

YOUR AI STARTS EVERY SESSION FROM ZERO → HERE'S THE ONE-FOLDER FIX.

Your Second Brain

starts with one folder · for IT Pros · BA · architect · tester · DBA · support · tech writer.

GATHER YOUR CONTEXT pick 3-5 files you already work from · drop them in your folder, Project, or chat

<p>PROJECT CONTEXT Brief · PRD · roadmap · stakeholder deck · meeting notes · sprint retros.</p>	<p>DECISIONS & STATE Slack / Teams threads · email decisions · last week's status.</p>	<p>PEOPLE & PROCESS Stakeholders · tickets (Jira · Linear · Asana) · SLAs · scope docs.</p>
<p>TECHNICAL SHAPE Diagrams · schemas / data dictionary · API contracts · runbooks.</p>	<p>PERSONAL PATTERNS Your review rubric · status / email / meeting-prep templates.</p>	

RUN THE PROMPT three path-specific versions on the website · pick yours

<code>context-stack-builder.md</code>	path A · B · C → pickbits.ai/sb
"Act as my Context Stack builder."	One role, one job. Keeps the AI from drifting to generic-assistant mode.
"Read only the folder I share. Write only to context-stack/."	Safety contract. No surprise writes anywhere else on your machine.
"Scan silently. Then ask 3-5 short questions in one message."	Inventory before noise; clarification over fabrication.
"Mark uncertain spots [TBD]. Debrief what you skipped."	Visible gaps, not invented confidence. You can trust what's there.

YOUR SECOND BRAIN three files the prompt builds · everything else is bonus

01	IDENTITY.MD rarely changes	Who you are. How you want AI to answer. Your non-negotiables.	context-stack/
02	PROJECT-*.MD weekly	One brief per active project. Goals, status, decisions, stakeholders, deadlines.	context-stack/
03	STATE.MD daily	What you touched today. What's blocked. What's next. This week's working memory.	context-stack/
+	BONUSES grow over time	Wrappers (Path A): CLAUDE.md / AGENTS.md auto-load the three files · patterns/ ; prompt templates · artifacts: code/docs <i>linked</i> from project file (never copied)	project root

KEEP IT FRESH

<p>DAILY LOAD ~ 60 sec</p> <ol style="list-style-type: none"> Open your AI. Path A: wrapper auto-loads. Path B: Project remembers. Path C: drag the three files in. Ask: "load these, tell me where I left off." Work. At session end, update <code>state.md</code> with what happened. 	<p>MONDAY RESET ~ 10 min</p> <ol style="list-style-type: none"> Update each project file with last week's decisions. Archive last week's <code>state.md</code>. Start a fresh one. Prune <code>patterns/</code> – delete anything unused in 30 days. <p>Module 3 (Routines on Desktop) automates this.</p>
---	--

EVERY CROSS-TEAM CONVERSATION STARTS FROM ZERO → THE SHARED-FOLDER FIX.

The Team Brain

when nodes connect into a mesh · for IT Pros · BA · architect · tester · DBA · support · tech writer.

GATHER YOUR CONTEXT pick 5-8 things that, taken together, make a new hire useful in week one

IDENTITY & PROCESS Charter · RACI · org chart · cadence doc · retro notes.	PROJECT SHAPE PRD / spec · roadmap · OKRs · scope doc · milestones.	DECISIONS & STATE Standup notes · sprint reviews · existing decisions log · risk register.	STAKEHOLDERS & SYSTEMS Stakeholder map · integrations list · runbook · vendor contacts.
--	---	--	---

RUN THE PROMPT three path-specific versions on the website · pick yours · one steward runs it

<code>team-brain-builder.md</code>	path A · B · C → pickbits.ai/sb
"Act as my Team Brain builder."	One role, one job. Avoids generic-assistant drift.
"Write only inside team-brain/. Skip secrets, vendor pricing, NDA."	Safety contract. Confidential files surface in the debrief, not the output.
"Inventory silently. Then ask 4-6 short questions."	Inventory before noise; clarification over fabrication.
"decisions.md starts mostly empty. Append-only thereafter."	Memory, not a working doc. Steward grows it weekly.
"Complement, don't replace, each teammate's context-stack/."	Personal layer = how you answer me. Team layer = the team's facts.

YOUR TEAM BRAIN four files the prompt builds · everything else is bonus

01	TEAM-IDENTITY.MD rarely changes	Who the team is. How we decide. Cadence. Roles. Vocab. Non-negotiables. Onboarding pointer.	team-brain/
02	PROJECT-SHARED.MD weekly	The shared brief. Goal, stakeholders, sprint status, who-owns-what, deadlines.	team-brain/
03	STATE-SHARED.MD daily-ish	What shipped. What's blocked. What's next – as a team. (Module 3 automates this.)	team-brain/
04	DECISIONS.MD append-only	Date · context · decision · alternatives · review. Your team's institutional memory.	team-brain/
+	BONUSES grow over time	Wrappers (Path A): CLAUDE.md / AGENTS.md auto-load both layers · members/<name>.md: per-teammate context · <i>Onboarding section</i> in team-identity.md gets new hires up in 10 min	repo root

KEEP IT FRESH

DAILY LOAD ~ 60 sec <ol style="list-style-type: none"> Open AI. Path A: wrappers auto-load <code>team-brain/</code> , Path B: shared Project remembers. Path C: drag <code>state-shared.md</code> in. Ask: "load these, where did the team leave off?" Work. End-of-day, steward updates <code>state-shared.md</code> with what shipped / what's blocked. 	FRIDAY SYNC ~ 15 min <ol style="list-style-type: none"> Steward replaces last week's <code>state-shared.md</code> with this week's summary. Appends week's decisions to <code>decisions.md</code> – newest first. Context · choice · alternatives · review. Scans <code>project-shared.md</code> for staleness. Notifies team: "Team Brain updated – pull the latest." <p>Module 3 (Routines on Desktop) automates the daily refresh.</p>
---	---

YOUR STATE FILE WAS WRITTEN BY HAND → NOW IT WRITES ITSELF.

Routines on Desktop

automate the daily refresh · for IT Pros · BA · architect · tester · DBA · support · tech writer.

PLUG IN YOUR SOURCES

routines need read-access to where work happened yesterday · pick what your team uses

CODE & TICKETS GitHub / GitLab – yesterday's PRs · Jira / Linear – status changes · CI logs.	CONVERSATIONS Slack / Teams standup channel · meeting notes · email decisions thread.	TEAM BRAIN (WRITE TARGET) team-brain/state-shared.md · decisions.md · project-shared.md staleness.	NOTIFICATION SURFACE Slack DM to steward · email digest · routines-inbox/ folder.
---	---	--	---

RUN THE PROMPT three path-specific versions on the website · pick yours · steward owns the schedule

<code>routines-builder.md</code>	path A · B · C → pickbits.ai/sb
"Act as my Routines builder."	One role, one job. Drafts three routine specs in 30 minutes.
"Routines write to team-brain/ only – never back to source systems."	Drafts only. Steward promotes. No surprise Jira comments or Slack posts.
"Cap each prompt under 500 tokens. Schedules pay per-token."	Cost guardrail. A noisy nightly routine can run hot fast.
"Default silent. Notify the steward only when a draft NEEDS REVIEW."	Inbox doesn't fill with "all clear" pings. Only the ambiguous lands.
"First week is shakedown. Tune. Then walk away."	Drafts are noisy on day one. Tighten the prompt by Friday.

YOUR ROUTINES three routines the prompt drafts · everything else is bonus

01	STATE-REFRESH nightly · ~02:00	Reads yesterday's PRs, Jira changes, standup notes. Drafts tomorrow's state-shared.md block.	team-brain/
02	PROJECT-STALENESS-CHECK sunday eve	Diff's project-shared.md against the live source-of-truth. Flags stale lines for Monday review.	team-brain/
03	DECISIONS-PROMOTION friday	Scans the week for "we decided X" patterns. Drafts append-entries to decisions.md for confirmation.	team-brain/
+	BONUSES grow over time	Trigger-based: on-PR-merge writes a one-liner · on-Jira-status-change flags scope drift · Notification rules: ping only when a draft NEEDS REVIEW · Cost guardrails: cap tokens per run	platform-side

KEEP IT FRESH

STEWARD'S MORNING PASS ~ 3 min <ol style="list-style-type: none"> Open the routines-inbox or platform draft queue. Skim the night's <code>state-refresh</code> draft. Approve, edit, or reject. Promote to <code>state-shared.md</code>. Notify team if anything material shipped. 	ROUTINE TUNE-UP ~ 15 min · monthly <ol style="list-style-type: none"> Review the month's drafts. Any consistently noisy? Tighten the prompt. Any consistently empty? Kill the routine or fix the connector. Watch the token bill. Cap budgets that crept up. <p>Module 4 (Connect Your Tools) covers connector scoping.</p>
--	--

YOUR AI NEEDS THE SAME ACCESS A HUMAN HAS → NOT MORE.

Connect Your Tools

connectors with scope-narrowing · for IT Pros · BA · architect · tester · DBA · support · tech writer.

INVENTORY YOUR CONNECTORS group by role · pick what your team has · default-skip the sensitive classes —

KNOWLEDGE Drive · SharePoint · Notion · Confluence · your team-brain folder · reference repo.	ACTIVITY GitHub · Jira (Atlassian Rovo MCP) · Linear · Slack/Teams (specific channels) · CI logs.	IDENTITY & CALENDAR Google Workspace · Outlook · on-call rotation · org chart (skim only).	DEFAULT-SKIP Email inbox · HR / payroll · legal hold · NDA repos · customer support (unless tightly scoped).
---	---	--	--

RUN THE PROMPT three path-specific versions on the website · pick yours · steward owns the audit —

<code>connectors-builder.md</code>	path A · B · C →	pickbits.ai/sb
"Act as my Connector Stack builder."	Drafts three artifacts in 30 minutes. Read-only by default.	
"Default to READ-ONLY. Recommend writes only when explicitly asked."	Write requires conscious decision + allowlist. Never silent.	
"Exclude before include. Block known-sensitive first."	Easier to audit "what's blocked" than "what's allowed". Smaller blast radius.	
"Specific scopes (repo:foo · branch:main), not 'GitHub'."	Generic scopes drift. Specific scopes survive audits.	
"Audit log is the contract. Without it, scope rules are aspirational."	If you can't see what got read, you can't trust what didn't.	

YOUR CONNECTOR STACK three artifacts the prompt builds · everything else is bonus —

01	CONNECTORS.MD rarely changes	The catalog. One row per authorized connector: name, platform, owner, what it reads, last review.	team-brain/
02	SCOPE-RULES.MD per-connector	The narrowing. Include patterns, exclude patterns, read-only flag, write-allowlist (if any).	team-brain/
03	AUDIT-POLICY.MD rarely changes	What's logged, retention, who reviews, alert rules, incident response steps.	team-brain/
+	BONUSES grow over time	Read-only default: writes require allowlist · MCP server: for custom internal tools (Path B) · Vendor-portable: markdown definitions migrate across <i>Claude / ChatGPT / Gemini</i>	platform-side

KEEP IT FRESH —

STEWARDS CONNECTOR AUDIT	~ 15 min · monthly	EXTERNAL REVIEW	~ 30 min · quarterly
1. Open the platform's connector log (or your MCP <code>audit.log</code>). 2. Top-N read counts – expected? Any ALERT events? 3. Remove anything unused for >30 days. Update <code>connectors.md</code> last-review.		1. Have someone OUTSIDE the team read <code>connectors.md</code> + <code>scope-rules.md</code> . 2. Look for scopes that grew without justification. 3. Look for write-allowlists that grew. Patch.	
		Module 5 (Meeting to Memory) covers transcript intake.	

full guide + path-specific prompts pickbits.ai/sb // next: Module 5 – Meeting to Memory

DECISIONS LAND IN MEETINGS → AND EVAPORATE BY FRIDAY. THE FIX.

Meeting to Memory

transcripts in · structured context out · for IT Pros · BA · architect · tester · DBA · support · tech writer.

PICK WHAT TO CAPTURE capture by track · default-skip the privacy-sensitive ones

<p>DECISIONS TRACK</p> <p>Sprint review · architecture / RFC review · leadership / steering · vendor decisions. → decisions.md</p>	<p>ACTION TRACK</p> <p>Standup · planning · cross-team sync · post-incident review action items. → state-shared.md</p>	<p>KNOWLEDGE TRACK</p> <p>Project kickoff · retro · customer interview · tech debt planning. → project-shared.md / new docs</p>	<p>DEFAULT-SKIP</p> <p>1:1s · performance reviews · all-hands info-only · brainstorm with no decision.</p>
---	---	--	---

RUN THE PROMPT three path-specific versions on the website · pick yours · steward owns the archive

meeting-stack-builder.md	path A · B · C →	pickbits.ai/sb
"Act as my Meeting Stack builder."	Drafts three artifacts + N templates in 30 minutes.	
"One template per meeting type. Output to a team-brain artifact."	Sprint review → decisions.md · standup → state-shared.md · kickoff → project-shared.md	
"Privacy filter strips PII before the AI sees the transcript."	Customer names, salary, NDA refs – drop in pre-filter, not post-hoc.	
"Archive is the staging buffer. Steward promotes; AI doesn't."	A human eyes every entry before it lands in team-brain. Friday Sync = 10 min.	
"Capture is opt-in. Default-skip is safe."	1:1s, performance, all-hands – never captured unless deliberately added.	

YOUR MEETING STACK three artifacts the prompt builds · everything else is bonus

01	MEETING-TEMPLATES.MD rarely changes	One entry per meeting type. Source filter, prompt body, output target, retention.	team-brain/
02	TRANSCRIPT-PIPELINE.MD per-platform	Pipeline: meeting tool → privacy filter → AI prompt → meeting-archive → team-brain promotion.	team-brain/
03	MEETING-ARCHIVE/ one folder per type	Staging buffer: decisions/, actions/, knowledge/. Steward promotes weekly. Retention per policy.	team-brain/
+	BONUSES grow over time	Auto-routing: tag recording with type, Module 03 routine runs the right template · Speaker awareness: diarisation preserves "who said what" · Privacy filter: non-negotiable for 1:1s and performance	pipeline-side

KEEP IT FRESH

<p>FRIDAY PROMOTION PASS ~ 10 min · weekly</p> <ol style="list-style-type: none"> Open this week's meeting-archive/ folders. Promote anything affecting team-brain (paste section into state-shared / decisions / project-shared). Leave noise in archive (or delete if truly empty). 	<p>TEMPLATE TUNE-UP ~ 15 min · monthly</p> <ol style="list-style-type: none"> Review the month's archive. Any template producing noisy or empty outputs? Tighten the prompt. Add new meeting types worth a template. Move anything from default-skip to capture-track if patterns warrant. <p>Module 6 (Pattern Library) names + parameterises the prompts you reuse.</p>
---	---

full guide + path-specific prompts pickbits.ai/sb // next: Module 6 – Pattern Library

YOU'VE WRITTEN THIS PROMPT THREE TIMES → NAME IT.

Pattern Library

named, parameterised prompts · for IT Pros · BA · architect · tester · DBA · support · tech writer.

INVENTORY YOUR RE-WRITES a pattern earns its name when you've written something three times

STATUS & COMMS status-update · all-hands-narrative · exec-brief · incident-postmortem.	DECISIONS & DESIGN decision-doc · architecture-decision-record · rfc-reviewer · vendor-evaluation.	REVIEWS pr-review · design-review · retro-reviewer · change-management-brief.	PERSONAL PRODUCTIVITY rubber-duck · weekly-review · ellis · reading-list builder.
--	--	---	---

RUN THE PROMPT three path-specific versions on the website · pick yours · grow the library over time

<code>pattern-library-builder.md</code>	path A · B · C → pickbits.ai/sb
"Act as my Pattern Library builder."	Drafts three artifacts + 6 starter patterns in 30 minutes.
"Each pattern under 400 tokens. Patterns are scaffolds, not solutions."	Tight scaffolds invite use. Bloat invites avoidance.
"Three uses or it's not a pattern."	Drafts in <code>patterns/_drafts/</code> . Promote on the third use.
"Calling syntax: Run pattern: <name> with <params>."	One convention across all paths. Wrappers (Path A) auto-resolve.
"Markdown is portable. The library survives platform migrations."	Some patterns work in Claude · ChatGPT · Gemini. The library is the moat.

YOUR PATTERN LIBRARY three artifacts the prompt builds · everything else is bonus

01 PATTERNS.MD rarely changes	The catalog. One row per pattern: name, when to use, parameters, output target, version, lineage.	team-brain/
02 PATTERNS/ grows over time	Folder of named markdown files. status-update.md · decision-doc.md · pr-review.md · etc.	team-brain/
03 PATTERN-USAGE.MD rarely changes	How to call patterns from a chat. Parameter syntax. Composition. Versioning convention.	team-brain/
+ BONUSES grow over time	Versioning: v1/v2 suffix on breaking changes; old in <code>_archive</code> · Lineage: <code>//</code> derived from: decision-doc v2" · Cross-team sharing: hand the <code>patterns/</code> folder over – it works in any stack	git history

KEEP IT FRESH

PROMOTE ON THIRD USE ~ 5 min <ol style="list-style-type: none"> You've written this prompt three times. Move from <code>_drafts/</code> to <code>patterns/</code>. Add a row to <code>patterns.md</code>. Tighten by 20% on the way through. 	LIBRARY AUDIT ~ 30 min · quarterly <ol style="list-style-type: none"> Open <code>patterns.md</code>. Anything unused 60+ days – archive. Anything you keep adding inline overrides to – bump version. Check lineage drift: newer patterns redundant with older ones? <p>Module 7 (AI-Native Operator) wires all six modules into one rhythm.</p>
---	---

full guide + path-specific prompts pickbits.ai/sb // next: Module 7 – AI-Native Operator

YOU HAVE THE STACK → NOW PROVE IT'S PAYING OFF.

AI-Native Operator

the capstone · six modules into one rhythm · with the metrics that prove it.

PICK WHAT TO MEASURE 3-5 metrics across all four classes · with baselines · read weekly · act monthly

INPUTS PRs / commits / Jira / week · standup signal density · meeting hours captured · connector reads.	OUTPUTS state-shared updates / week · decisions promoted · pattern invocations · onboarding hours saved.	PROCESS cycle-time · decision-latency · cross-team handoffs avoided · time-to-context-loaded.	STACK HEALTH audit alerts (target 0) · routine miss-rate · pattern adds-vs- archives · steward load (burnout).
---	---	---	--

RUN THE PROMPT three path-specific versions on the website · pick your level (solo · team · org)

<code>operator-stack-builder.md</code>	path A · B · C →	pickbits.ai/sb
"Act as my Operator Stack builder."	Drafts three artifacts in 30 minutes for your level.	
"3-5 metrics, no more. With baselines."	Anti-pattern: tracking 12, glancing at none. Better: 4, read every Monday.	
"One tweak per monthly review."	Resist redesign. Patterns drift when you tweak too much at once.	
"Stack temperature: green / yellow / red one-liner each morning."	Yellow: state stale or routine missed. Red: audit alerts pile up. Cheap insurance.	
"Failure modes catalog. Read at every operating review."	Steward burnout · connector creep · pattern bloat · audit drift. Catch early.	

YOUR OPERATOR STACK three artifacts the prompt builds · everything else is bonus

01 OPERATOR-RHYTHM.MD rarely changes	Daily 60-sec context load · Friday 15-min Sync · Monthly review · Quarterly external check. With "stack temperature" one-liner.	team-brain/
02 METRICS.MD grows over time	3-5 leading indicators. Cycle-time · decision-latency · handoffs-avoided · time-to-onboarded · pattern-reuse. With baselines.	team-brain/
03 OPERATING-REVIEW.MD monthly	Retro on the stack itself. Read metrics, scan failure modes, pick one tweak. Updates rhythm if needed.	team-brain/
+ BONUSES grow over time	Operator dashboard: 1-pager refreshed by Module 03 routine · Stack temperature: green / yellow / red morning glance · Failure modes catalog: <i>burnout · creep · bloat · drift · noise</i>	platform-side

KEEP IT FRESH

OPERATING REVIEW ~ 30 min · monthly <ol style="list-style-type: none"> 1. Read last month's metrics. Read failure-modes catalog. 2. Decide on <i>one</i> tweak. Just one. 3. Update artifacts; notify team. 	EXTERNAL REVIEW ~ 60 min · quarterly <ol style="list-style-type: none"> 1. Person OUTSIDE team / org reads everything (rhythm, metrics, review, plus M02-06 artifacts). 2. They flag drift you can't see. 3. Patch. Update operating-review failure modes if a new one surfaced. <p>Done. The series is complete; your operator practice has just begun.</p>
---	---

full series + 1:1 / team rollouts pickbits.ai/sb // coaching · consulting at pickbits.ai

© 2026 PICKBITS.AI // BY MARK PICKERING // MODULE 7 OF 7 · CAPSTONE // V1.0

COMPANION TO MODULE 01 · PERSONAL KNOWLEDGE DATABASE

Set up your **Second Brain** in any AI you already have

Click-by-click for six tools across three paths. The Module 01 prompt works in any AI tool with file context – only the setup steps differ. If your tool isn't listed, find the closest one; the workflow is identical, only the labels shift.

THE THREE PATHS

A On your computer

Agent reads your folder and writes the three context files plus a CLAUDE.md / AGENTS.md wrapper at the project root. Future sessions auto-load.

Claude Code · GitHub Copilot Agent · Cursor · Windsurf · Codex CLI · Gemini CLI · Aider

B In a cloud Project

Upload your docs to a Project once; chat against them across many sessions. You save the three files yourself.

Claude Projects · ChatGPT Projects · Gemini Gems · Custom GPTs · Notebook LM

C In any chat

Attach files at session start, run the prompt, save outputs. Works everywhere; you'll re-attach next session.

Microsoft 365 Copilot · ChatGPT · Claude.ai · Gemini · Perplexity

BEFORE YOU START

- 1. Files.** 3-5 documents from one project – briefs, notes, decision threads, schemas. (Detail in Module 01.)
- 2. An AI tool with file context.** Pick A, B, or C above based on what your tool can do.
- 3. ~10 minutes.** Setup is one-time. Daily load is < 60 seconds after that.

Full prompt + module page // pickbits.ai/pkd

PICKBITS.AI // SETUP COMPANION // 1 / 5

A On your computer

The agent runs the silent inventory, asks 3-5 questions, then writes the three context files into `context-stack/` plus a thin `CLAUDE.md` + `AGENTS.md` wrapper at the project root. The wrapper auto-loads the context files at every future session – daily load becomes *"open and ask where I left off."*

Claude Code

1. Open a terminal in the folder with your project source docs.
2. Run `claude`.
3. Paste the prompt (from pickbits.ai/pkd).
4. Answer the 3-5 questions.
5. Approve writes – limited to `context-stack/` plus the wrappers (`CLAUDE.md`, `AGENTS.md`) at the root.
6. The debrief lists what was built. Future sessions auto-load via `CLAUDE.md`.

Also works in: *Codex CLI, Aider, Gemini CLI – any terminal agent in a workspace.*

GitHub Copilot Agent

1. Open your project folder in VS Code.
2. Open Copilot Chat. Switch the mode dropdown from *Ask* to *Agent*.
3. Paste the prompt.
4. Approve tool calls when prompted (`readFile`, `writeFile`).
5. Answer the 3-5 questions.
6. Files appear in `context-stack/`. The wrapper at `AGENTS.md` auto-loads them next time – Copilot reads it as the cross-tool agent-config convention.

Also works in: *Cursor, Windsurf, Cline, Continue – any IDE-embedded agent with workspace write.*

Daily load on Path A: open a fresh chat in the same folder. Ask *"where did I leave off?"* The wrapper tells the agent which context files to load. ~30 seconds, no manual references needed.

B In a cloud Project

Drop your 3-5 source docs into a Project once and they're attached to every chat in that space – closest thing to "wakes up knowing" without folder write. The agent outputs the three context files as *markdown blocks*; you save them locally (and optionally re-upload to the Project so future chats inherit them).

Claude Projects

1. Sign in at claude.ai. Click *Projects* in the sidebar.
2. Click *Create project*. Name it (a short slug, e.g. **feedrunner**).
3. Open *Project knowledge*; upload your 3-5 source docs.
4. Open a new chat *inside the project*.
5. Paste the prompt. Answer the 3-5 questions.
6. Save the three returned markdown blocks to a local **context-stack/**. Re-upload them to the Project so future chats inherit them.

Also works in: *Notebook LM, any "create a workspace, attach files, chat" surface.*

ChatGPT Projects

1. Sign in at chatgpt.com. Click *+ New Project* (Plus / Team / Enterprise).
2. Name the project. Open *Files*; upload your 3-5 source docs.
3. (Optional) add custom instructions: "Treat attached docs as my Context Stack source."
4. Open a new chat *inside the project*.
5. Paste the prompt. Answer the 3-5 questions.
6. Save the three markdown blocks to a local **context-stack/**. Re-upload them to the Project so future chats inherit them.

Also works in: *Gemini Gems, Custom GPTs, Microsoft 365 Copilot Agents.*

Daily load on Path B: open a chat in the project (the three context files are already attached). Ask "*load identity.md + the current project-*.md + state.md, tell me where I left off.*" Update **state.md** at session end and re-upload.

C In any chat

Works anywhere, but you'll feel the friction by week two – every new chat means re-attaching files. The fix is to graduate to Path B once you know the workflow. Until then, this works fine; just budget 90 seconds for daily load instead of 60.

Microsoft 365 Copilot

1. Open Copilot – copilot.cloud.microsoft, the Edge sidebar, or in-app (Word, Outlook, Teams).
2. Toggle to *Work* scope so it can see your tenant's SharePoint / OneDrive.
3. Either reference docs by name in the prompt ("read the FeedReader brief in OneDrive/Projects/...") or attach them with the paperclip.
4. Paste the Module 01 prompt. Answer the 3-5 questions.
5. Copy each of the three markdown blocks. Save into a Word doc each, or as a local `.md` file.
6. Save to OneDrive in `context-stack/` – keeps them portable across devices.

Also works in: Copilot in Word, Outlook, or Teams; Copilot Pages – UI labels shift, the workflow is identical.

ChatGPT (no project)

1. Sign in at chatgpt.com. Start a new chat.
2. Click the paperclip. Upload your 3-5 source docs.
3. Paste the prompt. Answer the 3-5 questions.
4. Copy the three markdown blocks. Save to a local `context-stack/` folder.
5. Next session: re-attach the three context files *and* any updated source docs.
6. When tired of re-attaching, graduate to Path B (ChatGPT Projects on Plus / Team / Enterprise).

Also works in: Gemini app, Claude.ai chat, Perplexity Pro – any chat surface with file upload.

Daily load on Path C: attach `identity.md` + the relevant `project-*.md` + `state.md` to a fresh chat, ask "load these, tell me where I left off;" –90 seconds – friction is the cost of universality.

THE PROMPT – ANNOTATED DIGEST

context-stack-builder.md

Full prompt at pickbits.ai/pkd

"Act as my Context Stack builder."	One role, one job. Keeps the AI from drifting to generic-assistant mode.
"Read only the folder I share. Write only to context-stack/ and the Path A wrappers."	Safety contract. Scope-limited – no surprise writes elsewhere on your machine.
"Skip credentials, .env, anything secret. Note skips in debrief."	Visible omissions. You see what didn't go in, not just what did.
"Scan silently. Then ask 3–5 short questions in one message."	Inventory before noise; clarification over fabrication.
"Mark uncertain spots [TBD]. Debrief what you skipped."	Visible gaps, not invented confidence. You can trust what's there.

THE DRILL – SETUP IS ONE-TIME, THESE TWO ARE FOREVER

Daily Load

~30s (A) · 60–90s (B/C)

1. Open your AI. On Path A, the wrapper auto-loads. On B/C, attach the three files.
2. Ask: *"where did I leave off?"*
3. Work. At session end, update **state.md**.

Monday Reset

~10 min

1. Update each **project-*.md** with last week's decisions and status.
2. Archive last week's **state.md**. Start a fresh one.
3. Prune **patterns/** – delete anything unused in 30 days.

The failure mode – name it so you avoid it. Skip the Monday Reset for two weeks and **state.md** becomes a lie. Your AI will load it confidently and answer wrong. The whole system is downstream of this 10 minutes – protect it on your calendar.

1:1 coaching pickbits.ai/coaching // team training pickbits.ai/consulting // open source github.com/pickbitsai/context-stack

PICKBITS.AI // SETUP COMPANION // 5 / 5