

Reset3 v3.0 — Decisions Recap & Technical Plan

Purpose: Minimal AI integration that upgrades selection quality without turning Reset3 into a task manager.

v3.0 Decisions Recap

- **Prime directive:** reduce decision friction. No setup-heavy features.
- **Output contract stays:** exactly 3 items (1 Primary + 2 Support) with short reasons.
- **No chat UI:** AI is used only for clarity/selection, not conversation.
- **Guardrails stay:** brain dump persistence, prune-on-complete, Primary-first rule, multi-cycle daily use.
- **Build approach:** develop locally first using a small Express API proxy (real AI calls) + a DEV_MODE switch to avoid burning tokens while debugging.
- **Deployment intent:** once stable, deploy the API as a serverless proxy (Option A) to avoid managing an always-on server.

v3.0 Feature Scope (Minimal AI Integration)

- **AI Selection Engine (core):** replace heuristic scoring with a model call that returns structured JSON (Primary/Support + reasons).
- **AI Clarifier Mode (optional):** rewrite vague tasks into the next concrete action (no conversation).
- **AI Compression Mode (optional):** detect duplicates/overlaps and suggest merges (user confirms; no auto-delete).

Explicitly out of scope: accounts/sync, dashboards/analytics, categories/tags, points/streaks, Kanban, long-form coaching essays.

Technical Plan — Implementation To■Do List

Phase 0 — Prep

- Create git branch **v3-ai**.
- Add **.env** for API keys (never commit).

Phase 1 — Local Express API proxy (real AI calls)

- Create a small server folder (e.g., **/server**) with Express.
- Expose **POST /api/generate**.
- Backend holds the API key and calls the model; frontend never sees the key.
- Add basic safety: request size limit + simple rate limit (even locally).

Phase 2 — JSON contract + validation (non-negotiable)

- Define strict response schema: **primary** + **support[2]**, each with **id**, **text**, optional **minutes**, and **reason**.
- Validate server output. If invalid or API fails → fallback to v2 heuristic selection.
- Keep reasons short (target: < 20 words each).

Phase 3 — Frontend integration

- Replace v2 Generate logic with a call to **/api/generate**.
- Preserve v2 behaviors: persistence, prune-on-complete, Primary-first gating, cycles counter.

Phase 4 — Optional AI actions

- Add **Clarify Primary** button (same endpoint with mode flag or a separate route).
- Add **Compress Brain Dump** button that returns merge suggestions; user approves applying changes.

Phase 5 — Cost & spam controls

- Add a short cooldown (e.g., 10 seconds) on Generate.
- Implement **DEV_MODE** switch: return canned JSON while debugging UI; flip off to test real AI.

Phase 6 — Deploy (after local stability)

- Deploy the API layer as a **serverless proxy** (Option A) and keep the same request/response contract.
- Keep client unchanged; only swap base URL to the deployed endpoint.

Notes: Building with Express locally does *not* lock you into running a permanent server in production. It's a fast way to develop and debug. Deployment can still be serverless later.