



Knight Foundation School of Computer & Information Sciences

Spring 2026 Senior Design Project

Engineering & Computing

AdvisorAI: Intelligent Academic Advising Using RAG

Students: Anya Assing, Dabian Garnica, Fernando Azanza, Gabriela Saldana, Scarlett Herrera
Instructor/Faculty: Seyedmasoud Sadjadi, Florida International University

Problem

FIU students face a complex academic advising experience that creates bottlenecks for both students and advisors. The team identified concrete pain points on both sides

- **Students:** Repeatedly ask about academic dates, prerequisites, and elective eligibility; they cannot interpret the degree audit and they often don't know which electives satisfy requirements.
- **Advisors:** Manage 500+ students each during peak periods; spend time redirecting to links and re-explaining policies that exist online; lack availability at end of semesters.

An AI-driven advising assistant could automate these repetitive support tasks, improve information accessibility, and reserve human advising capacity for exception handling and high-impact decision support.

Current System

Students depend on multiple scattered and disconnected tools:

- **The Panther Degree Audit** (a text-heavy PDF most students don't know exists or can't interpret)
- **Static flowcharts** (where prerequisites are confused with general requirements and unlisted electives go unknown)
- **PSN for scheduling** which is nto always available for students to meet at peak demand
- **Scattered FIU portals for deadlines and policies.**

Advisors noted they wish students checked their audit before meetings; it answers most common questions, but its format isn't student-friendly.

Requirements

Functional:

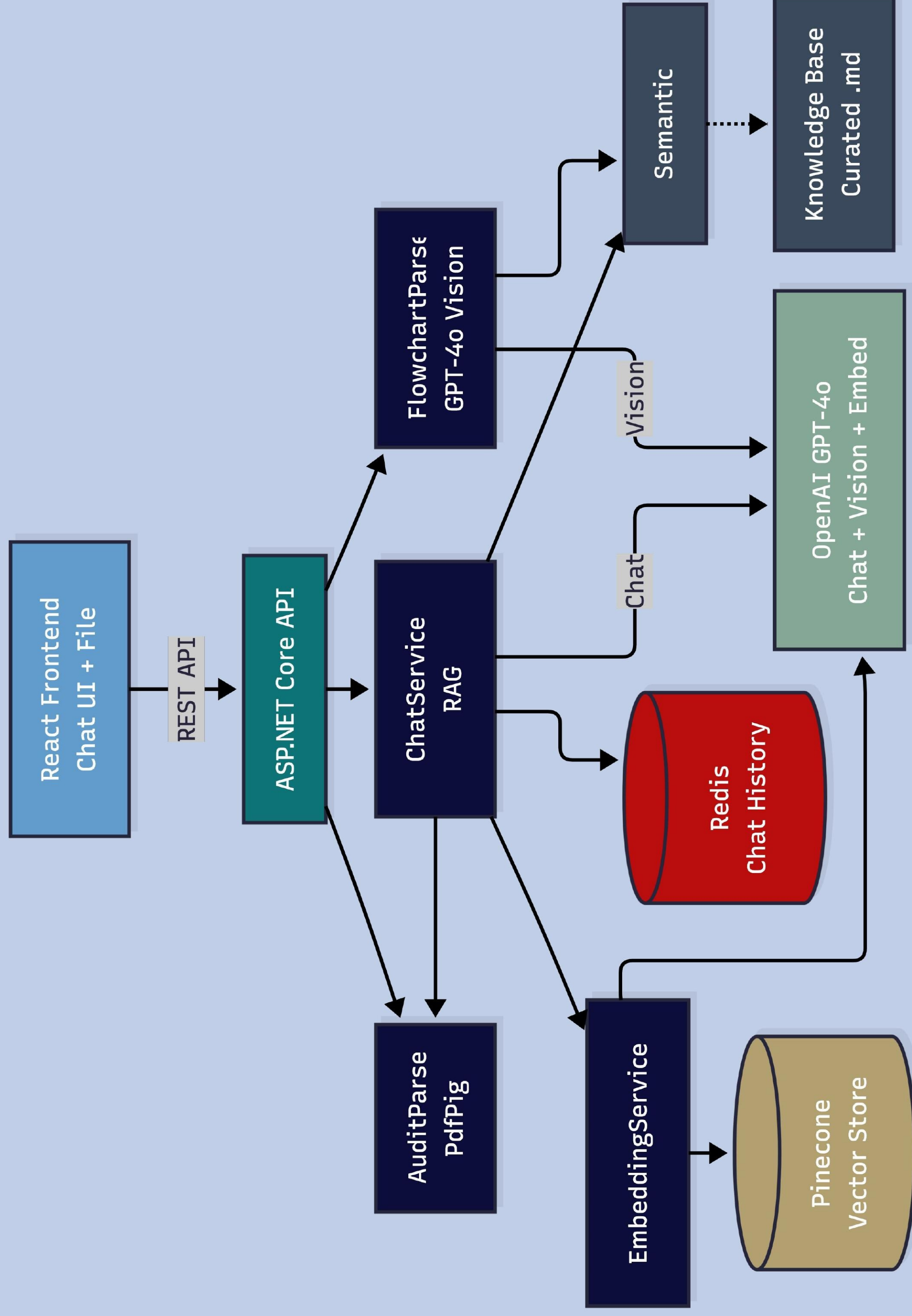
- Parse degree audit PDFs to extract requirements, satisfaction status, and courses taken
- Parse flowchart images via GPT-4o vision to extract courses, prerequisites, corequisites, and elective rules
- Store extracted curriculum data as vector embeddings in Pinecone for semantic retrieval
- Provide conversational AI advising combining retrieved context, uploaded documents, and conversation history
- Support dual response modes: audit-planning (course recommendations from uploaded records) and general-advising (policies, deadlines, resources)

Non-Functional:

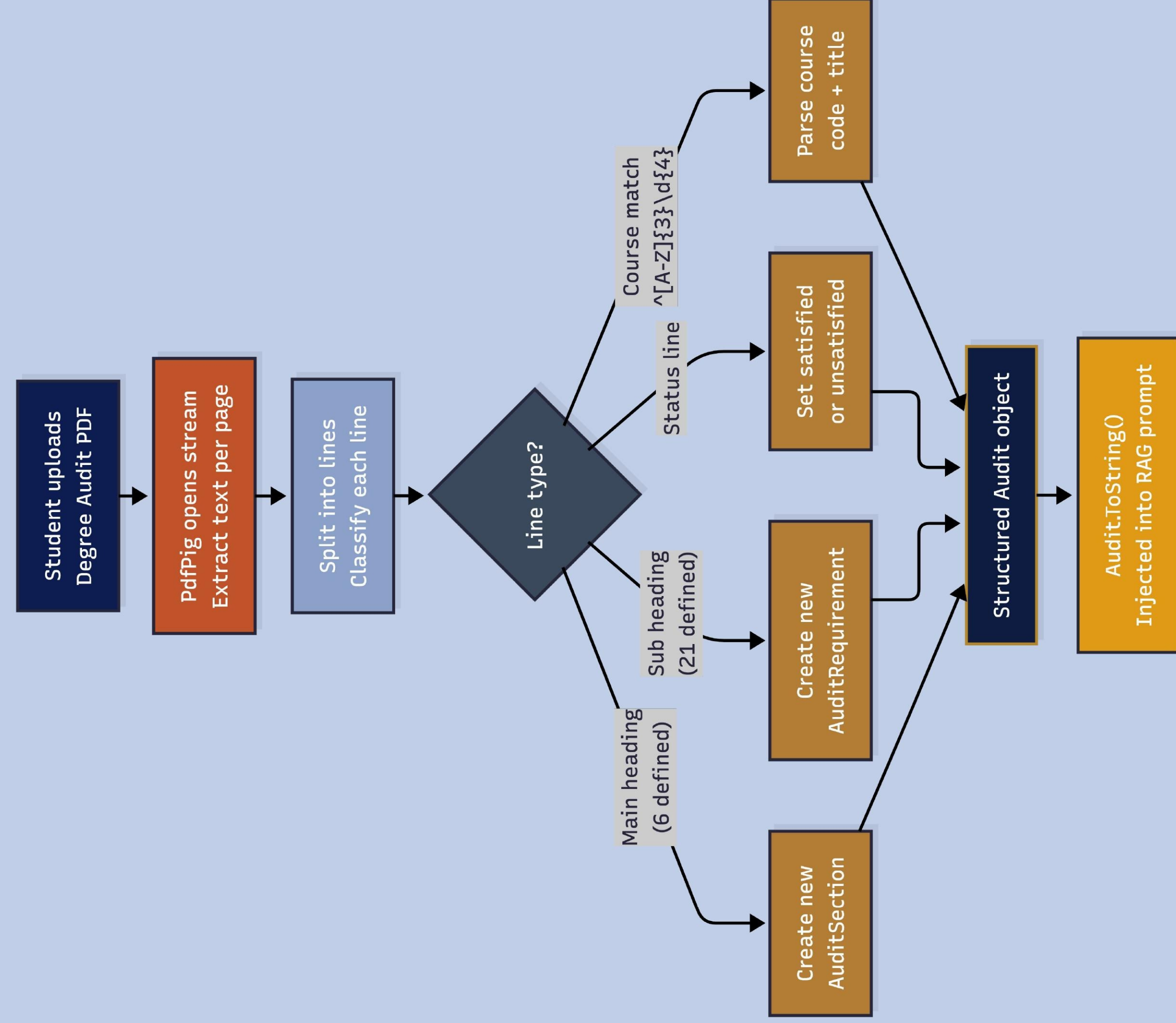
- FERPA-compliant ephemeral data handling, audits processed in memory, never persisted; conversations expire via Redis TTL
- Knowledge base includes source URLs to prevent hallucinations
- Interface-based dependency injection for extensible architecture

System

System Architecture



RAG Pipeline



Implementation

React: Component modularity + ecosystem (Vite for fast builds, Tailwind for scalable styling, React Router for navigation).



C# / ASP.NET Core: Strong typing, high performance, native dependency injection, enabling a maintainable MVC architecture extensible for future features.



Semantic Kernel: Key enabler and lightweight SDK that orchestrates all OpenAI calls (GPT-4o for chat and vision, text-embedding-3-small for embeddings) without boilerplate.



Redis: In-memory speed for per-session conversation caching, giving the model multi-turn context for grounded responses.

Verification

Verified through automated testing and manual validation. **MSTest** and **Moq** covered controller flows and audit parser logic, including parsing generated in-memory PDFs. Manual testing involved sending **real advising questions** and evaluating response quality, which drove iterative refinements to the embedding configuration and retrieved context for improved relevance and accuracy.

Summary

What we built: A RAG-powered AI advisor combining audit parsing (PdfPig), flowchart extraction (GPT-4o vision), and semantic search over curated FIU knowledge.

Why it matters: Handles the repetitive, informational questions consuming advisor time (500+ students each), freeing advisors for more high-impact advising time.

What shaped it: Six opportunity categories from real-world advisor interviews drove the knowledge base and architecture decisions.

Future Implementations: Automated audit type detection system, semester-by-semester schedule generation, and integration with live FIU systems (PSN, enrollment APIs).