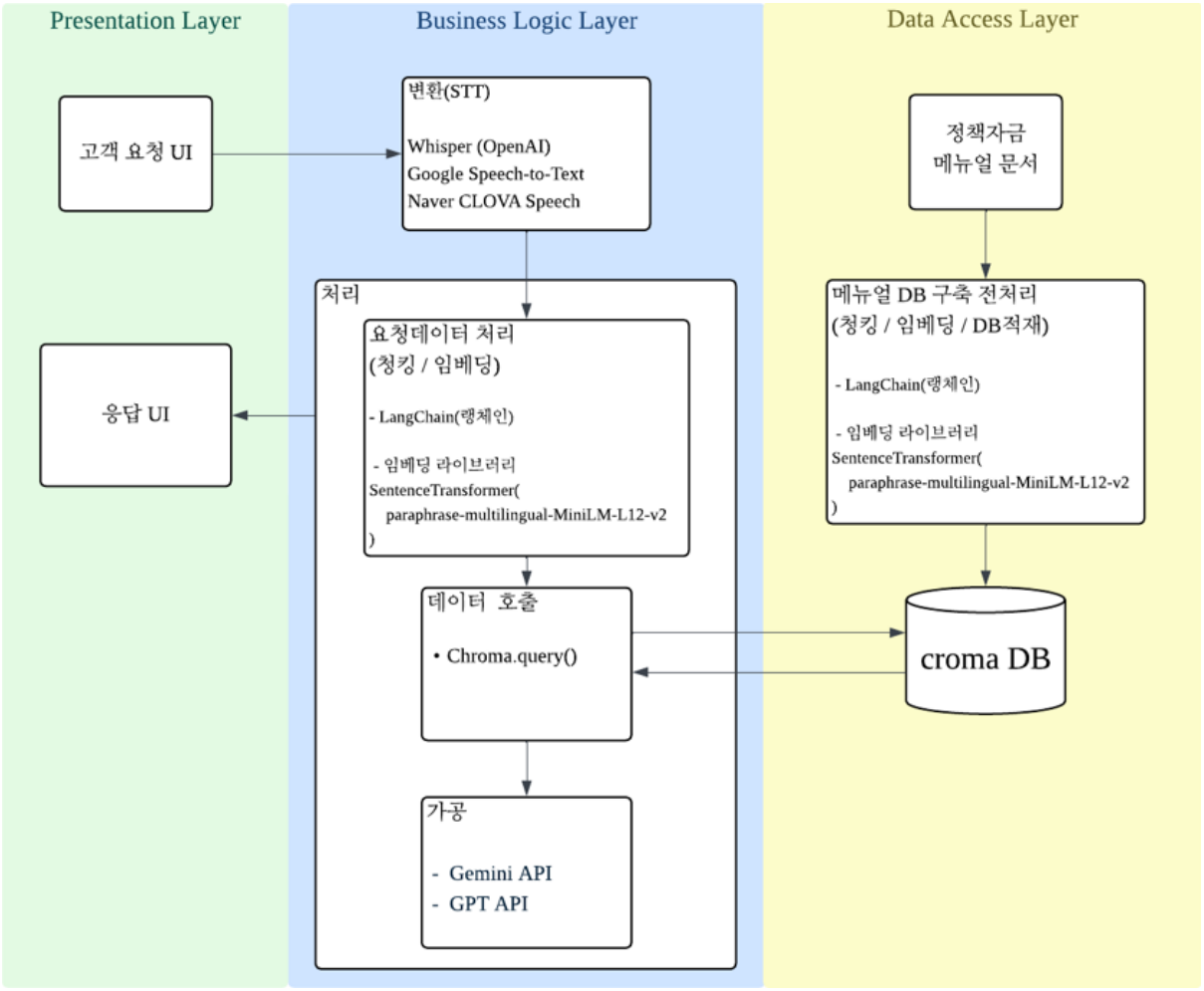
툭툭 시스템 구조 및 기술 선택 배경



툭툭 시스템 아키텍쳐

RAG와 파인튜닝

RAG

GPT는 그대로 두고 만들어둔 벡터 DB에서 관련 정보를 찾아 GPT에게 참고용으로 제공하는 방식입니다.

장점:

- 정책이 바뀌어도 문서만 업데이트하면 됨
- 빠르게 적용 가능하고 유연

하지만 일반적인 RAG에도 문제는 있음

- 문서를 통째로 GPT에 넣으면 답변 시간이 느리고 비용이 큼
- 상담사는 GPT의 긴 설명보다 간단하고 핵심만 정리된 요약을 더 원한다.

파인튜닝

GPT 모델 자체를 새로 훈련시켜 우리의 정책과 규칙을 모델 안에 미리 넣어두는 방식입니다.

장점:

- 질문이 들어오면 자동으로 정답을 말해줄 수 있다는 점

하지만 현실은...

- 시간도 오래 걸리고, 비용도 많이 든다
- 정책이 바뀔 때마다 다시 학습시켜야 하고
- 모든 경우의 수를 미리 넣는 건 사실상 불가능

그래서 우리는!

우리는 RAG 방식을 선택하되 요약 중심으로 가볍게 만든 형태로 다시 설계

- 문서를 미리 잘게 나누고 요약
- 질문이 들어오면 벡터 검색으로 관련 요약만 뽑아옵니다
- 그걸 GPT가 자연스럽게 다듬어서 상담사에게 보여줍니다
- 상담사는 긴 답변 대신 필요한 정보만 빠르게 확인 가능!

우리가 지향하는 방향은 명확합니다

- GPT가 주도해서 답변하는 시스템이 아니라
- 상담사가 중심이 되는 시스템입니다.
- AI는 결정을 대신하는 게 아니라
- 더 잘 판단할 수 있도록 도와주는 도구가 됩니다.

결론

우리는 **모델 자체를 바꾸는 Fine-tuning 방식이 아닌**
외부 지식에 기반한 요약형 RAG 시스템을 선택했습니다.

- Fine-tuning은 문서가 바뀔 때마다 재학습이 필요하지만 RAG는 문서만 교체하면 바로 반영할 수 있습니다.
- GPT는 대신 판단하는 게 아니라 핵심을 정리해줘야 합니다.
- 우리는 AI가 답을 정해주는 게 아니라 필요한 정보만 뽑아 정리해 보여주는 방식을 택했습니다.