

SMARTHON 기획서

#1 팀 소개

팀명	NLP 원정대
팀장/팀원	최규민

#2 프로젝트 기획

프로젝트명	CRA Bot, Class Registration Assistant Bot
한 줄 소개	세종대학교 학생들의 수강신청 문제를 해결할 수강편람 기반 QA 챗봇

개발 동기 및 필요성	<p>대학교에 갓 입학한 새내기(혹은 신입생)가 가장 어려움을 겪는 구간은 대학교에만 존재하는 수강신청이라는 개념이라고 생각한다.</p> <p>이는</p> <ol style="list-style-type: none">1) 대학 문서의 문법에 익숙하지 않고2) 공통필수, 교양, 전공 등의 구분에 대한 이해가 없으며3) 이를 소개하는 수강편람이 너무 방대하고 복잡하기 때문 <p>이다.</p> <p>이러한 이유에서인지, 신입생이 아닌 3학기 이상 재학한 학부생 또한 세종대학교의 졸업요건과 커리큘럼에 대해 잘 모르는 경우가 많다.</p> <p>어렵고 방대하고 복잡한 수강편람보다, 학생이 직접 수강신청을 준비하는 과정에서 생기는 궁금한 점들을 쉽게 풀어서, 간단히 요약하여 소개하는 서비스가 있다면, 신입생 및 재학생들의 수강 계획 수립에 큰 도움이 될 것이다.</p>
기대효과 (발전가능성)	<p>(참고) 이 프로젝트를 통해 얻을 수 있는 기대효과, 혹은 이 프로젝트에 추가할 수 있는 기술, 또는 이를 통한 발전 가능성 등</p> <ul style="list-style-type: none">- 신입생 및 재학생들의 용이한 수강 계획 수립- 학과 별 커리큘럼 추가를 통한 전공별 상세한 수강 안내 추가 가능-

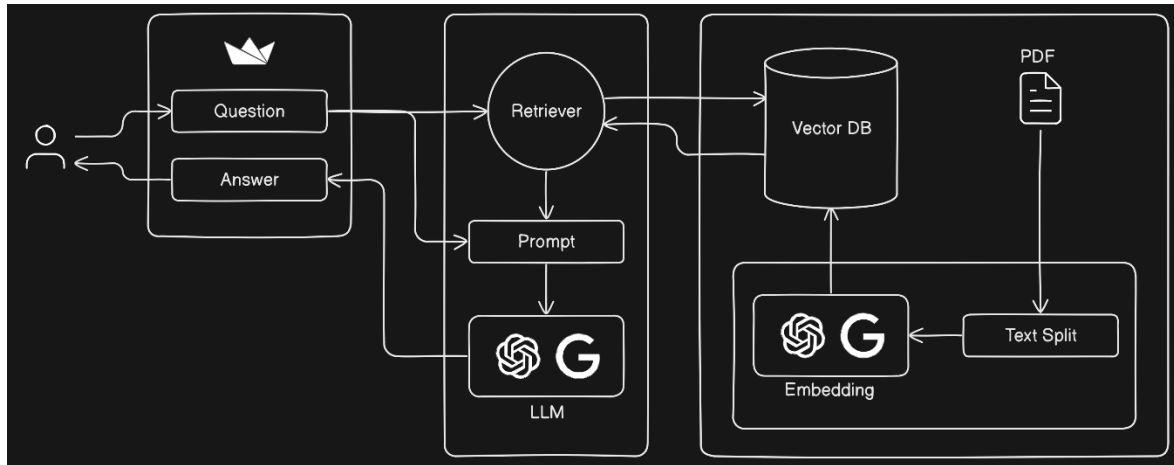
#2-1 사용할 AI와 학습 방법

기술 분야	RAG; Retrieval Augmented Generation
사용할 AI의 용도, 기능	<p>OpenAI API 혹은 Google Gemini API를 사용한다.</p> <p>즉, GPT 혹은 Gemini를 사용하며, 챗봇 시스템의 Backbone으로 사용한다.</p> <ol style="list-style-type: none">1) 임베딩: Split 된 PDF 본문을 벡터 DB에 임베드 하는 데에 사용2) 추론: 사용자의 질문 + 벡터DB에서 추출한 Context를 이용해 답변을 생성하는 데에 사용
학습 데이터 선정 및 활용 방법	<p>RAG는 Fine-tuning과 다르게 학습 데이터를 필요로 하지 않는다.</p> <p>대신, 해당되는 도메인의 Down-stream task에 특화되도록 답변 생성 시 참고할 정보를 별도의 벡터 Data base에서 발췌한다.</p> <p>따라서 본 프로젝트에서는 세종대학교 수강편람 PDF파일을 적당한 방식으로 Split 하여 벡터 DB에 저장할 것이다.</p> <p>현재 2025학년도 1학기 수강편람 PDF가 공개되지 않았으므로, 2024학년도 1학기 수강편람 PDF로 작업한다.</p> <p>2학기가 아닌 1학기 수강편람을 사용하는 이유는, 본 프로젝트가 타겟팅하는 주요 고객층이 대학 문서들의 문법에 익숙하지 않고, 커리큘럼에 대해 잘 모르는 신입생이기 때문이다.</p>

#2-2 프로젝트 구조도 (그림자료 및 설명)

프로젝트 구조도

(예시)

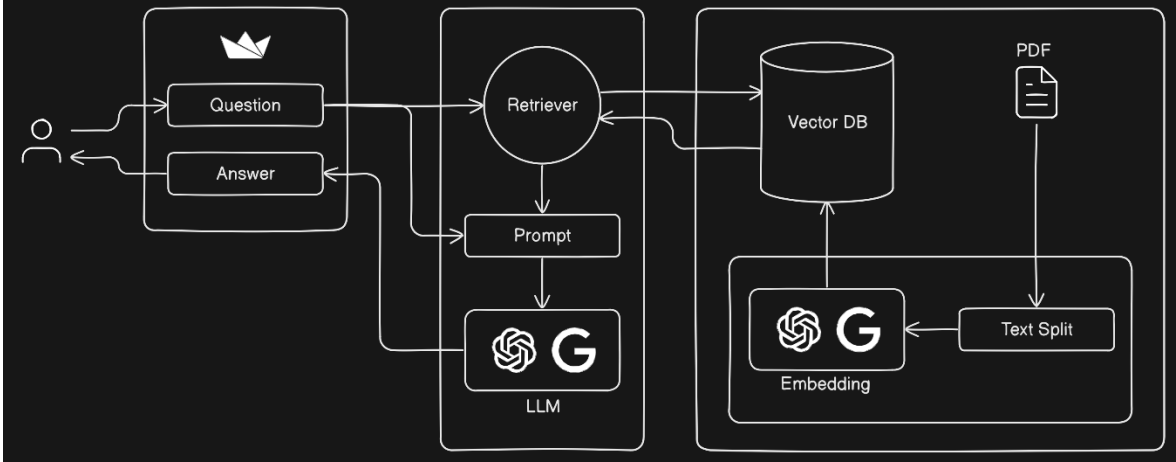


구조도에 대한 설명

1. 수강편람 PDF파일은 적당한 Chunk로 Split되어, 적당한 임베딩을 통해 벡터 DB에 등록된다.
2. 사용자가 Streamlit으로 구현된 UI에 질문을 입력하면, Retriever가 벡터DB에서 질문과 유사한 정보를 검색하고 이를 반환한다.
3. 반환된 정보는 Prompt에 추가되고, 이 Prompt가 LLM에게 입력된다.
4. LLM은 알맞은 답변을 생성하여 반환하고, 이 답변이 UI에 표시된다.

#3 목표 MVP

평가 배점에 포함되는 요소, 구체적이고 구현 가능하게 작성 바람.

최소 기능만 수행 시 프로젝트 구조도	
주요 기능과 MVP가 다른 점이 크게 없으므로 기존의 구조도를 첨부	
	
최소 기능 (MVP)	(예시) - 수강편람에 기초한 정보 반환 - UI 표시

#4 사전개발일정

※ 예시

날짜	목표 A 기획서 등 서류 완성	목표 B 벡터DB 구현 및 LLM 연결	목표 C UI 구현
1/9~1/12			
1/13~1/15			
1/16~1/17			
1/18~1/20			