

The Ship-Your-App Cheat Sheet

Your 5-Step System to Build, Deploy & Get Paid in Under 60 Seconds

01

Build It with Claude Code

Claude Code is Anthropic's terminal-based AI coding agent that builds your entire app autonomously. You describe the app in plain English and it writes the files, runs the commands, and fixes its own errors. Your job is to direct and review, not to type every line.

QUICK TIPS

- Start with a clear idea and a screenshot of what you want.
- Let it scaffold the whole project first, then iterate feature by feature.
- Review each change before moving on — you stay the director.

02

Run 8 Agents at Once with cmux

cmux is a free tool that lets you run and manage up to eight Claude Code agents in parallel. Instead of waiting on a single thread, you can build several features — or several apps — at the same time. Think of it as mission control for your AI builders.

QUICK TIPS

- Use it to split one big build into parallel feature branches.
- It's free — no reason not to run more than one agent.

Plan 80% Upfront with the Master Prompt

Screenshot your app idea, hand the image to Claude, and paste the prompt below. It forces Claude to ask before it assumes, cut your idea down to the smallest version that actually works, and wait for your go before writing a line — then hold itself to a four-point check before it can call the build finished.

PASTE THIS INTO CLAUDE CODE

I want to build: [YOUR APP IDEA]

Before you write any code:

1. Ask me up to 3 questions, only the ones you truly can't guess.
2. List the core features, then pick the smallest version that still works. Everything else goes on a "later" list.
3. Tell me the exact tools you'll use and why, in one line each.
4. Stop and wait for me to say "go."

Then build Phase 1 only. Ugly is fine. Working is the whole point. This is the food stand, not the restaurant: one thing, done well.

Before you tell me it's finished, check all four yourself:

1. It runs, no red errors.
2. Every button and form does what it says.
3. Real data, not fake placeholder data. Show me the actual output.
4. Bad input shows a message instead of crashing.

If one fails, fix it and re-check all four. Don't hand me broken work.

How to talk to me:

- I'm not a coder. Plain English. Define any technical word in one short line.
- When you need something from me, give me numbered steps: exactly what to click, type, or paste, and where.
- Put API keys in .env only. Never in the code.
- Save my progress to git the moment Phase 1 works. That's my undo button.

End every message with:

WHAT I DID: 2 bullets

YOUR TURN: numbered steps, or "nothing, say go"



QUICK TIPS

- Always give Claude a visual reference — a competitor, a sketch, or a Figma frame.
- Approve the plan before any code is written.

Back Everything Up to GitHub

GitHub is your free cloud filing cabinet for code. Push your project so nothing is ever lost and you have full version history. It also gives Railway a source to deploy from in the next step.



QUICK TIPS

- Commit early and often so you can roll back any mistake.
- A private repo keeps your build yours until you're ready to share.

Deploy on Railway & Plug in Stripe

Railway takes your GitHub repo and puts your app online with a live custom domain in a single deploy — no server config or DevOps. Connect the repo, let Railway build and host it, point your domain at the service, then drop in Stripe checkout so the app can take payments. That's the full loop: build, deploy, get paid.

💡 QUICK TIPS

- Connect your GitHub repo directly so every push redeploys automatically.
- Add a custom domain in one click for a professional link.
- Plug in Stripe last — it only charges a fee when you make a sale.

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>