

# LLM과 RAG를 활용한 코드 리뷰 자동화 어시스턴트

팀원

강보권 고상희 류영교 송기영 이준용

# 목차

0

## 제안서 소개

- 프로젝트의 목적 소개

1

## 프로젝트 주제 소개

- 프로젝트 주제 소개
- 과제 목적 및 필요성
- 기대효과 및 활용방안

2

## 과제 추진 방법 및 일정 소개

- 개발 환경 및 기술스택
- 시스템 전체 구조
- 회의 및 과제 추진 일정
- 예상 성과 소개

# 제안서 소개

## 추진배경

- GitHub Copilot를 비롯하여 AI를 활용한 코드 작성 및 검토에 대한 관심이 증가
- 코드 안전성과 품질을 효율적으로 개선하려는 기업 및 개발팀 수요가 증가
- RAG 기법을 활용하여 구체적이고 맥락있는 리뷰 피드백 제공 목표

## 기대효과

- 코드 품질 및 보안 향상에 기여할 수 있는 자동화된 코드 리뷰 시스템 구현 경험
- 대규모 언어 모델 및 RAG 기법을 개발 워크플로우에 적용하는 방법에 대한 이해도 제고
- 프로젝트 결과물을 기반으로 실무에서 AI 기반 코드 리뷰를 확장할 수 있는 토대 마련

Chapter 1

# 프로젝트 주제 소개

1

프로젝트 주제 소개

2

과제 목적 및 필요성

3

기대효과 및 활용방안

# 프로젝트 주제 소개

## LLM과 RAG를 활용한 코드 리뷰 자동화 어시스턴트란?

현재) 개발자들이 코드를 직접 리뷰해야 하므로 많은 시간이 걸린다.

해결) 코딩 컨벤션 문서를 학습한 어시스턴트가 즉시 코드 분석 및 개선점을 제공하여 리뷰 시간이 단축된다.

## 예시

```
def remove_duplicates(input_list):
    return list(set(input_list))
my_list = [1, 2, 2, 3, 4, 4, 5]
print(remove_duplicates(my_list))

def is_palindrome(s):
    s = s.replace(" ", "").lower()
    return s == s[::-1]
word = "A man a plan a canal Panama"
print(f"Is the word '{word}' a palindrome? {is_palindrome(word)}")
```

1. `remove_duplicates(input_list)`: 리스트에서 중복된 요소를 제거하고 유일한 값들로 이루어진 리스트를 반환.
2. `is_palindrome(s)`: 문자열에서 공백을 제거하고 소문자로 변환한 후, 앞뒤가 같은지 검사하여 회문 여부를 반환.

# 과제 목적 및 필요성

- 1 AI 기반 코드 작성 및 검토에 대한 관심 증가
- 2 코드 안전성과 품질 개선에 대한 기업 및 개발팀의 수요 증가
- 3 RAG 기법을 활용하여 맥락 있는 코드 리뷰 제공

# 기대효과 및 활용방안

## 기대효과

- 1 코드 품질 및 보안 향상에 기여할 수 있는 자동화된 코드 리뷰 시스템 구현 경험
- 2 대규모 언어 모델 및 RAG 기법을 개발 워크플로우에 적용하는 방법에 대한 이해도 제고

## 활용방안

- 1 프로젝트 결과물을 기반으로 실무에서 AI 기반 코드 리뷰를 확장할 수 있는 토대 마련

Chapter 2

# 과제 추진 방법 및 일정 소개

1

시스템 전체 구조

2

개발 환경 및 기술스택

3

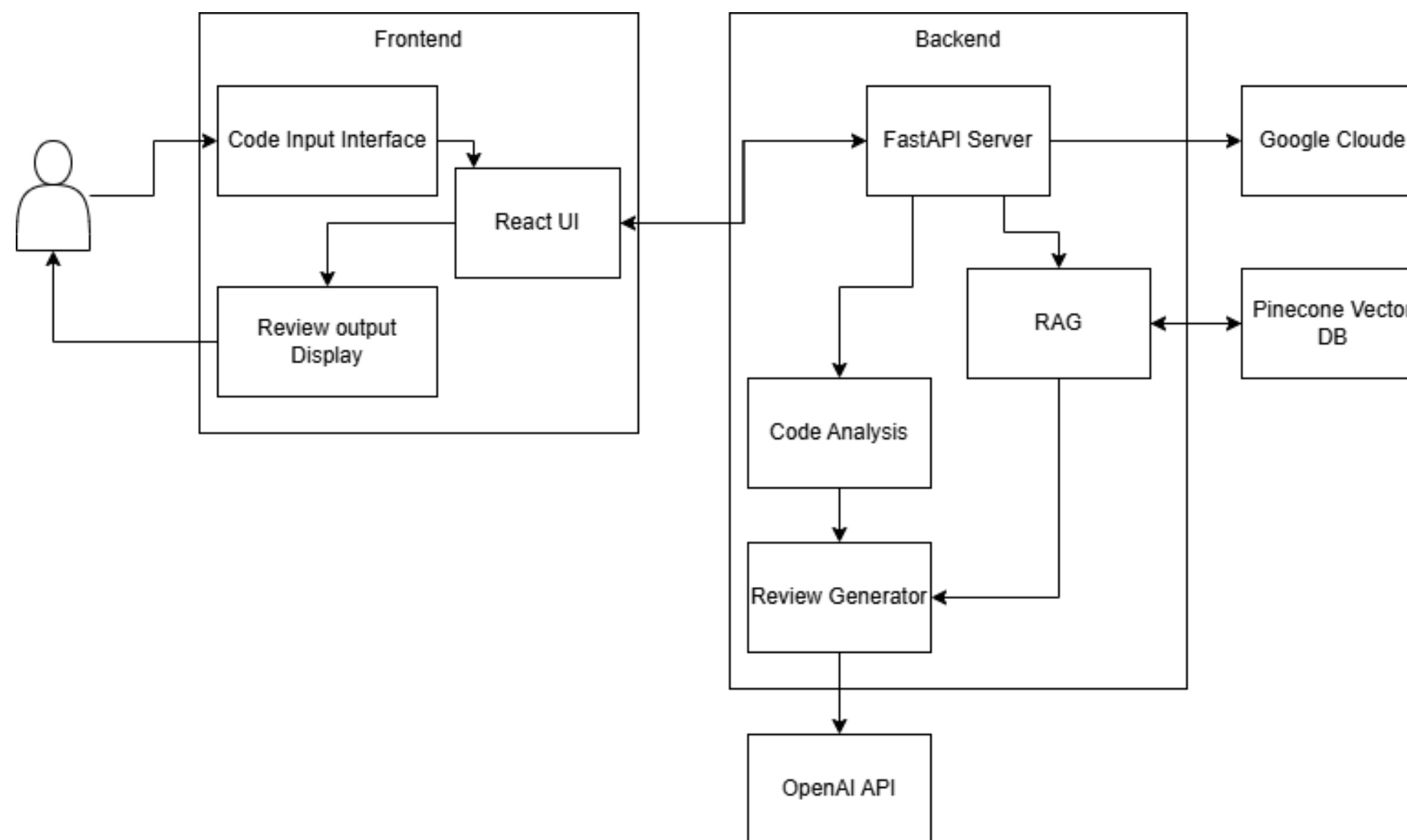
회의 및 과제 추진 일정

4

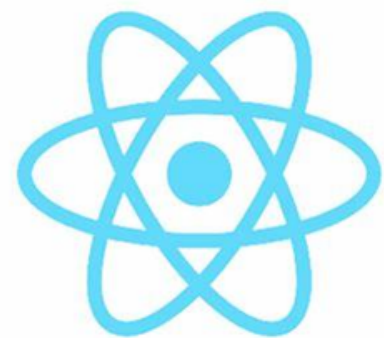
예상 성과 소개



# 시스템 전체 구조



# 개발 환경 및 기술스택



React

 FastAPI



Google Cloud

 OpenAI  
API



Pinecone

# 개발 환경 및 기술스택



# 회의 및 과제 추진 일정

1

매주 화요일 오후 7시 멘토와의 온라인 회의

2

매주 금요일 오후 7시 멘토 없이 대면 회의



# 예상 성과 소개



LLM과 RAG를 활용한  
코드 리뷰 자동화 어시스턴트  
SW 저작권 등록

**감사합니다**