



# 보고사 : RAG 시스템 구축 프로젝트 계획서

## 1. 프로젝트 개요

- 프로젝트명: 보고사
- 참여 기업: FnGuide

## 2. 비즈니스 문제 정의

- 비즈니스 문제 정의
  - 기존 고객 유지(이탈 가능성 감소) 및 신규 고객 유입
- AI를 활용한 해결 가능성
  - 기존 접근법
    - 단방향으로 제공되는 정보로 확장적인 정보 접근 불가(추후 질의 어려움)
    - 이용자의 금융 지식 수준에 따른 정보 격차 및 불평등 문제
    - 원하는 정보에 접근하는데 소요되는 시간, 비용 문제
  - 해결 가능성
    - RAG 시스템을 통한 양방향 상호작용으로 금융 정보 접근성 향상
    - 동적 상호작용이 가능한 정보 수집 툴 제공(사용자 피드백 실시간 반영)
    - 개인 맞춤형 응답을 통한 시간 및 비용 문제 개선

## 3. 프로젝트 목표

- 주요 목표: 사용자 경험 향상, 신뢰도 유지
  - 시스템의 답변 정확도 향상
- 세부 목표
  - 시스템 간의 차이점을 집중적으로 분석하여 지속 개선
  - 기본 RAG 시스템 vs 최종 RAG 시스템
    - 데이터 파싱 및 로딩
    - 임베딩 모델 테스트
    - 벡터 스토어 구축 방식 개선
    - advanced 및 modular rag 적용
    - 프롬프트 엔지니어링
    - 성능 지표 측정

## 4. 선행 연구 및 사례 분석

- 선행 연구 검토
  - AutoRAG를 이용한 금융 문서에 가장 최적화된RAG 시스템 구현에 관한 연구

- 유사 프로젝트 사례
  - <https://www.skelterlabs.com/blog/rag-securities> 신한투자증권 X 스퀔터랩스
  - [https://blog-ko.allganize.ai/alli-finance-llm-1/?utm\\_source=chatgpt.com](https://blog-ko.allganize.ai/alli-finance-llm-1/?utm_source=chatgpt.com)

## 5. 데이터 수집 및 준비

- 데이터 소스
  - [KIS Weekly 보고서](#)
- 데이터 수집 방법
  - 웹 자동화 방식을 이용한 PDF를 다운로드 및 사용 자동화
- 데이터 전처리
  - 복합 데이터
    - 파일의 특성에 맞는 최적 로더를 사용하여 처리
  - 시계열
    - 날짜를 기반으로 메타 데이터를 생성하여 바로 활용하는 방식이 있음(추후 더 고려가 필요)

## 6. AI 모델 개발 및 선택

- 관련 모델 검토/특징: LangChain, LangGraph, LangServe(Streamlit 대체) 등 사용
  - 세부 방식은 추후 논의(랭체인 라이브러리, LLM 모델, 허깅페이스 등 다양하게 테스트)
- 모델 평가 방법: AutoRAG(추후 변경 가능)

## 7. 프로젝트 일정 및 자원

- 프로젝트 일정
  - 2월
    - 2주차(2/10 ~ 2/16) : 텍스트 기반 Base 모델 개발, PDF 파싱
    - 3주차(2/17 ~ 2/23) : PDF 파싱, 벡터 DB 구축
    - 4주차(2/24 ~ 3/2) : 벡터 DB 구축, 문서 검색 성능 평가(AutoRAG)
  - 3월
    - 1주차(3/3 ~ 3/9) : 임베딩 최적화 후 Naive RAG 테스트(프롬프트 엔지니어링, 성능평가)
    - 2주차(3/10 ~ 3/16) : Advanced RAG 시스템 개발 후 테스트(프롬프트 엔지니어링, 성능평가)
    - 3주차(3/17 ~ 3/23) : Modular RAG 시스템 구축 및 배포(프롬프트 엔지니어링, 성능평가)
    - 4주차(3/24~3/26) : 발표 준비
- 필요 자원
  - LLM API Key(OpenAI, Anthropic, Upstage 등)
  - AutoRAG 구현을 위한 하드웨어 자원 or GCP
  - 서비스 확장을 위한 추가적인 데이터셋(API를 통해 확보할 수 있는 리포트 등)

## 8. 리스크 관리

- **주요 리스크 요인:** 신뢰도 문제, 비용 발생 문제
  1. 시스템이 제공하는 정보에 대한 정확도 및 신뢰도 문제 발생 가능
  2. 시스템 아키텍처에서 발생 가능한 LLM API 비용 발생이 과도할 수 있는 문제
- **리스크 완화 전략**
  - 신뢰도 문제
    - 수집된 데이터로 답변의 범위 한정
    - 관련도 높은 정보 출처 제공
    - 다양한 RAG 아키텍처를 통한 검색 정확도 향상
  - 비용 발생 문제
    - 유의미한 데이터 수집 및 가공
    - 벡터스토어 최적화

## 9. 결과 및 기대 효과

- **예상 결과:** 금융 리포트 분석 및 정보 제공 RAG 시스템 구축
  - 사용자 피드백을 반영하여 동적이고 맞춤형 응답을 제공하는 시스템 완성
  - 빠르고 정확한 정보 접근을 통해 사용자의 의사결정을 지원하는 시스템 구현
- **비즈니스 임팩트:** 고객 경험 개선, 신규 고객 유치 가능
  - **고객 경험 개선**
    - 고객이 더욱 빠르고 정확한 정보를 얻을 수 있으며, 불필요한 시간 절약 가능
    - 양방향 상호작용을 통해 고객의 피드백 실시간 반영
    - 고객의 요구에 맞는 맞춤형 서비스를 제공하여 만족도 향상
  - **신규 고객 유치**
    - 효율적이고 정확한 정보 제공을 통해 신규 고객 유치 효과 기대
  - **기존 고객 유지**
    - 정보의 접근성, 정확도 개선을 통해 고객의 이탈 가능성 감소 및 지속적인 고객 관계 유지
  - **차별화된 경쟁력 확보**
    - 금융 정보 제공에 있어 동적이고 개인화된 접근을 통한 경쟁사와의 차별화

## 10. 참고 문헌 및 레퍼런스

- **참고 문헌**
  - AutoRAG를 이용한 금융 문서에 가장 최적화된 RAG 시스템 구현에 관한 연구
  - Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
  - AutoRAG 평가 지표
- **레퍼런스**
  - 데이터셋 : KIS Weekly 보고서
  - 참고자료
    - <랭체인LangChain 노트> - LangChain 한국어 튜토리얼 🇰🇷

- <https://github.com/teddylee777/langchain-kr>
- [이토록 쉬운 RAG 시스템 구축을 위한 랭체인 실전 가이드](#)