

# 개발 과제: 내부 문서 검색을 위한 **RAG** 기반 **AI** 챗봇 프로토타입 구축

## 과제 목표

다양한 형태의 문서(docs, PDF, sheet, image)를 효율적으로 검색하고, 질문에 대한 답변을 생성해주는 **RAG(Retrieval-Augmented Generation)** 기반의 **AI** 챗봇을 개발합니다. 이 과정을 통해 분산된 사내 지식을 하나로 모아 업무 생산성을 극대화하는 경험을 시뮬레이션합니다.

## 핵심 요구사항

### 1. 문서 관리 시스템

- 지원 파일 형식: **PDF, DOCX, XLSX/CSV, PNG/JPG**
- 파일 업로드: 드래그앤드롭(Drag & Drop)을 지원하는 직관적인 UI
- 처리 상태 시각화: 파일별로 **대기중, 처리중, 완료, 실패**와 같은 처리 상태를 명확하게 표시
- 문서 관리: 업로드된 문서 목록을 확인하고, 개별 문서를 삭제하거나 재처리할 수 있는 기능

### 2. 문서 처리 파이프라인 (**RAG Pipeline**)

- 문서 업로드 → 텍스트 추출 (Pre-processing) → 의미 단위 청킹(Chunking) → 임베딩(Embedding) → 벡터 DB 저장으로 이어지는 자동화된 파이프라인을 구축합니다.
- 데이터베이스는 로컬 환경에 자유롭게 구성합니다.

### 3. **RAG** 기반 챗봇 인터페이스

- 채팅 UI: 사용자가 메시지를 입력하고 대화 기록을 확인할 수 있는 기본적인 채팅 인터페이스
- 답변 출처 표시: AI가 답변을 생성할 때 참고한 원본 문서(또는 문서의 일부)를 명시하여 신뢰성을 확보
- 스트리밍 응답: 사용자 경험 향상을 위해 답변이 생성되는 과정을 실시간으로 보여주는 스트리밍 방식 지원
- 대화 맥락 유지: 이전 대화 내용을 기억하고 이어지는 질문에 적절하게 답변하는 기능

## [Challenge] **RAG** 정확도 개선 방안 제시 및 구현

**RAG**의 성능은 단순한 파이프라인 구축만으로 보장되지 않습니다. 기본적인 **RAG** 아키텍처의 한계를 인식하고, 정확도를 높이기 위한 자신만의 아이디어를 최소 1개 이상 제시하고 프로토타입에 적용해 보세요.

## 기술 스택

- 프론트엔드, 백엔드, AI/ML 모델 등 모든 기술 스택은 자유롭게 선택 가능합니다.
- 선택한 기술 스택과 그 이유를 **README.md** 파일에 명확히 기재해주세요.

## 제출 사항

1. **GitHub** 저장소 링크:
  - 프로젝트 실행 방법을 상세히 기술한 **README.md** 파일을 반드시 포함해주세요.
  - **README.md**에는 아래 내용을 포함하여 작성해주시길 권장합니다.
    - 프로젝트 개요 및 실행 방법
    - 전체 시스템 아키텍처 다이어그램
    - 주요 기술 선택 이유
    - **RAG** 정확도 개선을 위한 고민의 과정과 구현 내용
    - 과제를 진행하며 마주했던 어려움과 해결 과정
2. 예제 문서: 과제 테스트에 사용한 **아카이브.zip** 또는 다른 예제 파일을 함께 제출해주세요.

## 과제 기간 및 문의

- 제출 기한: **9/10 (수) 오전 9시 (KST)**
- 질의응답: 과제 진행 중 궁금한 점이나 요구사항이 불명확한 경우, 주저하지 말고 이메일 회신으로 질문해주세요. (질문하는 능력 또한 중요한 평가 요소입니다.)