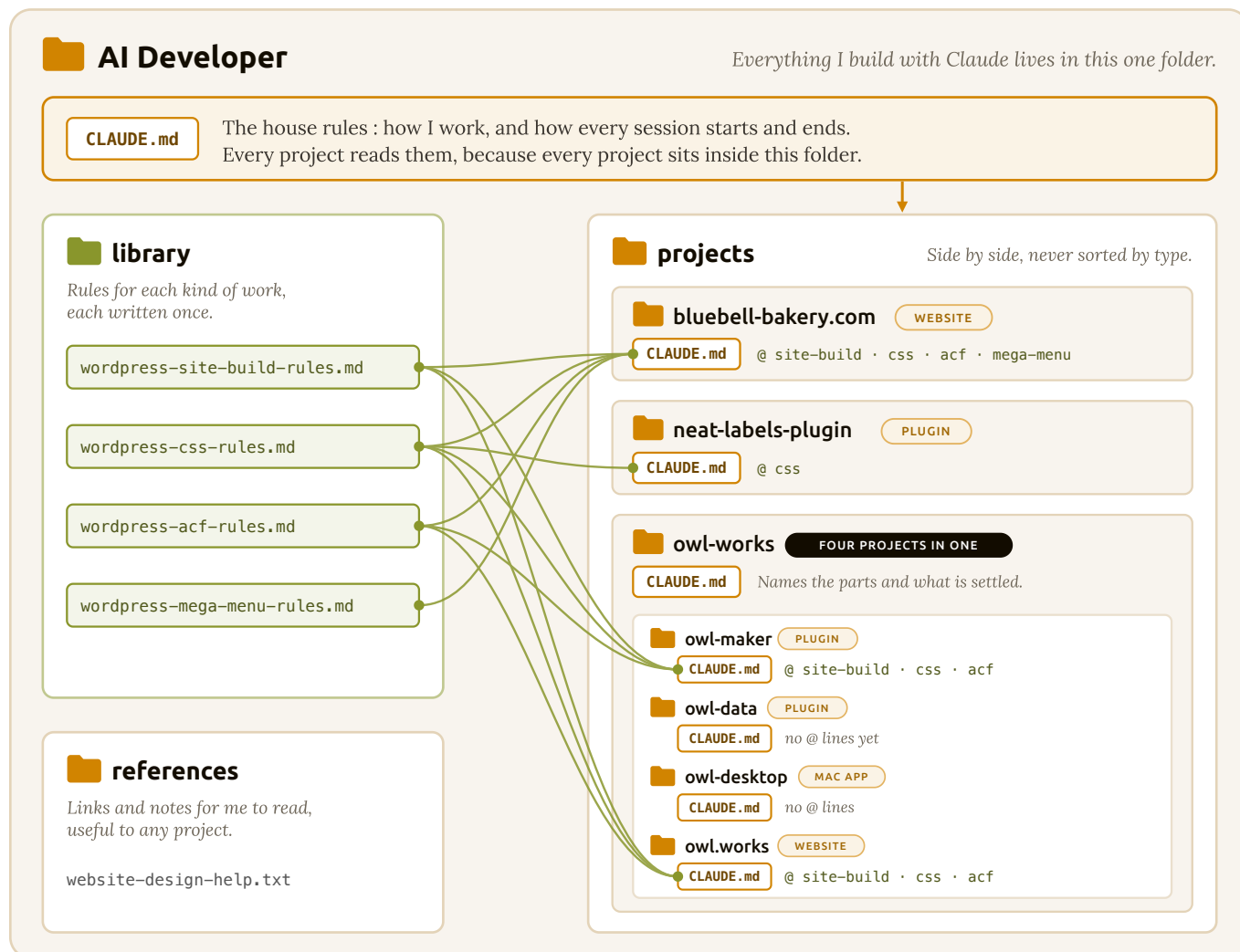


Folder structures for Claude Code

One library, and every project side by side

How I lay out the folder where I work with Claude Code. Every rule is written once and kept in a library. Every project sits side by side in one folder, and its CLAUDE.md pulls in only the rules it needs.

Only CLAUDE.md is standard. Every other file is my own invention, written for the way I work. They may not suit you, even if you build WordPress sites too, so copy the layout and write rules that fit your own work.



The house rules

Claude Code reads the CLAUDE.md in the folder you start it in and in every folder above. Rules kept at the top of AI Developer reach every project inside it.

The library

A line in a CLAUDE.md made of @ and a path pulls in a library file. Each project imports only the rules it needs, and a rule changed once reaches every project that imports it.

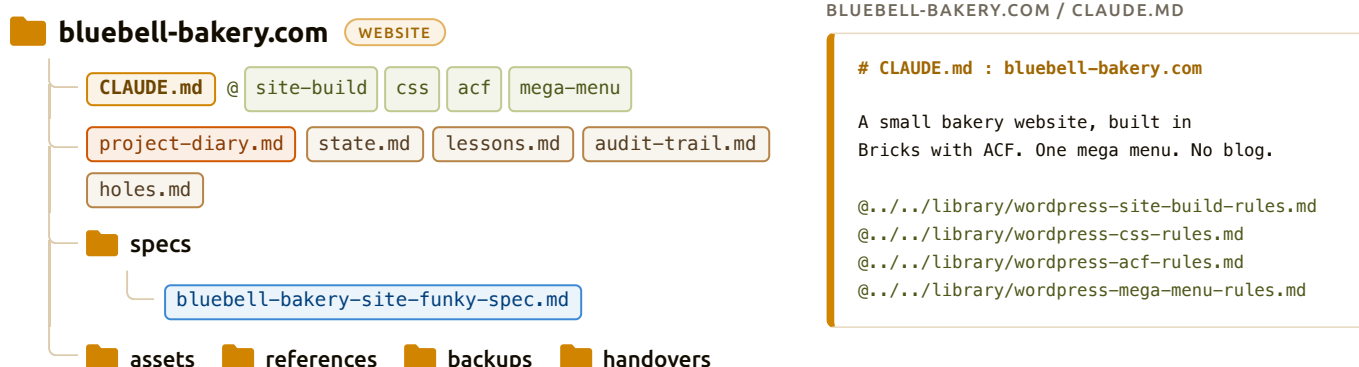
One depth

Every project sits at the same depth, so their @ lines read the same : @../.. /library/. Each part of owl-works sits one folder deeper and needs one more ../.

Inside a project

Every project is laid out the same way

The same notes, the same folders. What changes is the **CLAUDE.md** : what the project is, and which library files it imports.



What each file is for

CLAUDE.md is the only standard file : Claude Code looks for it by itself. Every other file here is my own, and Claude reads it only because a CLAUDE.md tells it to.

CLAUDE.md	The standard file. What the project is, anything only it needs, and its @ lines.	audit-trail.md	A dated record of everything significant that happened.
project-diary.md	My concerns and wishes. I write it, and Claude reads it first.	holes.md	Gaps found in the specs, each put to me as a question. Wherever there are specs.
state.md	Where things stand now.	funky specs	What to build, agreed before building : a map of the whole project, then one spec per module.
lessons.md	Short, dated lessons, so a mistake is only made once.	folders	specs, assets, references, backups and handovers. A plugin adds its code and can add tests.

A session. At the start, Claude reads the diary and asks me about it. When I say "close", it proposes new lines for lessons.md and audit-trail.md and a new state.md, and writes nothing until I say yes.

Before you copy this

- The path starts at the CLAUDE.md**, not at the folder where you start Claude. Move a project one folder deeper and each @ line needs another `../`.
- Approve the imports when asked.** If you decline, Claude Code won't load them in that project and won't ask again. Type `/context` to see what a session has loaded.
- Imports load in full** at the start of every session. Keep library files lean, and put long rules you need only now and then in a skill, which loads only when it's used.
- A space in a folder name** needs a backslash in an @ line : `@../My\ Rules/css-rules.md`. A path in quotes isn't imported.
- These instructions are for Claude Code.**

The project names are invented. The files and rules are my own, made for the way I work, and may not suit yours. Every statement about Claude Code was checked against its documentation in October 2026.