# AI assistance methodology

Used a mix of Cdex and Claude code with VsCode IDE to generate codes and iterations. Claude code is very good and vague prompt to propose a plan and give you idea when not sure. Very good at making sure all files in the repo are up to date. However use lots of credits very fast. So I learn to get use of Codex more. And improve my prompt.

Claude code clearly excel at figuring things out and this is very helpful in exploration [hase. So I hav been using claude code mostly when I was trying to figure out things with sometimes missing cntext. While I was using cidex with most of the time clear instructions. Exemple of prompt for the Streamlit refactor below, done by codex. Codex you need to explicitly reference files in the prompt.

Also use claude to review the code, if hallucinations, errors, incoherences. And give improvement plans.

# Screenshots needed

# What to capture Where in report How to get it

1 Ingest page — single PDF upload with extraction result, token usage summary, noise level §4.1 / §7.2 Upload one PDF, screenshot the result panel

2 Ingest page — bulk mode — progress bar, summary table (saved/skipped/failed) §7.2 step 15 Upload 3+ PDFs in bulk mode

The image displays a user interface for uploading PDF documents for bulk processing, with a list of example documents and a summary of the extraction process.

AI-generated content may be incorrect.

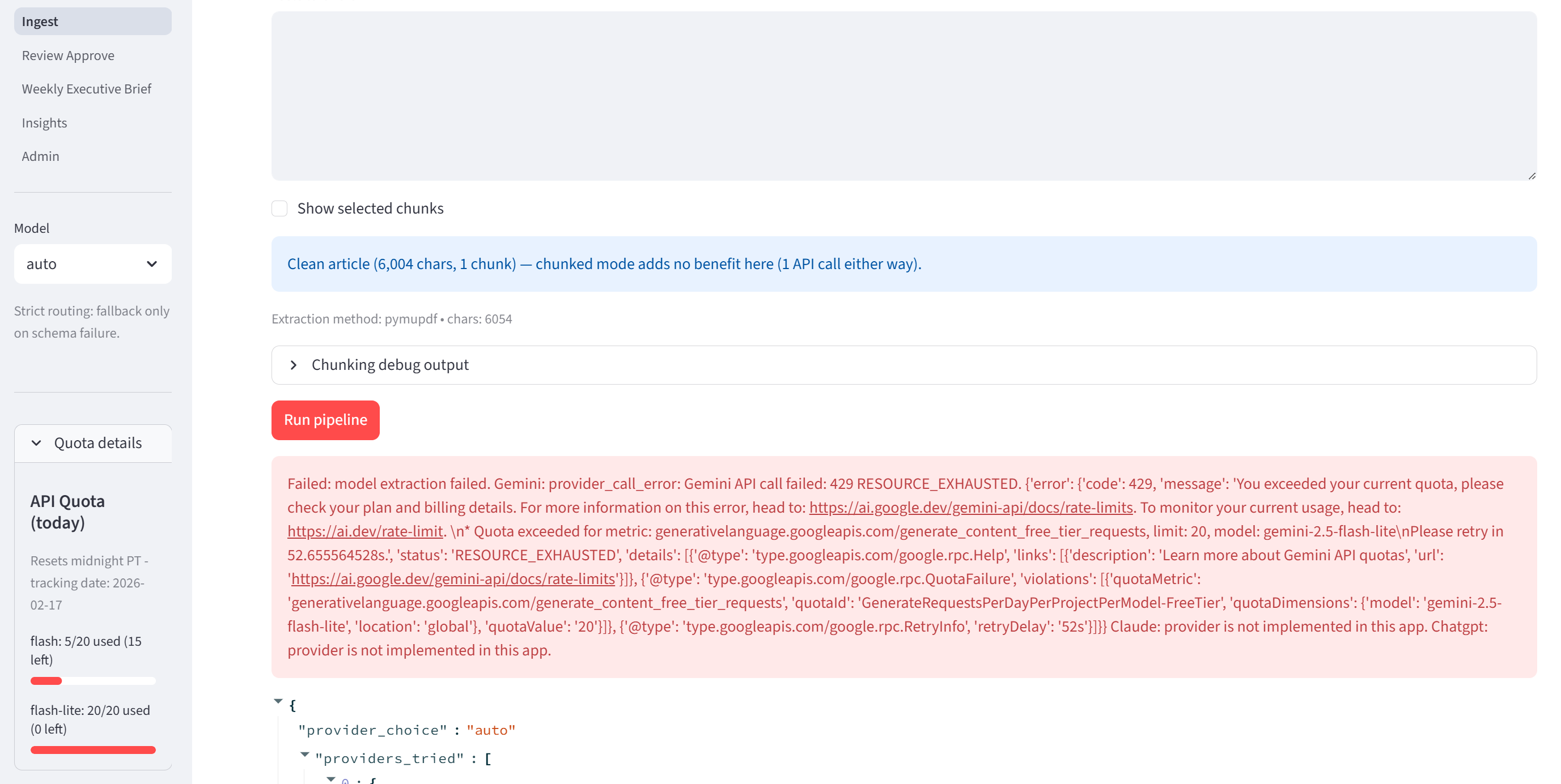
3 Inbox page — title-first expandable cards, inline approve buttons, batch actions §4.1 Open Inbox with 5+ records

4 ~~Record detail page — confidence detail breakdown, Quick Approve button §4.1 Open one record, expand confidence detail~~

6 Insighst — Priority Distribution — stacked bars + High-Ratio line §4.1 Same dashboard view

7 Weekly Brief — AI-generated — executive brief output with citations §4.1 Generate one AI brief

8 API quota sidebar — progress bars showing per-model usage §7.5.9 Open Ingest page with some calls used



~~9 Documents library — filtered view with evidence previews §4.1 Open Documents page with filters applied~~

Diagrams to create

# What to diagram Where in report Content

10 System architecture — end-to-end flow from PDF upload to executive brief §4.1 PDF → text extraction → clean/chunk → model routing → postprocess → validate → store → brief/dashboard

11 Spec-to-code evolution (Phase 1 → Phase 3) §6.1 ASCII placeholder already in report — convert to clean diagram. 5 spec files → 3 touchpoints

12 Extraction prompt before vs after §7.5.8 Side-by-side showing the added topic guidance + competitor context sections

13 Two-tier region architecture §6.1 or §7.2 DISPLAY\_REGIONS (broad buckets) vs FOOTPRINT\_REGIONS (country-level) with collapse map

Schema / content to fill in

# What Where Source

14 JSON schema excerpt §6.2 (line 302) Copy from record\_response\_schema() in src/model\_router.py or use the clean version from spec/SKILL\_final.md §9

15 Results data §9 (lines 675-685) Run 5-10 real ingests, record: total docs, schema pass rate, mean tokens, mean time, override rate, strengths/weaknesses

The image appears to be a software interface displaying a JSON object, possibly related to data analytics or data processing.

AI-generated content may be incorrect.

## Check for duplicate document in ingest before model call:

The image shows a user interface with various options for managing text content, including fields for title, review, and API quota, with a note about a duplicate record detected during extraction.

AI-generated content may be incorrect.

# Example of quality improvement:

Western European passenger car registrations fall 2.7% year over year in January

AI-generated content may be incorrect.

The image contains a Quality Checklist for comparing JSON and S&P PDF data, highlighting discrepancies in numerical values, titles, entities, and geographical mentions.

AI-generated content may be incorrect.

Actually the high blocker captures by ChatGPT are correct but also, when I checked the original pdf file was showing 2.7%and 2.8%, andalso 860,319 and 860,318. So 1) and 2) were not really errors. Just the quality of source file. That was correct behavior.

However 3), geo signal distortion has been fixed. This was capture in the extraction;

US.

AI-generated content may be incorrect.

From source file:

The image depicts a forecast of slight growth in 2025 and 1.0% in 2026, with the unstable geopolitical situation expected to negatively impact global trade and German economic growth due to rising protectionism and disruptions in global supply chains.

AI-generated content may be incorrect.

Fix with claude code:

Geo Signal Distortion Documentation: US mentioned in source but not extracted; solution includes adding extraction rule for country_mentions.

AI-generated content may be incorrect.

US

AI-generated content may be incorrect.

Test cases for evaluation (§8.3)

# What Status

16 Case A: Market note (region-heavy) — source snippet + JSON + brief Needs real ingest

17 Case B: Competitor/OEM program (priority-driven) — source snippet + JSON + brief Needs real ingest

18 Case C: Government/regulatory (actor type driven) — source snippet + JSON + brief Needs real ingest

Total: 9 screenshots + 4 diagrams + 1 schema fill + 1 results section + 3 test cases = 18 items.

# Maintenance heavy risks

**a small constant like that usually *reduces* maintenance**, as long as you treat it like a configuration list and keep it stable.

What makes it complex (or not) depends on *where it lives* and *how often it changes*.

**When it stays simple ✅**

It’s low-maintenance if:

* It lives in **one place** (e.g., constants.py)
* It’s **small and readable**
* You add a short comment explaining the intent (priority escalation = “exec attention”)
* You add a tiny unit test / smoke test that proves escalation works

In that setup, updating it is literally editing a list.

**When it becomes maintenance-heavy ⚠️**

It can get annoying if:

* You start adding many special-case lists (“themes for region X”, “themes for OEM Y”, “themes for supplier vs OEM”)
* Priority logic becomes a maze of overlapping rules
* The theme names drift over time (renamed themes) and this set isn’t updated → silent misbehavior

# Streamlit UI/UX improvements:

Do many pages, cognitive load. Need to refactore the streamlit interface to make the UI feels more a pipeine not a toolbox.

Ingest → Validate → Approve → Synthesize Weekly Brief

**Stage 1 — Capture**

**Stage 2 — Validate**

**Stage 3 — Approve**

**Stage 4 — Weekly Brief**

**Stage 5 — Insights (optional analytics)**

So I sent the stremalit pages to ChatGPTthat has my entire Cognitra context and ask to propose a plan to improve that Streamlit interface and once I agree with the plan, I ask him to right a prompt for codex (see below)

Before

Prompt:

You are Codex operating inside a Streamlit multipage repo (Cognitra). Your job: refactor the UI so the app reads like a single, clear workflow: Ingest → Review/Approve → Weekly Executive Brief → Insights, with Admin hidden under Advanced.

Constraints

- Do not change the core pipeline logic (ingest/extract/postprocess/validate/store) unless required by the UI refactor.

- Keep storage format and APIs the same (data/records.jsonl, src/storage.py functions).

- Preserve backward compatibility: old records may lack newer fields → always use .get() with defaults.

- Keep behavior deterministic except where it already uses LLM (weekly brief synthesis).

- Minimize risk: move code, don’t rewrite algorithms.

Repo context (current pages)

- pages/01\_Ingest.py

- pages/02\_Inbox.py

- pages/03\_Record.py

- pages/04\_Dashboard.py

- pages/05\_Weekly\_Brief.py

- pages/06\_Review\_Brief.py

- pages/07\_Documents.py

- pages/08\_Admin.py

Target navigation (4 main pages)

1) Ingest (keep as page 01)

2) Review & Approve (merge Inbox + Record; remove standalone Record page)

3) Weekly Executive Brief (merge Weekly\_Brief + Review\_Brief)

4) Insights (Dashboard)

Advanced (collapsed / hidden): Admin utilities

High-level refactor plan

A) Merge Inbox + Record

- Create a new page: pages/02\_Review\_Approve.py (or rename 02\_Inbox.py to 02\_Review\_Approve.py).

- This page must contain:

1. The existing filter + list UI from Inbox (priority/status/source/search).

2. A record detail panel (right side) that shows and edits the selected record JSON (like Record page), including:

- Rendered Intelligence Brief preview

- Schema validation errors (validate\_record)

- Save changes (overwrite\_records)

- Quick Approve / Disapprove

- Reviewed By + Notes

- Exclude-from-brief toggle

- Selection UX:

- Use a table (st.dataframe with row selection if available in this Streamlit version) OR a selectbox fallback.

- Clicking “Open full record” should no longer switch pages; it should set st.session\_state["selected\_record\_id"] and open details in the same page.

- Delete pages/03\_Record.py after migration (or leave stub that redirects to 02\_Review\_Approve).

B) Merge Weekly\_Brief + Review\_Brief into one flagship page

- Create/rename to: pages/03\_Weekly\_Executive\_Brief.py (or keep 05 but refactor and rename later).

- This page must include:

1. Candidate selection for a period (days back).

2. A clear “Approved-only” default selection (only Approved and not excluded).

3. AI brief generation (existing synthesize\_weekly\_brief\_llm flow).

4. Save brief + download + open in email client (existing features).

5. “Review” features from Review\_Brief:

- Load latest saved brief markdown

- Compare to previous brief diff (optional, collapsed)

- Show brief metadata (selected ids, week\_range, usage)

- Show table of candidate/approved records for the brief period

- Flag approved records missing from selection

- Ability to toggle Approved / Exclude-from-brief for a selected record (inline controls)

- Remove pages/06\_Review\_Brief.py after migration (or leave a stub that redirects).

C) Remove/retire Documents page

- Delete pages/07\_Documents.py (or leave a stub). Its functionality is redundant; review page already provides source links and browsing.

D) Insights page

- Keep pages/04\_Dashboard.py but rename it to pages/04\_Insights.py (or keep filename but change title to “Insights”).

- Ensure it is clearly “optional analytics” and not part of the core pipeline.

E) Admin

- Keep pages/08\_Admin.py but:

- Rename title to “Advanced / Admin”

- Add a warning caption: “Developer/analyst utilities”

- Optionally hide destructive buttons (clear all records) behind an “I understand” checkbox.

UX cleanup requirements (must implement)

1) Every main page should show a small progress ribbon at the top:

“1 Ingest → 2 Review → 3 Weekly Brief → 4 Insights”

Highlight current step (simple markdown is fine).

2) Collapse advanced diagnostics on Ingest behind expanders:

- Quota details

- Overrides applied

- Chunking debug output

3) Reduce visual noise in Review page:

- Collapsed list view should show: Title, Source, Date, Priority/Confidence, Status

- Details show evidence + insights + metadata + JSON editor.

4) Preserve all existing functionality (approve/disapprove, notes, reviewed\_by, exclude flags, saving records, rendering briefs).

Implementation steps

1) Scan the repo and identify shared helpers duplicated across pages (e.g., \_norm\_status, safe list parsing). Consolidate into src/ui\_helpers.py (or similar) and import from pages.

2) Implement new merged pages first, leaving existing pages intact until new pages work.

3) Add a temporary redirect stub in old pages:

- 03\_Record.py → message + st.switch\_page to the new Review page

- 06\_Review\_Brief.py → redirect to Weekly Executive Brief

- 07\_Documents.py → redirect to Review page (or display “removed”)

4) Update page numbering to match the new workflow ordering so sidebar reads correctly.

Acceptance criteria (definition of done)

- App sidebar shows only: Ingest, Review & Approve, Weekly Executive Brief, Insights, Advanced/Admin.

- User can:

- Ingest PDFs and generate records as before.

- Review and edit record JSON in the Review page without leaving the page.

- Approve/disapprove/exclude records.

- Generate AI weekly brief, review latest saved brief, compare diff, and save/export.

- No import-time errors; Streamlit app runs.

- Minimal code duplication (helpers extracted).

- Keep existing function calls to src.\* modules intact; avoid changing business logic.

Deliverables

- Code changes across pages + any new helper module.

- Remove or redirect deprecated pages.

- Brief README update (optional): describe the new workflow navigation.

Now start by:

1) Listing the current page behaviors and any shared helpers you find.

2) Implement merge A (Review & Approve) with a working selection + detail split layout.

3) Then implement merge B (Weekly Executive Brief).

4) Finally cleanup pages (remove/redirect) and rename titles.

Proceed.

The image displays a user interface for an Automotive Market Intelligence (MVP) platform, showcasing various functionalities such as data ingestion, record management, and dashboard analytics.

AI-generated content may be incorrect.

The image displays a user interface for a document management system, with options to upload PDFs, paste text, and manage quota and file limits.

AI-generated content may be incorrect.

The image displays a document or dashboard interface with various tabs and statuses, including 'Inbox', 'Ingest', 'Record', and 'Review Status', with different priorities and sources such as 'Approved', 'Disapproved', and 'Pending'. It also shows a specific document titled 'Toyota Motor Appoints CFO Kenta Kon as New CEO in Profit Push - Bloomberg' with a status of 'Approved' and key insights about Toyota's leadership change and profitability efforts.

AI-generated content may be incorrect.

The image displays a dashboard interface with a news snippet about Toyota appointing Kenta Kon as the new CEO for their profit push, approved and marked with high confidence.

AI-generated content may be incorrect.

The image displays a weekly brief document with a list of selected items, including a new CEO appointment for Toyota, and notes on duplicate suppression.

AI-generated content may be incorrect.

The document is a technical review brief for Mercedes-Benz, discussing the company's new S-Class model, its financial situation, and executive changes at Toyota.

AI-generated content may be incorrect.

Toyota Promotes Akio Toyoda as CEO - Bloomberg.pdf,

AI-generated content may be incorrect.

The image displays a user interface for a data management system, featuring various tabs for managing and exporting data, including options for deduplication and exporting records in different formats.

AI-generated content may be incorrect.

AFTER

CODEX work summary:

**What I Found**

1. Current page split before refactor:
   * [01\_Ingest.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/): ingest pipeline + diagnostics.
   * [02\_Inbox.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/) + [03\_Record.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/): review flow split across two pages.
   * [05\_Weekly\_Brief.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/) + [06\_Review\_Brief.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/): brief generation/review split.
   * [07\_Documents.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/): overlapping browsing functionality.
   * [04\_Dashboard.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/): analytics page.
   * [08\_Admin.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/): utilities.
2. Shared helper duplication:
   * \_norm\_status duplicated in [02\_Inbox.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/), [03\_Record.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/), [07\_Documents.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/).
   * \_safe\_list duplicated in [04\_Dashboard.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/), [07\_Documents.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/).
   * Repeated link fallback, list joining, and record update patterns across review pages.

**Implemented**

1. Added shared helper module [ui\_helpers.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/) with:
   * normalize\_review\_status, safe\_list, join\_list, best\_record\_link, workflow\_ribbon.
2. Merge A complete:
   * New unified review page [02\_Review\_Approve.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/).
   * Includes filter/list queue + selection + detail pane + JSON editor + validation + rendered brief preview + approve/disapprove + reviewed\_by/notes + exclude toggle + save.
   * “Open full record” now keeps selection in-page via st.session\_state["selected\_record\_id"].
3. Merge B complete:
   * New unified flagship page [03\_Weekly\_Executive\_Brief.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/).
   * Includes candidate selection by period, approved-only default selection, deterministic + AI brief generation, save/download/email.
   * Includes latest saved brief load, metadata view, previous diff, candidate coverage table, missing-approved warnings, inline approve/exclude controls.
4. Insights refactor:
   * Replaced dashboard page with [04\_Insights.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/) (renamed title/position, analytics preserved).
5. Ingest UX cleanup:
   * Added workflow ribbon in [01\_Ingest.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/).
   * Collapsed diagnostics into expanders:
     + “Quota details”
     + “Overrides Applied”
     + “Chunking debug output” (single/manual preview).
6. Admin update:
   * Renamed to “Advanced / Admin” in [08\_Admin.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/).
   * Added warning caption and “I understand” checkbox gate for clear-all action.
7. Navigation cleanup:
   * Removed deprecated pages from pages/ so sidebar now shows only:
     + [01\_Ingest.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/)
     + [02\_Review\_Approve.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/)
     + [03\_Weekly\_Executive\_Brief.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/)
     + [04\_Insights.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/)
     + [08\_Admin.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/)
8. Home + docs:
   * Updated [Home.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/) workflow copy.
   * Updated [README.md](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/) with new workflow navigation.

**Validation**

1. Compiled updated modules successfully:
   * [ui\_helpers.py](https://file+.vscode-resource.vscode-cdn.net/c%3A/Users/sandr/.vscode/extensions/openai.chatgpt-0.5.75-win32-arm64/webview/)
2. Tests pass:
   * .\.venv\Scripts\python -m pytest -q (all passing).

And results

# Biggest frustration

This happened so many times, in just 5 minutes!

The text displays a timestamp indicating a reset time for a specific time zone, suggesting a reset or resetting action has occurred at 1:00 PM in the America/Los_Angeles time zone.

AI-generated content may be incorrect.

Very difficult to get good quality out when you want to have a control on the output. For exemple

Here, your country\_mentions are:

* Germany → **Western Europe**
* China → **China**
* United States → **US**

So regions\_relevant\_to\_kiekert should be exactly:

* ["Western Europe", "China", "US"]

But your record has:

* ["Western Europe", "China", "US", "Asia"]

# Quality improvement of records and briefs

Quality checking of extraction has been the hardesr part and hard to monitor, so I created a quick quality check module so I can compare the quality improvement over time.

Evaluation criteria

 **Did it stay within the evidence of the approved records?**

 **Did it synthesize well without inventing new facts?**

 **Did it create useful Kiekert implications/actions?**

**]**

# Quality

**What was implemented**

**1. Company alias groups — auto-derived (18 groups, up from 4)**

* [quality.py:109-155](vscode-webview://1ejn67e6ticskuf69or3qbflcj81kngea44i3u7jau3tq1u4gj1f/src/quality.py#L109-L155) — \_build\_company\_alias\_groups() auto-derives from postprocess maps + enriched with 14 extra OEM alias sets (Ford, GM, Hyundai, Stellantis, Nissan, Honda, Renault, Geely, Tata, Volvo, JLR, Kia, Audi, Porsche)
* Zero maintenance going forward — adding a new OEM to postprocess maps automatically expands QC coverage

**2. Granular evidence grounding**

* [quality.py:479-480](vscode-webview://1ejn67e6ticskuf69or3qbflcj81kngea44i3u7jau3tq1u4gj1f/src/quality.py#L479-L480) — configurable thresholds: EVIDENCE\_GROUNDING\_THRESHOLD = 0.60, EVIDENCE\_NEAR\_MISS\_THRESHOLD = 0.45
* Now reports per-bullet overlap percentages in finding notes
* **Two-tier severity**: Hard misses (<45%) → High, Near misses (45-60%) → Medium
* Reports avg\_overlap per record for threshold tuning visibility

**3. Duplicate-record cross-check**

* [quality.py:694-754](vscode-webview://1ejn67e6ticskuf69or3qbflcj81kngea44i3u7jau3tq1u4gj1f/src/quality.py#L694-L754) — \_check\_duplicate\_records() runs post-hoc on all target records
* **Exact match**: groups by dedupe\_key → duplicate\_record\_in\_brief (High)
* **Fuzzy match**: SequenceMatcher at 0.85 on title fingerprints → near\_duplicate\_titles (Medium)
* Skips pairs already caught by exact key to avoid double-reporting

**4. KPI trend tracking**

* [quality.py:1400-1475](vscode-webview://1ejn67e6ticskuf69or3qbflcj81kngea44i3u7jau3tq1u4gj1f/src/quality.py#L1400-L1475) — compute\_quality\_trends() compares current run against last N runs
* Direction-aware: knows which KPIs improve when they go up vs down
* Regression alerts trigger at 5% for ratio KPIs, 2-point for count KPIs
* New **kpi\_trends** sheet in Excel export showing run-over-run movement

**5. Feedback loop (Phase A)**

* [quality.py:1482-1575](vscode-webview://1ejn67e6ticskuf69or3qbflcj81kngea44i3u7jau3tq1u4gj1f/src/quality.py#L1482-L1575) — generate\_extraction\_feedback() aggregates chronic issues (appearing in 50%+ of recent runs)
* Maps 12 finding types to actionable prompt/config suggestions (template-driven, no LLM)
* CLI output now shows chronic issues and suggestions after each run

**6. Updated CLI output**

* [run\_quality.py:46-87](vscode-webview://1ejn67e6ticskuf69or3qbflcj81kngea44i3u7jau3tq1u4gj1f/scripts/run_quality.py#L46-L87) — prints trend deltas, regression alerts, chronic issues, and prompt suggestions

**All 97 existing tests pass.**

# Index

**grounded: the evidence-support status of a QC finding.**

**Use this definition in the index/glossary:**

* **Yes: directly verifiable from source evidence (record fields/PDF text or valid REC mapping).**
* **Partial: some support, but full verification was not possible (for example missing/unreadable PDF).**
* **No: not verifiable from available source evidence (unsupported/ungrounded).**