

Academic Advising AI Assistant Using Retrieval-Augmented Generation

BANA 6620 — University of Colorado Denver

Authors: Abinaya Krishnamoorthy and Daniela Salas Rios

Date: October 8, 2025

Abstract

This project builds a Streamlit-based Academic Advising Assistant using Retrieval-Augmented Generation (RAG) to answer questions **strictly from instructor-provided CU Denver Business School materials**. We ingested five datasets—Degree Plan Templates, 4-Year Plans (2024–25), Business Course Catalog Descriptions, Minors, and Fall 2005 Class List—spanning Word, Excel, and PDF. Documents were chunked (~1,000 characters) and embedded into **Chroma** vector stores using **mxbai-embed-large** (per-dataset indexes) and **nomic-embed-text** (a unified combined index), both via Ollama. In the submitted app, users select a dataset; the system retrieves top-k relevant chunks and prompts **Llama 3.2** to produce concise answers. In a parallel prototype, a single combined Chroma index supports cross-dataset queries with similarity/MMR retrieval, a retrieval-only fallback, and timeouts. Outcomes: a working RAG pipeline, functioning answers for core advising queries, and clear next steps—real-user testing to tune chunking/k and stronger workflow consistency for reproducibility.

Introduction

Academic advising information is dispersed across degree plans, catalogs, minor sheets, and historical class lists, often in different formats. Manual search is slow and error-prone. RAG addresses this by retrieving relevant passages from a fixed corpus and generating answers grounded in evidence. This project consolidates course skills—Python, LangChain, embeddings, vector databases, and prompt design—into an advising assistant limited to the instructor’s corpus. We implemented two complementary variants: (1) a **dataset-scoped** app that retrieves from one index at a time (simple and reproducible), and (2) a **unified-index** app that queries a combined store with configurable retrieval options while remaining within the same corpus.

Methods

Data & scope. We used only instructor-provided materials: Degree Plan Templates (Excel), 4-Year Plans (Word), Business Course Catalog Descriptions (Word), Minors (PDF), and Fall 2005 Class List (Excel).

Ingestion & preprocessing. Excel sheets were converted to one Document per row with normalized headers and key fields retained (course, term, instructor). Word documents were parsed with Unstructured/Docx2txt; PDFs with PyPDFLoader. All Documents carry source metadata and, where relevant, sheet and row_index.

Chunking & embeddings. Text was split into ~1,000-character chunks (Excel rows ~800) using LangChain splitters to balance recall and precision. Indexes were embedded

with **mxbai-embed-large** (per-dataset) or **nomic-embed-text**(unified), and persisted in Chroma.

Retrieval & generation. The dataset-scoped app uses similarity search with adjustable **k** (default 8). The unified-index app supports similarity or MMR ($k/\text{fetch_k}/\lambda$) plus a retrieval-only fallback and timeout guardrails. **Llama 3.2** (Ollama) generates concise answers from retrieved context; if evidence is missing, the app is instructed to say so.

Application. Streamlit provides a minimal interface for querying and viewing answers, with an optional Sources panel. Repository organization separates data/, ingest/, vectorstores/, RAG/, and outputs/ for clarity.

Output Examples (selected)

Q1. *What are the graduation requirements for the Accounting major?*

A. The app reports that explicit graduation requirements are not stated in the retrieved context; it infers an Accounting specialization of 6 credits within the Major Core and notes that other requirements may exist but are not present in the corpus snippet used.

Q2. *Show the 8-semester plan for the Management major.*

A. The app compiles an 8-semester outline (e.g., MGMT/BANA/ISMG/BLAW sequences with credits) from plan snippets in the corpus; exact sequencing may vary by catalog year.

Q3. *Provide the official course description for ACCT 3220.*

A. Returns the catalog description, including prerequisites (ACCT 2220, BANA 2010, both \geq C-), cross-listing (ACCT 6031), credit hours (3), and grading basis.

Q4. *Compare the Entrepreneurship vs. Finance minors (credits, required courses).*

A. Both total 15 credits (5 courses). Entrepreneurship specifies ENTP 3200/3230/3299 plus two ENTP electives; Finance requires FNCE 3000/3500/3600 (or alternate with approval) plus two additional upper-division FNCE courses.

(Additional examples with screenshots are provided in Appendix A.)

Conclusion

We delivered two variants: a **dataset-scoped app** that was stable and reproducible, and a **unified-index app** that added flexibility (similarity/MMR retrieval, retrieval-only fallback, timeout guardrails) while remaining within the corpus. The system is **not perfect**: with highly formatted or sparsely covered documents it often returned “information not available in the context,” which preserves grounding but can yield incomplete or generic answers when chunking is suboptimal. Performance was sensitive to **document structure** and to time invested in chunking and testing (e.g., formatted .docx, multi-column PDFs, wide Excel), affecting recall and completeness. Operationally, vector stores can be empty or misplaced if ingestion paths are wrong; without health checks, this only shows up at query time. No formal accuracy metrics

or **real-user testing** were conducted, so current validation is qualitative. Next steps include short user-testing sessions to calibrate per-dataset chunk size/overlap and retrieval **k**, adding **inline citations** ([1], [2] with a Sources panel) to improve transparency and debugging, and lightweight evaluation sheets to guide tuning.

References

University of Colorado Denver Business School. (2024–2025). *Academic advising materials: Degree plan templates; 4-year plans (2024–2025); business course catalog descriptions; minors; Fall 2005 class list* [Unpublished course datasets].

Wang, Z. (2025). *Pre-recorded class sessions* [Course videos]. BANA 6620, University of Colorado Denver.

Appendices (supplementary)

Appendix A — Additional Output Examples.

- Full answer text and screenshots for:
 - Management 8-semester plan (by semester),
 - MKTG 3100 sections/time/location for **Fall 2005** (note: project corpus year),
 - Additional prerequisite lookups and minor comparisons,
 - Queries that correctly returned “not in context.”

Appendix B — Implementation Details.

- Loaders and parameters; chunk sizes/overlaps per source; embedding models used per index; retrieval settings (k, MMR fetch_k/ λ); app guardrails (retrieval-only, timeouts).
- Repository layout and run steps (ingest → verify vector counts → run app).
- Health checks (index counts, model match), and typical ingestion pitfalls and fixes.

Appendix C — Evaluation & Next Steps.

- Lightweight metrics template (10–20 questions tracking *answerable Y/N*, *correct Y/N*, dataset, k, chunking).
- Plan for real-user testing sessions and parameter tuning.
- Design sketch for inline citations and a Sources panel mapping [n] to file/sheet/row.