agent to explain the root cause in plain English before touching code again.

Verify Auth and Payments Yourself

Before anything goes live, click through the money paths with your own hands: sign up as a new user, log out and back in, and complete a test-mode checkout. Claude Code wires standard providers — Supabase-style auth and database, Stripe-style payments — but you are the founder, and the founder verifies revenue paths personally.

💡 QUICK TIPS

- Keep payments in test mode until signup, login, and checkout all pass in one clean run.
- Check the database after each test — a real record saved is the only proof a flow works.

Expand With .md Master Prompts

Once v1 runs, every new feature gets its own focused .md spec written the same way: admin dashboard, analytics, live chat, leaderboards. Twenty minutes of writing per spec replaces what used to be a developer invoice — hand each file to Claude Code and direct the build like a founder.

💡 QUICK TIPS

- One spec per feature keeps builds small, reviewable, and easy to roll back.
- Ship the retention layer early — analytics and leaderboards tell you what users actually do.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>