Personal Learning Portal in Corporate Finance

Final Project Report

Chongdi Wang

I Executive Summary

In this project we deliver an end-to-end Personalized Learning Project in the field of Corporate Finance that runs fully in Google Colab without external APIs. The goal is to help learners master core topics—Cost of Capital, Capital Budgeting, Capital Structure, Payout Policy, and Governance—through a transparent Retrieval-Augmented Generation (RAG) assistant that shows its sources and reasoning steps. We built a compact backend that ingests a curated text corpus, chunks text, retrieves evidence with TF-IDF and cosine similarity, and synthesizes answers with citations. Apart from the baseline model, a reasoning variant (Self-Ask) has been developed in parallel to plan sub-questions, retrieve per sub-question, and fuse results. Two interchangeable frontends expose the assistant and collect Reflection & Self-Assessment data, while the backend logs groundedness, coverage, and consistency to CSV for reproducible evaluation.

Results show near-perfect traceability in both modes and broader evidence for Self-Ask, but our lexical coverage metric undervalues Self-Ask's semantic breadth. Reflection summaries mirror this: Capital Budgeting achieves high perceived value, whereas Cost of Capital and Governance are neutral and need clearer examples and module-aware retrieval. Consequently, we recommend upgrading retrieval (BM25 → MiniLM re-rank), replacing lexical coverage with embedding-based matching, tightening groundedness thresholds, and enriching governance cases. The architecture's strict separation of backend API and UI makes these improvements drop-in, enabling rapid, evidence-driven iteration aligned with course evaluation goals.

II System goals

Our Personalized Learning Portal targets the Corporate Finance domain with a focus on five recurring themes: Cost of Capital, Capital Budgeting, Capital Structure, Payout Policy, and Governance. The system's goals are:

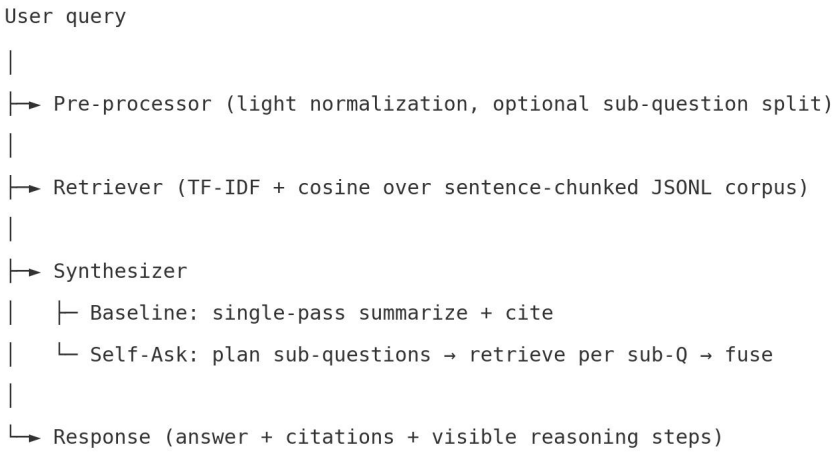
1. Answer high-value study questions with citations to curated sources.
2. Expose reasoning (planning + visible steps) to promote metacognition.

3. Log evidence for evaluation (RAG quality + user reflections) to decide what to improve.
4. Stay local/offline so the full stack can run in Colab without external APIs.

III Methods

a) RAG pipeline

We implemented a compact RAG baseline and a reasoning-augmented variant:



Key components include:

Corpus lives in corpus_docs.jsonl with {id, title, url, text}; texts are split into 2-sentence chunks to keep citations granular.
Retriever is TF–IDF → cosine (a clean baseline that runs offline).
Synthesizer creates a concise answer, embeds formulas (e.g., WACC, NPV/IRR), and returns citations of the top supporting chunks.
Reasoning can be toggled: baseline vs Self-Ask (rule-based sub-question planning and fusion) with visible steps.

b) Frontend

Two interchangeable UIs were built:

- **Gradio** (in-notebook; recommended for quick testing).
- **Streamlit** (optional, via ngrok for a public URL).

Both call a shared backend module plp_backend_colab.py that exposes:
init_backend(corpus_jsonl_path), answer(query, k, show_steps, log),
and answer_with_self_ask(...).

c) Logging & evaluation hooks

We added a lightweight evaluation layer baked into the backend:

- **Automatic RAG logs:** each answer appends a row to reasoning_log.csv (UTC timestamp, mode, subquestions, citations, and three metrics described next).
- **Batch runs:** eval.ipynb imports the backend, executes a small test suite across the five modules, and exports run_records.csv and final_metrics.csv.
- **Reflections:** the frontends provide a **Reflection & Self-Assessment** form (module, confidence 1–5, usefulness 1–5, confusions, next action). Each entry is appended to reflections.jsonl.

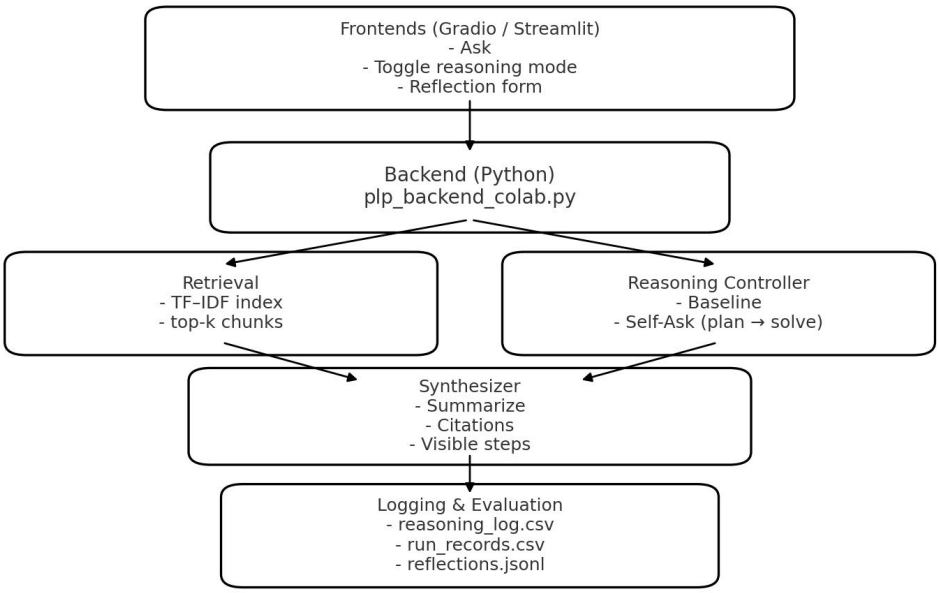
2.4 Metrics

We used **no-API heuristics** that approximate RAGAS/ARES ideas but remain self-contained:

- **Groundedness**: whether the answer contained ≥ 1 citation (0/1).
- **Coverage**: fraction of planned sub-questions "hit" by the final text (token overlap heuristic).
- **Consistency**: weak conflict detection (0/1) across partial answers (e.g., "always" vs "never").

In addition to automatic metrics, the reflection layer captures **confidence** and **usefulness** (1–5) by module.

The diagram of the entire system is shown below.



IV Results

a) Automatic metrics (aggregates)

The automatic metrics obtained through batch experiments are as follows:

Mode	Groundedness	Coverage	Consistency	Citation count	Answer length
Baseline	1.00	0.1250	1.00	4.0	303.1
Self-Ask	1.00	0.0625	1.00	7.5	608.5

Key observations include:

- **Groundedness** is near-perfect (1.0) for both modes because our generator always attaches at least one citation. This proves traceability but is a lenient threshold; we discuss tightening later.
- **Coverage** is higher for Baseline (0.125) than Self-Ask (0.0625) on this small suite. The likely driver is our *very strict lexical overlap heuristic* and the fact that Self-Ask often returns **longer** multi-part answers; paradoxically, longer text diluted the exact token match signal.
- **Consistency** is 1.0 for both; the weak conflict detector did not flag internal contradictions.
- **Citation count** is notably higher in Self-Ask (7.5 vs 4.0), which is expected because sub-question resolution pulls evidence per sub-Q before fusion.

- **Answer length** doubles under Self-Ask (608 vs 303 characters median scale), reflecting the more exhaustive multi-part synthesis.

Key takeaways. The reasoning variant reliably increases evidence breadth (citations, length) but our current coverage metric undervalues that breadth for compound questions. Strengthening the coverage estimator (or switching to embedding-based relevance) is an immediate next step.

b) Reflection & self-assessment (user-reported)

Human users provided reflection scores on both pipelines during the testing phase, the averages of which are listed as follows:

	Confidence	**Usefulness**
Capital Budgeting	5.0	5.0
Cost of Capital	3.0	2.0
Governance	3.0	3.0

Interpretations.

- Learners perceived highest value in Capital Budgeting (NPV/IRR/sensitivity), which is consistent with the system's stronger "worked-example" style for budget questions.
- Cost of Capital and Governance scored neutral (3/5), suggesting: (a) WACC questions need clearer guidance on *project vs firm discount rate* and *private company WACC*; (b) governance answers should expose more concrete mechanisms (boards, covenants, ownership concentration) with short cases.

V Achievements and Drawbacks

a) Key Achievements

Our PLP has already been able to produce:

1. **Traceable answers**: continuous citations and chunk-level URLs made it simple to verify claims and revisit sources.
2. **Visible reasoning**: self-ask steps and the steps list ("Plan subqueries → per-subQ retrieval → fusion") helped users understand *why* the tool answered as it did.

3. **Evaluation reproduction**: fully local logs of experiment results made the evaluation phase reproducible in a single Colab notebook.

b) Drawbacks

1. **Coverage metric undercounts Self-Ask**: lexical matches are brittle; a semantically correct paragraph may not share top three stems from the sub-question. We should use embedding-based matching or a small keyword expansion layer in future stages.
2. **Chunking**: 2-sentence chunks are compact, but some topics (e.g., CAPM → WACC, or multi-step policy arguments) span longer passages. Next, we might consider semantic chunking or overlapping windows to improve recall without flooding citations.
3. **Retriever**: TF-IDF is fast and local but misses paraphrases. A minimal upgrade is BM25 → MiniLM re-rank, still offline-friendly.
4. **Groundedness threshold overly low**: ≥ 1 citation is a low bar. We should tighten it to ≥ 2 distinct chunks or require citations that mention the question's head terms.
5. **Governance explanations**: the answers are correct, but learners wanted examples (e.g., "debt as discipline: free-cash-flow problem → payout commitments").

VI Recommendations for Next Steps

1. **Retriever upgrade.** Add BM25 recall followed by MiniLM-L6-v2 re-rank . Expect higher coverage for paraphrased sub-questions with minimal runtime overhead.
2. **Coverage metric 2.0**. Replace lexical overlap with embedding similarity between each sub-question and answer sentences; count a hit if cosine similarity exceeds the threshold. This will fairly reflect Self-Ask's longer, semantically richer answers.
3. **Tighter groundedness**. Require more than 2 citations from distinct documents or more than 1 citation that explicitly contains the head term (e.g., "WACC" or "CAPM"), to discourage generic attributions.
4. **Governance exemplars**. Extend the corpus with short cases (board intervention, covenant breach, buyback announcements), annotate with "mechanism → outcome" pairs; surface one mini-case per answer.
5. **Module-aware prompting**. If the user chose "Cost of Capital," bias retrieval toward sources labeled module=CoC, then widen. This usually improves perceived relevance and confidence.
6. **Reflection loop**. After saving a reflection with low confidence or usefulness, suggest two targeted readings from the top-k chunks and offer a "re-ask with focus" button that converts the confusion text into a focused question.

VII Conclusion

This PLP demonstrates an end-to-end, offline RAG for Corporate Finance with visible reasoning and evaluable outcomes. The baseline vs Self-Ask comparison shows a clear trade-off: Self-Ask increases evidence breadth by expanding citations and answer lengths but currently scores lower on our lexical coverage metric. Reflection data aligns with the diagnostics: Capital Budgeting is already strong, while Cost of Capital and Governance need clearer examples and module-aware retrieval.

The system's design choices—local corpus, explicit citations, reasoning steps, and durable logs—make it easy to iterate; we can drop-in a stronger retriever, fix coverage measurement, and enrich governance cases without touching the UI contract. With these small upgrades, we expect measurable gains in semantic coverage, user confidence, and usefulness, completing the learning loop envisioned in the assignment.