

# Your First App: What You Actually Need

---

*For the person who keeps almost starting.*

I was a bartender for 10 years. Three kids. I was pregnant with my second when I went back to school — working doubles, CS classes at night — because I googled "what career makes the most money and matches my personality" and computer science came up. Not joking.

Now I'm a Lead Engineer building my own products in my spare minutes or after the kids go to sleep. Warm tea. Four repos open (currently)

The thing that changed wasn't my situation — it never got calmer. I stopped waiting for it to. You figure out pretty fast that "when things slow down" or "when I free up" isn't a real date on the calendar.

This is six pages of what I wish someone had handed me. The checklist. The prompts. The template I use on every single project. The mistakes I made so you don't have to. By the time you finish, you'll have enough to start.

01 · Setup Checklist

02 · Prompts That Work

03 · CLAUDE.md Template

04 · Commands

05 · VS Code Shortcuts

06 · What's Next

# Before You Write a Single Line of Code

10 things that actually matter. Not 40. These 10.

- A laptop — any laptop.**  
Mac not required. A \$300 refurbished machine works.
- VS Code installed.**  
code.visualstudio.com. Free. Claude Code runs inside it.
- Node.js installed.**  
nodejs.org. Don't need to understand it yet — it's the engine the car runs on.
- A terminal you're not scared of.**  
Open it, type one thing, press enter. That's the skill to start.
- A Claude account.**  
claude.ai. Free to start. \$20/mo if you're building daily.
- An idea — even a rough one.**  
"Track what my kids eat" is a real app. "Build something with AI" is not an idea yet.
- 30 minutes of focus.**  
Thirty minutes with the other tabs actually closed.
- Somewhere to write things down.**  
Notes, Notion, a doc. A place to dump thoughts.
- Willingness to read error messages.**  
Half of building is reading what broke. That's just programming.
- Permission to start ugly.**  
Their first version looked worse. They shipped it anyway.

If you already have all 10, open VS Code right now. Don't finish the PDF first.

## WHAT YOU DON'T NEED

No CS degree. No bootcamp certificate. No portfolio, no perfect idea, no cofounders, no VC, no Figma wireframes, no one's permission. I had none of that when I started. None.



Already have 3 of these checked? That's more than I had when I started.

# How to Actually Talk to Claude Code

How you prompt Claude is how you get results. There's no magic — Claude doesn't read your mind, it reads your words. Most beginners write vague prompts, get vague results, and blame the tool. The tool's fine. The prompt needed work.

## CONTEXT

What the project is. What's already built. What NOT to touch.

## TASK

Exactly what to build or fix — specific, not vague.

## CONSTRAINTS

Framework, format, expected output, edge cases.

## BEFORE YOU SEND — ASK YOURSELF

- Did I give enough context about what already exists?
- Did I say what I don't want Claude to touch?
- Did I describe what "done" actually looks like?

## SAVE YOUR FINGERS

Don't feel like typing a long prompt? Don't. Use voice input and just talk.

 Win + H Windows voice typing ·  Fn Fn Mac dictation

TASK	✗ BAD	✓ GOOD
Building a page	"Build me a landing page"	"Build a landing page for a meal tracker. Hero, headline, 3 features, sign-up button. Tailwind CSS. No animations."
Adding a feature	"Add a login"	"Add email/password login with Supabase Auth. Already set up. Don't touch the existing dashboard — only login form and redirect."
Fixing a bug	"It's not working"	<code>TypeError: Cannot read properties of undefined (reading 'map')</code> in RecipeList on load. [paste code]. Think data isn't loaded yet — how do I fix it?
Database	"Set up a database"	"Set up a Supabase table 'entries': id (uuid pk), user_id (uuid refs auth.users), content (text), created_at (timestamp default now). Show me how to insert a row from Next.js."

**Specific + context + expected output = Claude actually helps you.**

*Vague in, vague out. If you wouldn't hand this to a contractor and expect them to figure it out — don't send it to Claude.*

# Your Secret Weapon: The CLAUDE.md File

## WHAT IT IS

A plain text file you put in your project root. Every time Claude Code opens your project, it reads this file first — so you set your stack, your rules, your current goal once and Claude carries it. You stop repeating yourself.

## WHY IT MATTERS

I set this up before I write a single line on every project. No more re-explaining the setup. No more Claude suggesting a framework you already said you don't want. One file and the whole session is different.

```
CLAUDE.md

# Project Name    → Meal Tracker App

# Description     → Simple web app to log meals and see a weekly summary.
                  Solo project, early build.

# Tech Stack     → Next.js 14 (App Router), Tailwind CSS,
                  Supabase (PostgreSQL + Auth), Vercel

# What To NEVER Do → No Pages Router. No new packages without asking.
                  No empty TODOs. No TypeScript changes without asking.

# File Structure  → /app (pages/layouts), /components (under 100 lines),
                  /lib (utils + Supabase client)

# Code Style     → Readable over clever. Functional components.
                  async/await. Ask before building.

# Current Goal   → Build and test the daily entry form.
                  Logic first - styling later.
```

Update this file when you add a new package, finish a major feature, or change your current goal. Keep "Current Goal" current — it makes a real difference.

# Save This Page

## TERMINAL COMMANDS

`$ npm run dev`  
Starts your server at localhost:3000. Type this every time you sit down.

`$ git add . && git commit -m "msg"`  
Saves your work. Your undo button — do it after every session.

`$ git push`  
Sends to GitHub. On Vercel, also triggers a new deployment.

`$ npm install [package-name]`  
Adds a library. When Claude says install something, this is the command.

`$ npm run build`  
Catches errors before deploying. Run before every push to production.

`$ git status`  
Shows what files changed and what's staged. Run before committing.

`$ git log --oneline`  
See recent commits in a clean list. Find when something broke.

`$ node -v && npm -v`  
Confirms Node and npm are installed and shows your version.

## VS CODE SHORTCUTS

`Ctrl + P` Quick Open — find any file by name. No clicking through folders.

`Ctrl + Shift + F` Search All Files — find any word or code across your entire project.

`Ctrl + Shift + P` Command Palette — find any VS Code command by typing it.

`Ctrl + `` Integrated Terminal — open/close terminal inside VS Code.

`Ctrl + /` Toggle Comment — comment or uncomment a line instantly.

`Alt + Shift + F` Format Document — auto-format and clean up your entire file.

`F2` Rename Symbol — rename a variable everywhere in the file at once.

`Ctrl + K + S` Keyboard Shortcuts Editor — see and

## 5 MISTAKES BEGINNERS MAKE

- 1 Copying code without reading it.**  
Paste, it works, then it breaks and you don't know why.  
**Fix:** read it. Ask Claude to explain it line by line.
- 2 Building without version control.**  
Can't go back when things break. **Fix:** Git day one. Commit every session.
- 3 Skipping the CLAUDE.md.**  
Keep getting wrong suggestions. **Fix:** write the file — page 4. Now.
- 4 Building everything before testing anything.**  
300 lines, nothing works. **Fix:** form renders → submits → saves. One step.
- 5 Quitting on the first red error.**  
Errors are messages. **Fix:** paste to Claude with context. You'll learn fast.

## IF CLAUDE GETS IT WRONG

### WRONG QUESTION ANSWERED

"That's not right. Here's what I was actually trying to do: [explain]. Can you try again?"

### COMPLEX TASK

"Before you write any code — walk me through your plan first. I want to make sure we're aligned."

### CHANGE BROKE SOMETHING

"Error: [paste]. Code you gave me: [paste]. What went wrong and how do we fix it?"

# You Have Everything You Need to Start

I'm not going to tell you it's easy. Some days you're going to stare at an error message for an hour and feel like nothing is working. That's normal — that's what building looks like from the inside. The part nobody shows you is that an hour later it clicks, and you understand something you didn't before.

You have what you need. The checklist is right here. The template is ready to copy. The prompts are written out. Just open the terminal.

## WHEN YOU'RE STUCK — 3 STEPS

1. Read the error out loud. The full message. It usually tells you exactly what broke.
2. Google the error exactly as written. Someone has hit this before.
3. Paste it to Claude with context: what you were trying to do + what you expected.

## COMMON FIRST PROJECTS — PICK ONE AND START

[expense tracker](#)[habit log](#)[recipe saver](#)[link collector](#)[daily journal](#)

### Claude Code Kickstart — \$27

[yessieperez.com/courses](https://yessieperez.com/courses)

Six short modules. Zero to a working app. Built for people who learn by doing.

### Build con Chispa — free

[skool.com/build-con-chispa-9733](https://skool.com/build-con-chispa-9733)

A community for builders also running the rest of their life. Cafeito hours every week.

**Okay. Go build.**

