START CODING WITH AI

One page. Opinionated. Skip the expensive mistakes.

1 · PICK ONE TOOL, STOP SHOPPING

Use Claude Code: go to claude.com/download, subscribe to Pro ($20/mo), and open the app (there's no terminal involved). It edits files in a folder on your computer, asks before each change, and plans before it builds. Already paying for ChatGPT Plus? **Codex is included** (chatgpt.com/codex), use that instead. Either works. Switching tools every week is how you learn nothing.

2 · BEFORE YOUR FIRST PROMPT

You don't set anything up by hand. The agent does its own setup. Your first prompt, verbatim:

```
Set up git in this folder and commit everything.
Create your instruction file with these rules:
- Small changes. Explain each in plain English.
- Ask before installing anything new.
- Commit after every change that works.
Put .env and all secret files in .gitignore.
```

- **Secrets:** keys live in .env, never in code or chats. Keep the repo private. A leaked key gets rotated the same day; deleting it doesn't un-leak it.
- **Leave approval mode on.** If a tutorial says to turn on "YOLO mode," close the tab.
- **No add-ons.** Skip MCP servers, skills, and plugins from blogs or videos. The stock tool is enough for your first six months.

3 · THE LOOP

`Plan → smallest change → **read the diff** → prove it works → commit → repeat.`

The diff is the red/green before-and-after the tool shows; never accept one you don't understand. Make the agent prove its work: "run it and show me the output." Fails twice at the same fix? Don't argue. Start a new conversation with a better prompt.

4 · THE FOUR KILLERS

1. **Shipping code you can't explain.** If you can't say what a change does, don't accept it. Ask: "explain this change in plain English." Real companies have shipped real breaches this way.
2. **No undo.** Agents edit dozens of files in one burst and can't reliably un-edit them. Git can: it's your undo button for code. Commit every time something works.
3. **The endless session.** Output quality rots as a session grows. New task, new conversation. Two failed corrections, new conversation.
4. **Real systems, real damage.** Never hand the agent production credentials, customer data, or a live database. Git undoes code, not sent emails or dropped tables. Before real users touch your app, ask: "how could a stranger read someone else's data here? Show me where that's blocked."

5 · MAGIC WORDS

Agents know the discipline; these phrases switch it on.

- `"Plan first. Don't write any code yet."`
- `"Ask me clarifying questions before you start."`
- `"Make the smallest change that works."`
- `"No features I didn't ask for."`
- `"Write a failing test first, then make it pass."`
- `"Fix the root cause. Don't suppress the error."`
- `"Don't refactor code unrelated to this task."`

6 · LEARN WHAT YOU DON'T KNOW

Use a chat session (Claude or ChatGPT) as your translator:

- `"What terms describe this problem? Teach me the vocabulary."`
- `"What would a senior engineer ask about this plan?"`
- `In a fresh conversation: "What's wrong with this plan?" (paste plans, never keys or customer data)`

One rule of thumb: phrases for intent, the instruction file for conventions, and the tool's permission settings for anything containing "never" or "always." Ask the agent: "block yourself permanently from touching X."

Want a guide at the keyboard? Our first session is three hours, free:
hello@lgtm.systems