**Work Description**

Build a secure, internal tool for our editors that converts curated notes into finalized microlearning content using a **two-step LangGraph pipeline**: **Generator** (toggle between **Claude Sonnet 4.5** via Anthropic or **Gemini 2.5 Pro** via Google AI Studio) → **Formatter** (always **Gemini 2.5 Flash**) → **Validator** → **(Formatter retry once, same prompt & original DRAFT\_1)** → final **plain-text** output. Editors will use a **single-page UI** with toggles/inputs, and the backend will expose one **POST /run** endpoint. Prompts and outputs are **plain text** (no JSON). Host on **Cloud Run**; start with a simple password gate, with a path to Google IAP/SSO later.

**Milestone 1 — LangGraph Engine (Backend only)**

**Goal:** Implement the production-ready pipeline and API with tests and containerization.

**Scope**

* **Backend:** Python **FastAPI** with one endpoint POST /run.
* **Orchestration:** **LangGraph** flow  
  load\_prompts → generator (Claude or Gemini Pro) → formatter (Gemini Flash) → validator → (formatter retry ≤1 using same prompt & original DRAFT\_1) → done/fail.
* **Prompts (plain text):**
  + prompts/mcq.generator.template.txt (uses placeholders {{TEXT\_TO\_ANALYZE}}, {{NUM\_QUESTIONS}}, {{FOCUS\_AREAS}})
  + prompts/mcq.formatter.txt
  + prompts/nonmcq.generator.txt (uses placeholders {{TEXT\_TO\_ANALYZE}}, {{NUM\_QUESTIONS}}, {{FOCUS\_AREAS}})
  + prompts/nonmcq.formatter.txt
* **Validator (deterministic, code—not LLM):**
  + MCQ: Title → vignette → options A..D(/E) → Correct Answer/Answer: → Explanation → Analysis of Other Options (per-letter) → Key Insights.
  + NMCQ: Clinical Vignette n: … → vignette → items 1. <Type>: (True/False, Yes/No, Drop Down) each with Answer: + Explanation:; Drop Down requires ≥2 options before Answer:.
  + Validators are aligned with our downstream parsers (accept A) or A., Answer: or Correct Answer:, header variants).
* **Model integrations:**
  + **Anthropic** (Claude Sonnet 4.5) via official SDK.
  + **Google AI Studio** (Gemini 2.5 Pro & 2.5 Flash) via @google/generative-ai/google-generativeai SDKs.
  + Gemini 2.5 PRO calls temperature=0.51, top\_p=0.95; Flash temperature=.51, top\_p=.95; timeouts & basic retry on 429/5xx.
  + Input size guard: e.g., 150k chars; return a friendly error if exceeded.
  + Cap tokens to avoid misuse
* **Config / Secrets:**
  + Env vars: GOOGLE\_API\_KEY, ANTHROPIC\_API\_KEY, APP\_SECRET, optional EDITOR\_PASSWORD, MAX\_FORMATTER\_RETRIES=1 (default 1), optional GEMINI\_PRO, GEMINI\_FLASH, CLAUDE\_MODEL.
  + Keys stored in **Secret Manager**; not exposed to browser.
* **Deliverables:**
  + Source repo with app.py, pipeline.py, , tests, Dockerfile, requirements.txt/pyproject.toml.
  + **Tests:** unit tests for validators, golden tests for MCQ/NMCQ, negative cases; simple CLI script for local runs.
  + **Observability:** structured logging (model IDs, latency, attempts, validator errors), /healthz and /version (includes prompt file hashes & retry cap).
  + **Container** builds and runs locally.
* **Acceptance Criteria:**
  + POST /run returns validated **plain-text** output; on failure after one retry, returns **422** with error list + partial text.
  + Validators pass our provided samples; downstream parsers load the generated files without errors.
  + No secrets in client; env-configurable; container builds cleanly. Configurable temp, top\_p

**Milestone 2 — Single-Page UI + Integration**

**Goal:** Provide a simple, secure editor-facing page and wire it to the backend.

**Scope**

* **UI (single page, minimal styling):**
  + Toggle: **MCQ / Non-MCQ**
  + Toggle: **Generator** = **Claude Sonnet 4.5** | **Gemini 2.5 Pro**
  + Inputs: **Notes (textarea)**, **# Questions** (MCQ and NMCQ), **Focus Area** (optional)

UI knobs → prompt injection: Ensure {{NUM\_QUESTIONS}} and {{FOCUS\_AREAS}} are injected into both MCQ and NMCQ generator template, and {{TEXT\_TO\_ANALYZE}} takes the Notes text.

* + **Run** button; **read-only** output box (plain text); download as .txt.
* Auth

○ Don’t send a static password in the Authorization header from the browser. Prefer either:

IAP (Google SSO) asap (best) or

Minimal login form + HttpOnly session cookie (server verifies; no secrets in JS).

Keep CORS locked and add basic rate-limits.

* **Wiring:** UI posts JSON to POST /run; show success text or validation errors (with partial).
* **Hosting:** Serve UI and API from the **same Cloud Run** service (one container).
* **Deliverables:**
  + static/index.html (minimal Next.js/React if possible) + integration JS.
  + Deployment manifest/notes for Cloud Run, CORS lock-down, request size limit.
  + README updates for env, deploy, and IAP migration.
* **Acceptance Criteria:**
  + Editors can submit inputs, select model/content type, and receive final validated **plain-text** results.
  + Password gate active; keys remain server-side; logs show request/latency/attempts.
  + Deployed Cloud Run URL provided and verified.

**What we’ll provide**

* Sample prompts and target formats (MCQ & NMCQ).
* Sample inputs for golden tests.
* Notes on downstream parser expectations (already reflected in validators).

**Validator example (can share downstream flow that use python scripts to scape text file to create csv file if needed)**

**MCQ — Deterministic Rules (single-best-answer)**

**Required order**

1. **Title line** — Question <n> - <title> (hyphen can be -, --, en/em dash)
   * **Regex: ^Question\s+\d+(?:\s\*[-–—]{1,2}\s\*.+)?$**
2. **Vignette/Stem** — ≥1 nonempty line(s) after the title and **before** options.
   * **(A line ending with ? is recommended but not required.)**
3. **Options block** — **4–5** contiguous lines, each:
   * **A)/A. … D)/D. (and optional E)/E.)**
   * **Regex per line: ^[A-E][\)\.]\s+.+$**
4. **Correct answer** — exactly one line:
   * **Correct Answer: <A-E> or Answer: <A-E>**
5. **Explanation header** — one of:
   * **Explanation of the Correct Answer: or Explanation:**
   * **Followed by ≥1 nonempty line.**
6. **Analysis of other options** — header must be one of:
   * **Analysis of the Other Options (Distractors):**
   * **Analysis of Other Options:**
   * **Distractors:**
   * **For each option letter present in (3), a paragraph starting with that letter**
     + **e.g., A) … / A. …**
7. **Key Insights** — header Key Insights: followed by nonempty text.

**Other**

* Plain text only; blank lines allowed between sections.
* No duplicate Correct Answer: or duplicate section headers.

**NMCQ — Deterministic Rules (vignette + mixed question types)**

**Required order**

1. **Title line** — Clinical Vignette <n>: <title>
   * **Regex: ^Clinical Vignette\s+\d+:\s+.+$**
2. **Vignette body** — ≥1 nonempty line(s).
3. **Questions and Answers:** (recommended; optional)
   * **If present: ^Questions and Answers:\s\*$**

**Question items (one or more)**

* Each starts with:  
  ^<number>\.\s\*(True/False|Yes/No|Drop Down Question[s]?|Drop-?Down(?: Question[s]?)?)\s\*:\s\*<text>
* Then must include:
  + Answer: <value> (per type rules below)
  + Explanation: <nonempty text>
* **Type rules**
  + **True/False** → Answer: must be True or False
  + **Yes/No** → Answer: must be Yes or No
  + **Drop Down** → **≥2 options listed before Answer:**
    - Either line-by-line options (any nonempty lines not starting with a new item/Answer:/Explanation:)
    - **or** an Options: a, b, c line (commas or | separators)