

Build Roadmap and Checklist

The full journey from empty folder to working AI POS. Follow the steps in order — the sequence matters. Expect many hours in total across a few sessions; don't try to do it in one sitting.

How the package fits together

The numbered folders are groupings, not steps. You'll move between folders as you build; every step below names the exact file and folder, so you always know where to reach.

- **0 - Start Here** — read first. Concepts and this roadmap.
- **1 - Copy this onto your C drive** — a ready-made empty AI POS folder; copy it to your C drive and rename it to `AI Access`. The easy way to do the Phase 1 folder step.
- **2 - Build Guides** — the guides you work through in conversation with AI.
- **3 - Templates** — the blank files your conversations fill in.
- **4 - Mo's AI Access** — a complete worked example. Skim it before you start; check your work against it as you build.

This roadmap is the only sequence. Follow it top to bottom — everything else in the package is a toolbox it reaches into.

Phase 1 — The Foundation (the longest phase — spread it across sessions; do these in order)

- [] **Read the concepts** — files 1 and 2 in this Start Here folder (15+ min)
- [] **Skim Mo's AI Access** — `4 - Mo's AI Access` — see what the destination looks like; then read `2 - Build Guides/what_a_build_conversation_looks_like.pdf` so you know how the building actually works (15 min)
- [] **Create your folder structure** — the easy way: copy the `1 - Copy this onto your C drive` folder onto your C drive (your user folder — `C:\Users\YourName\` — is a good home), then rename it to `AI Access` (the recommended name; or personalise it like Mo's "Mo's AI Access"). The folders inside are already named correctly, so your paths will match the boot file. (2 min) — **or** build it by hand: open File Explorer (Mac: Finder) and recreate the folder map in `3 - Templates/claude_md_boot_template.md` — same names, same nesting; the entries without a trailing slash (`CLAUDE.md` , `AGENTS.md` , `_project_template.md`) are files, not folders, so skip them here — they come in later steps. (5+ min) — **or**, if your AI tool is already connected to a folder, just ask it to build the structure for you.

- [] **Identity** — copy the kick-off prompt and everything below it from `3 - Templates/identity_template.md`, paste into your AI, and let it interview you — this creates your `identity.md` file (30+ min)
- [] **Voice standards** — open a new chat and attach 3–5 pieces of your writing (using the paperclip or + button in the message box), then copy the kick-off prompt and everything below it from `3 - Templates/voice_standards_template.md` — this creates your `voice_standards.md` file (45+ min — *the longest step, and the one that most repays the effort*)
- [] **Rules** — copy the kick-off prompt and everything below it from `2 - Build Guides/rules_builder.md`, paste into your AI, and work the questions — this creates your `rules.md` file (30+ min)

Phase 2 — Switch It On (one short session)

- [] **Boot file** — this step is different: no interview, no conversation — just a short fill-in. Open `3 - Templates/claude_md_boot_template.md` and follow the numbered instructions at its top. The result is your `CLAUDE.md` file, saved in the top level of your AI Access folder (10+ min)
- [] **(Optional) AGENTS.md companion** — the boot-file template's final step also saves a short `AGENTS.md` pointer next to `CLAUDE.md`, so non-Claude agent tools (Codex, Copilot, Gemini CLI) can find your system too. Claude-only? Skip it. (2 min)
- [] **Personal instructions** — follow `2 - Build Guides/personal_instructions_guide.md`; paste the result into your AI tool's settings (15+ min)
- [] **Test it** — open a fresh session with your folder connected (see "Connecting your folder" in Before You Start) and ask: "What do you know about me?" If the answer sounds like you, the system is live. (No folder access on your plan? Your version of this step: upload your Layer 1 files to a project and run the same test — see "What you need" in Before You Start.)

Checkpoint: Layer 1 — Identity & Context of your AI POS is now working. Everything after this point is compounding improvement — add it as need arises, not all at once.

Phase 3 — The Engine: Layer 2 — Skills & Rhythms (build over your first weeks of use)

- [] **First reference file** — `2 - Build Guides/reference_files_discovery_guide.md`; start with the output type you produce most
- [] **First skill** — `2 - Build Guides/skills_guide.md`; wait until you've done a task twice and wished you didn't have to re-explain it
- [] **First prompt template** — `2 - Build Guides/prompt_templates_guide.md`; for your highest-stakes recurring conversation
- [] **First scheduled task** — `2 - Build Guides/scheduled_tasks_guide.md`; a Monday morning setup is the classic starting point

Phase 4 — The Live Layer: Layer 3 — Active Context

Layer 3 is deliberately the smallest layer — and that's a design choice, not an oversight. Your live context (what's happening right now) doesn't live in files that go stale; it lives in your Projects folder and comes fresh from your connected tools.

- [] **Mark the design choice** — copy `3 - Templates/active_context_readme.md` into your `Layer 3 - Active Context` folder as `README.md` (folder connected? just ask: *"Copy the active context readme into my Layer 3 folder."*). The README records why the folder is nearly empty, so future-you — and the AI — know it's intentional
- [] **First project** — say *"I want to set up a new project for X"*; your rules file handles the rest. (Built your folders by hand rather than from the starter folder? First copy `3 - Templates/project_template.md` into your Projects folder as `_project_template.md` — the setup routine uses it.) Projects are where active context actually lives: one `_context.md` per piece of work, read by the AI whenever you mention that project

Phase 5 — Keep It Alive: Layer 4 — Navigation

- [] **Quick Reference Card** — don't build this by hand; it's the AI's job. Once you have more than a few components, say: *"Scan my AI POS — every skill, prompt template, scheduled task, reference file, and project — and build my Quick Reference Card from the template in 3 - Templates, saved to Layer 4 - Navigation."* It stays current the same way: every guide in this package already ends with "add it to my Quick Reference Card"
 - [] **Maintenance** — one ask does the whole step: *"Copy the maintenance guide from 3 - Templates into my Layer 4 - Navigation folder, and set up a quarterly scheduled task that runs its review checklist."* (No scheduler in your tool? A recurring calendar reminder every three months does the same job)
-

Your finished files — where everything ends up

By the end of this roadmap, these are the files your system contains and where each lives. This is a map, not a to-do list — the steps above create every one of them, and the AI handles the saving.

Finished file	Where it lives
<code>identity.md</code>	Layer 1 - Identity & Context
<code>voice_standards.md</code>	Layer 1 - Identity & Context
<code>rules.md</code>	Layer 1 - Identity & Context
<code>personal_instructions.md</code>	Layer 1 (reference copy — the live copy is in your AI tool's settings)
<code>CLAUDE.md</code>	Top level of your AI Access folder
<code>AGENTS.md</code> (optional — cross-tool pointer)	Top level of your AI Access folder
<code>[type]_standards.md</code> (e.g. <code>document_standards.md</code>)	Layer 2 / Reference Files
Skill files	Layer 2 / Skills
Prompt templates	Layer 2 / Prompt Templates
Scheduled task prompts	Layer 2 / Scheduled Tasks
<code>README.md</code> (copied as-is)	Layer 3 - Active Context
<code>quick_reference_card.md</code> (built by the AI)	Layer 4 - Navigation
<code>maintenance.md</code> (copied as-is)	Layer 4 - Navigation
<code>_project_template.md</code> (kept blank)	Projects folder
One <code>_context.md</code> per project	Projects/Active/[project name]

The one rule that matters most

Build Layer 1 properly before touching anything in Phase 3. A brilliant skill sitting on a generic identity produces generic output. The foundation is where the value comes from.