SK네트웍스 Family AI과정 7기

모델배포 시스템 구성도

□ 개요

- 산출물 단계 : 모델배포
- 평가 산출물 : 시스템 구성도
- 제출 일자 : 2025-03-28
- 깃허브 경로 : https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN07-FINAL-5Team
- 작성 팀원 : 김나예

개요	• AI 모의면접
	  ○ 본 시스템은 LLM을 활용한 AI 모의 면접 웹 애플리케이션으로, 사용자의 이력/포트폴리오 및 희망 기업/ 직무를 기반으로 개인에게 특화된 모의 면접을 제공합니다.
	• 특징
	  ○ Hybrid RAG 기술을 이용해 사용자의 구직 정보에 적합한 채용 공고를 추천합니다.
	  ○ LLM을 활용해 사용자의 구직 정보를 기반으로 예상 면접 질문을 생성하고, 답변에 대한 평가를 제공합니다.
	  ○ 면접 총평, 5개 영역에서의 평가 점수를 담은 종합 레포트와 면접 질문 별 피드백을 제공합니다.

구성 요소	• 사용자 인터페이스 (Frontend)
	  ○ Streamlit을 기반으로 구축
	  ○ EC2 인스턴스 내에서 Docker 컨테이너로 서버 실행
	  ○ PDF 파일 업로드, 모의면접, 종합 리포트를 포함한 인터페이스 제공
	  ○ 디바이스 연결성: 사용자 디바이스의 웹캠과 마이크를 이용
	• 백엔드 애플리케이션 (Backend)
	  ○ FastAPI로 백엔드 서버 구축
	  ○ EC2 인스턴스 내에서 Docker 컨테이너로 서버 실행
	  ○ 사용자 요청 처리 및 DB 통신 수행
	• 데이터 저장 및 처리
	  ○ PostgreSQL: 채용 공고 웹사이트로부터 크롤링한 채용 공고 상세 정보, 기업 정보, 면접후기 데이터 저장
	  ○ Elastic Search: Hybrid RAG의 구성요소로 유저 정보와 채용 공고 데이터 기반 검색 수행
	  ○ ChromaDB: Hybrid RAG의 구성요소로 벡터화 된 유저 정보와 채용 공고 데이터 기반 검색 수행

구성 요소	

	• **스토리지**
	  ○ Amazon S3: 클라이언트에서 업로드 한 PDF 파일과 채용 공고 이미지 파일을 저장
	• **LLM 에이전트**
	  ○ OpenAI gpt LLM API를 활용해 설계한 에이전트
	  ○ 면접 질문 생성 에이전트, 답변 평가 및 종합 레포트 생성 에이전트로 구성
	• **호스팅 및 배포 환경**
	  ○ Amazon EC2: 전체 시스템을 호스팅
	  ○ Docker: Streamlit 및 FastAPI 애플리케이션을 개별 컨테이너로 구성해 서버 실행
	• **Github**
	  ○ 코드 작성 및 개발자 간 협업과 버전 관리를 위한 저장소

데이터 흐름	Case 1 : 공고 추천 1. 유저가 Client(FE)를 통해 이력서/포트폴리오/자소서 PDF 파일 업로드 2. BE에서 Storage에 PDF 파일 저장 3. 파일 저장과 동시에 BE의 Document Parser를 통해 PDF의 텍스트 추출 4. 추출한 텍스트와 채용 공고 텍스트 데이터를 토큰화 후 임베딩 하여 VectorDB와 Retriever(ES)에 전달 5. 검색을 통해 결정된 Top_k Document ID를 BE에 반환 6. BE에서 RDB의 채용 공고 데이터 중 Top_k Document ID에 해당하는 채용 공고 정보를 Client에 반환 Case 2 : 면접 질문 생성 1. 유저가 Client(FE)를 통해 기업, 직무, 경력 여부 선택 2. 입력 받은 정보로 RDB에서 채용 공고 텍스트, 면접 후기 텍스트, 기출 질문 텍스트를 Select 하여 반환 3. 반환된 텍스트와 VectorDB, Elastic Search에서 Select한 이력서/포트폴리오/자소서 Parsing Data를 LLM에 전달해 질문 생성 요청

	4. 생성된 질문들을 BE에 반환하고 TTS 모듈로 전달 5. 변환된 음성을 Client에 반환 **Case 3 : 면접 답변 평가 & 종합 레포트 생성** 1. 유저가 디바이스 내장 마이크를 통해 발화 음성을 Client에 입력 2. BE에서 음성을 Binary Audio로 변환 후 STT 모듈에 요청 3. 변환된 텍스트는 RDB에 저장 4. BE에서 변환된 텍스트를 LLM에 전달한 후, VectorDB/Retriever의 구직 정보를 참조하여 권장 답변을 생성하고 사용자 답변을 평가하도록 요청 5. 생성된 답변과 평가는 RDB에 저장 6. RDB에서 면접 질문, 권장 답변, 답변 평가 텍스트를 Select 하여 BE에 반환 7. BE에서 평가 데이터를 Client에 반환 (종합 레포트)