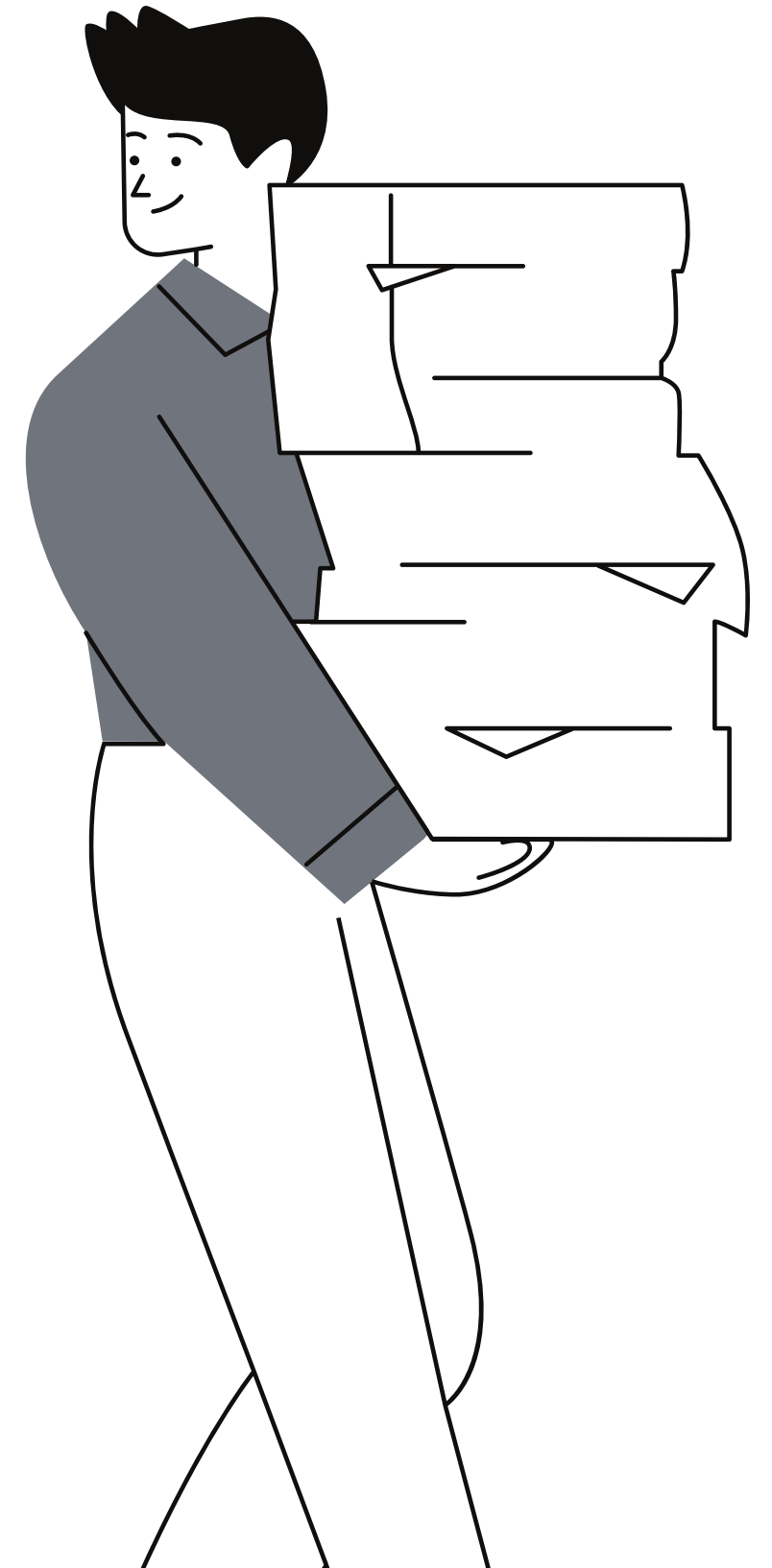


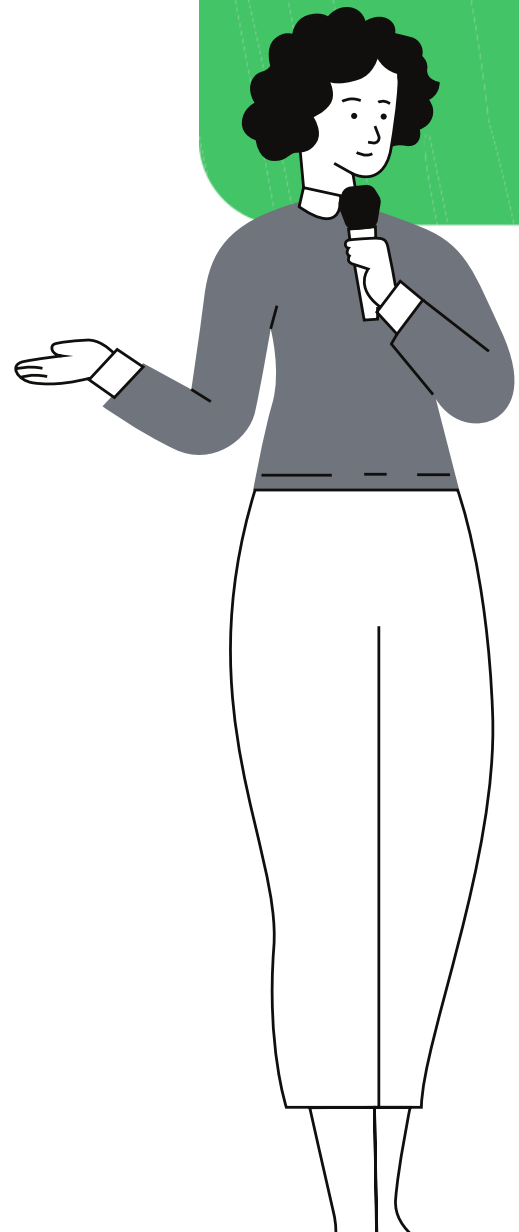


# 챗 Gpt를 활용한, 날씨 기반 의상 추천 서비스

Team 챗Trio



# 목차



1 프로젝트 소개

2 시연

3 활용 기술 설명

4 코드 설명

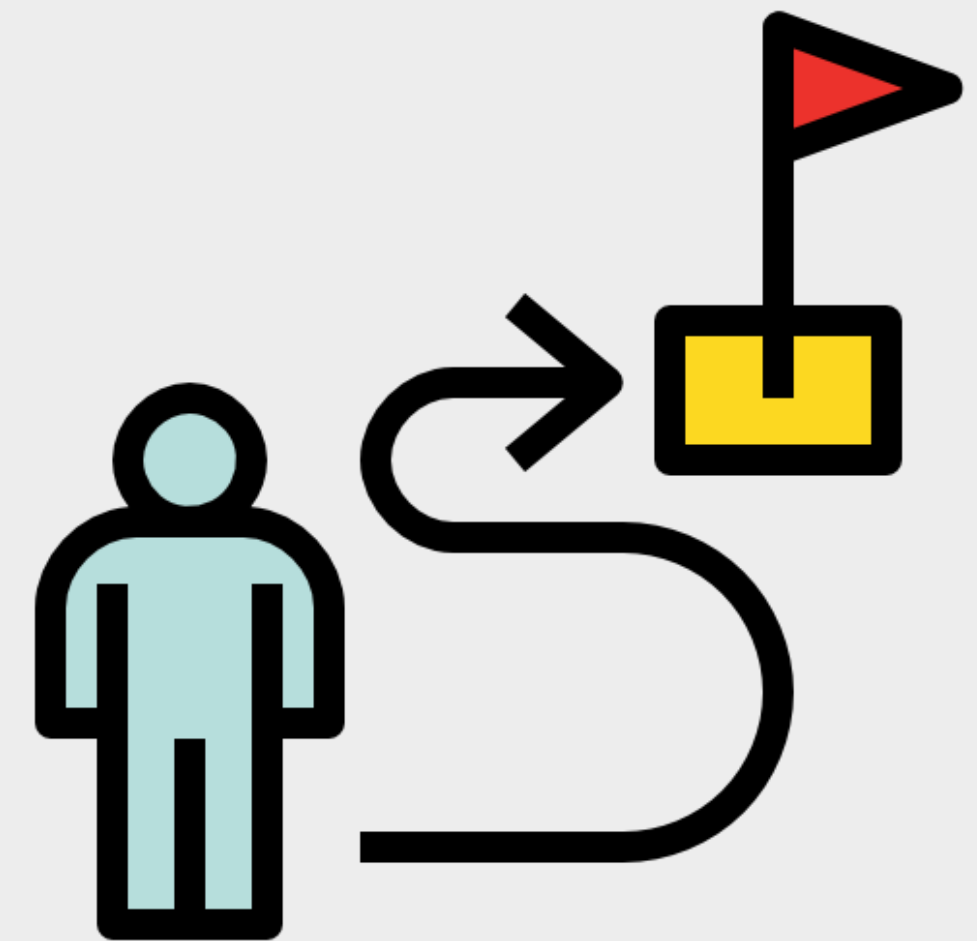
5 향후 개선 사항

6 팀 소개

7 Q&A

# 프로젝트 목적

1. python 언어를 활용한 ai 서비스 제공
2. 웹 기반 서비스 제공
3. 소비자의 편의 향상
4. AI기반 포트폴리오 제작



# 프로젝트 일정

(2023.05.02 ~ 2023.05.12)

5월 2일(1)

-팀 결성 및 첫 회의

5월 3일(2)

-프로젝트 방향 결정  
-프로젝트 일정 조정

5월 4일~7일(3)

-ChatGpt API 발급  
-파이썬 알고리즘 구현  
-Flask 서버 구현

5월 8일~10일(4)

-프론트엔드 구현  
-알고리즘 개선  
-기능 추가

5월 11일(5)

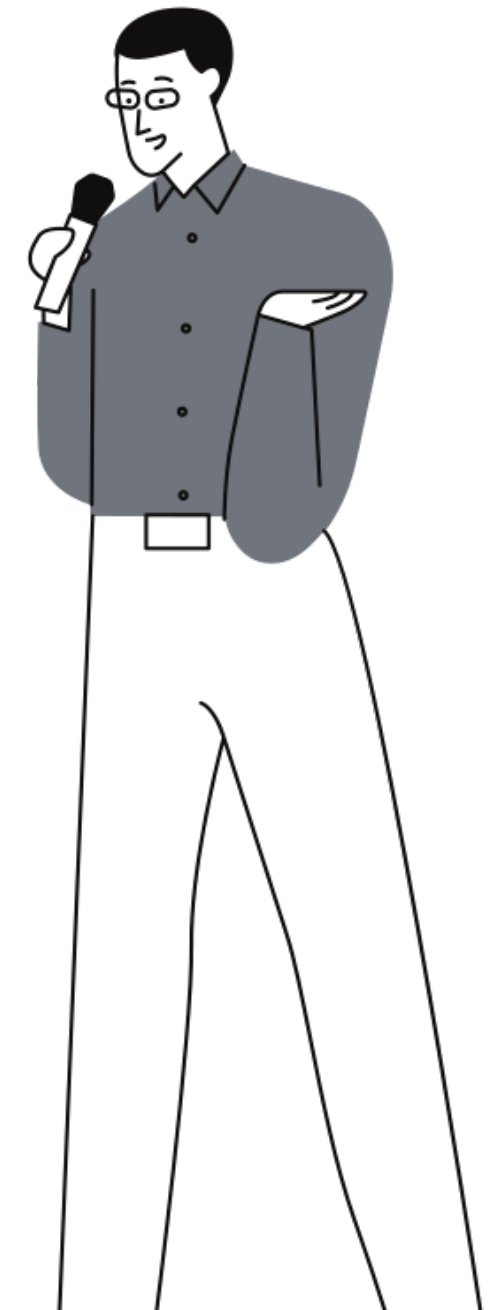
-프레젠테이션 준비  
-발표 준비  
-최종 정리

## 2023.05.12 발표



준비됐나요?

# 프로젝트 시연



# 활용 기술

lct\_mini\_project

Front-end



CSS



HTML



Flask

web development,  
one drop at a time

Back-end



Chat Gpt(open\_ai)



python<sup>TM</sup>

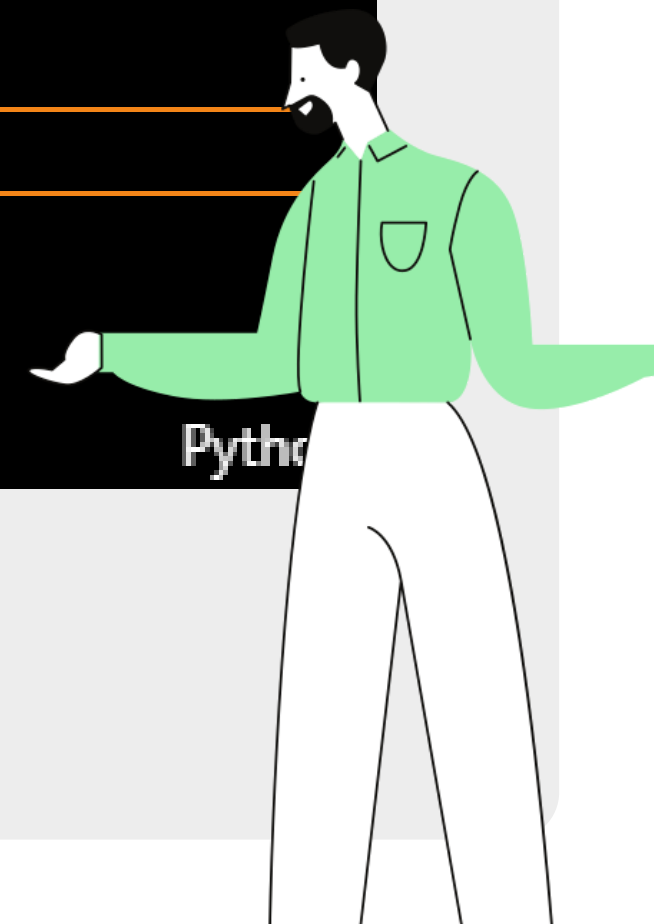


# 코드 설명(APP.PY)

lct\_mini\_project

## 기본 설정

```
### 사전 설정 ###  
import openai # openai 라이브러리  
from urllib.request import urlopen # 크롤링 라이브러리  
from bs4 import BeautifulSoup # 크롤링 라이브러리  
from flask import Flask, render_template, request # 플라스크 라이브러리  
  
msg_input = [] # 명령어 리스트  
msg_result = [] # 결과값 리스트  
aiToken = 3 # 사용회수
```

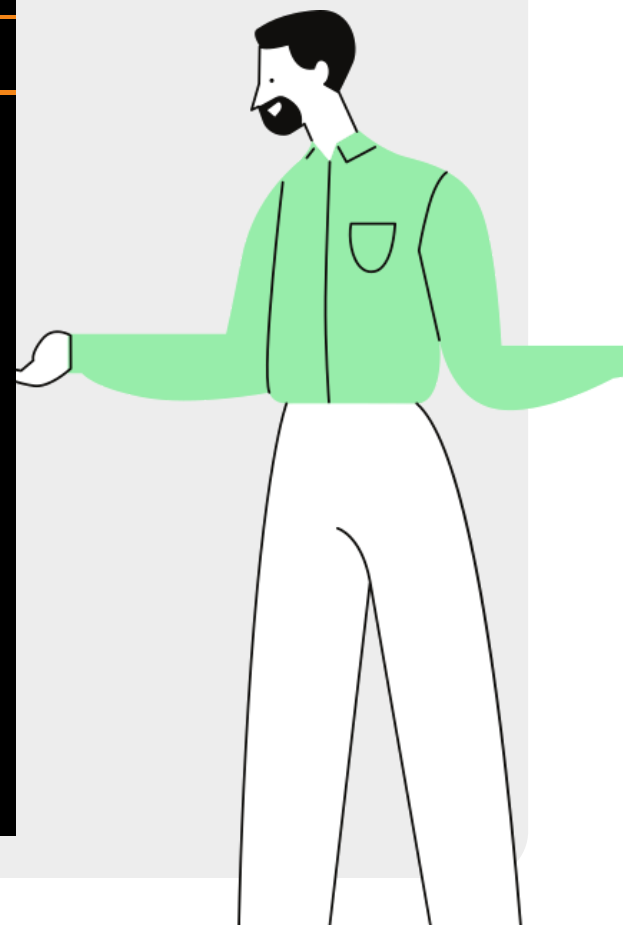


시간: 분



```
def openAi(msg):  
    key1 = "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$" #key값 input  
    openai.api_key = key1 # 키설정  
    # {"role": "user", "content": msg}) => 답변,  
    msg_input.append( {"role": "user", "content": msg})  
  
    # 답변 생성, "model값" 변경으로 다른 모델 사용가능  
    response = openai.ChatCompletion.create(  
        model="gpt-3.5-turbo",  
        messages=msg_input  
    )  
    # 답변, 답변에 따라 [1],[2]...까지 수 증가  
    answers=response.choices[0].message.content.split("\n\n")  
    print(answers)  
    return answers # 답변
```

## Python



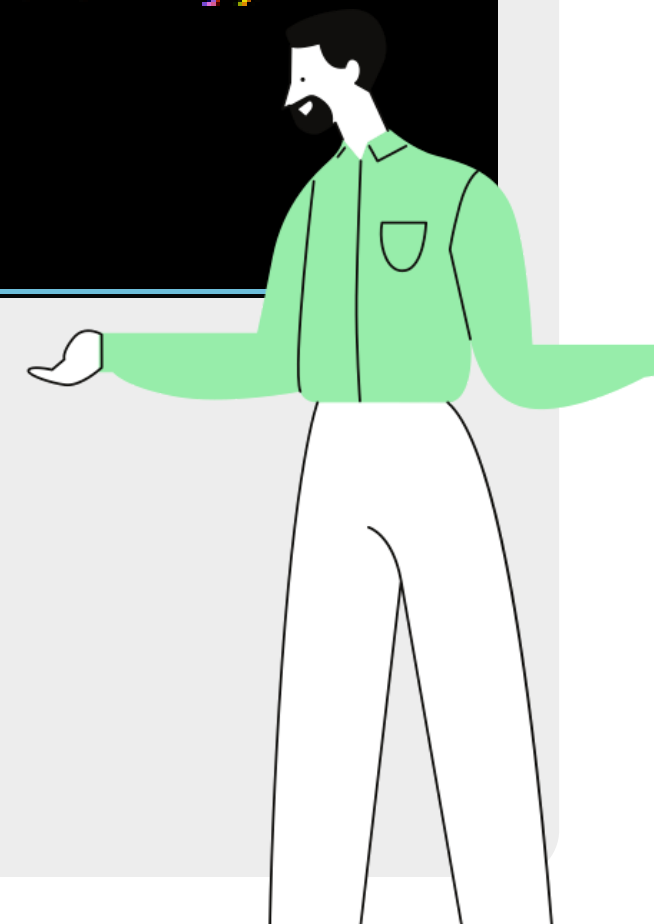
# 코드 설명(APP.PY)

lct\_mini\_project

## 함수 basicAssistant():CahtGpt 기본 명령어 설정

```
def basicAssistant():  
    msg_input.append(  
        {"role": "system", "content": """"너는 최고의 코디네이터고 나에게 옷을 추천해주는 사람이야"""})  
    msg_input.append(  
        {"role": "assistant", "content": """"메이드처럼 나에게 답해줘"""})
```

시간: 분



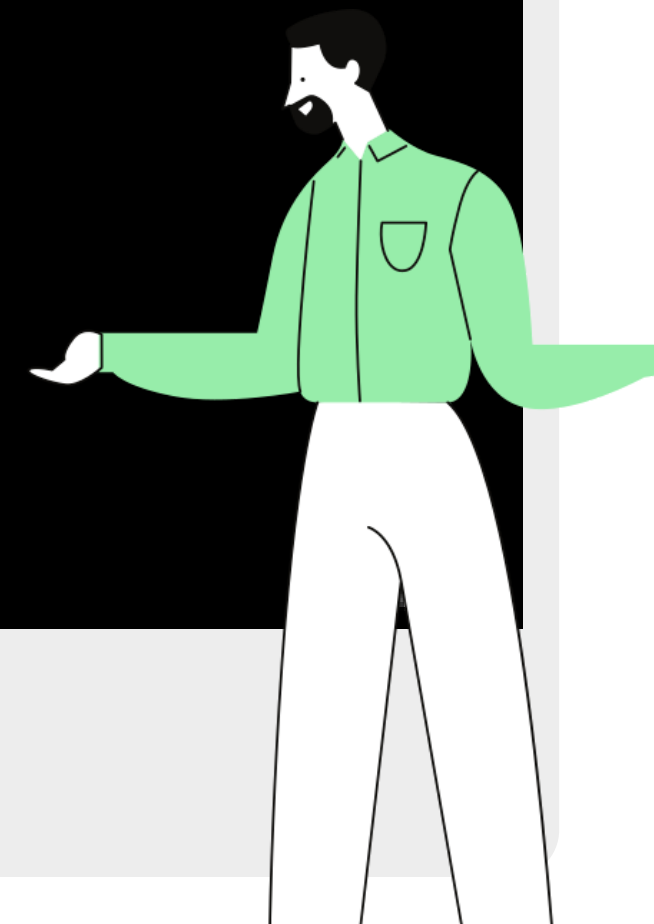
# 코드 설명(APP.PY)

lct\_mini\_project

## 함수 weatherAssistant():실시간 날씨 크롤링 및 명령어 입력

```
def weatherAssistant():  
    url = "https://weather.naver.com/" # 네이버 날씨에서 크롤링  
    page = urlopen(url)  
    soup = BeautifulSoup(page, 'lxml')  
    # 날씨  
    weath = soup.find("span", class_="weather").text  
    # 온도  
    temp = soup.find("strong", class_="current").text  
    now_temp = temp[1:-1]  
    weather = "오늘 날씨는 "+weath+", "+now_temp  
    # chat에 assistant 추가  
    msg_input.append({"role": "assistant", "content": weather})
```

시간: 분



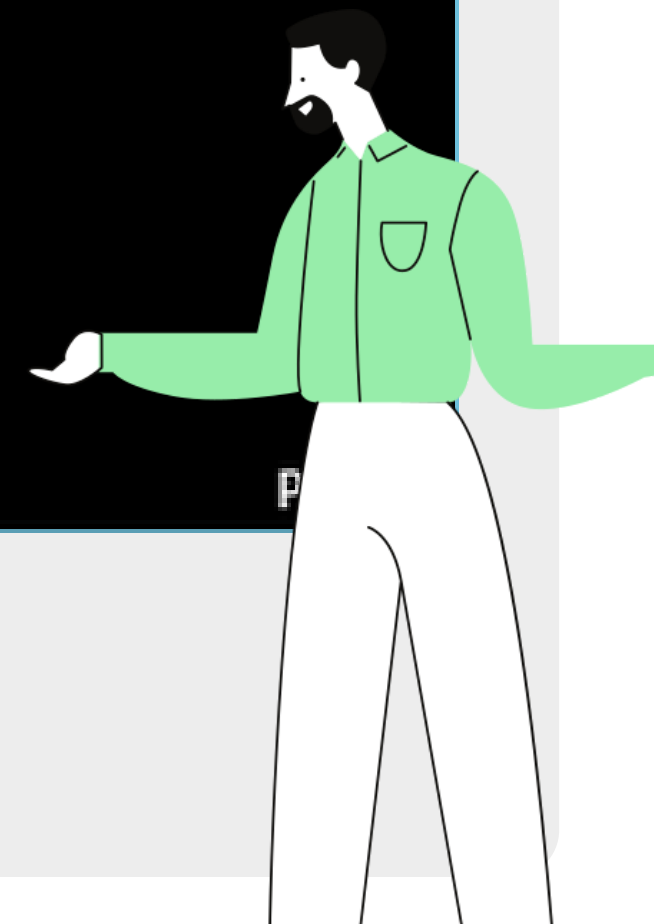
# 코드 설명(APP.PY)

lct\_mini\_project

## Falsk:python언어 html언어 연결

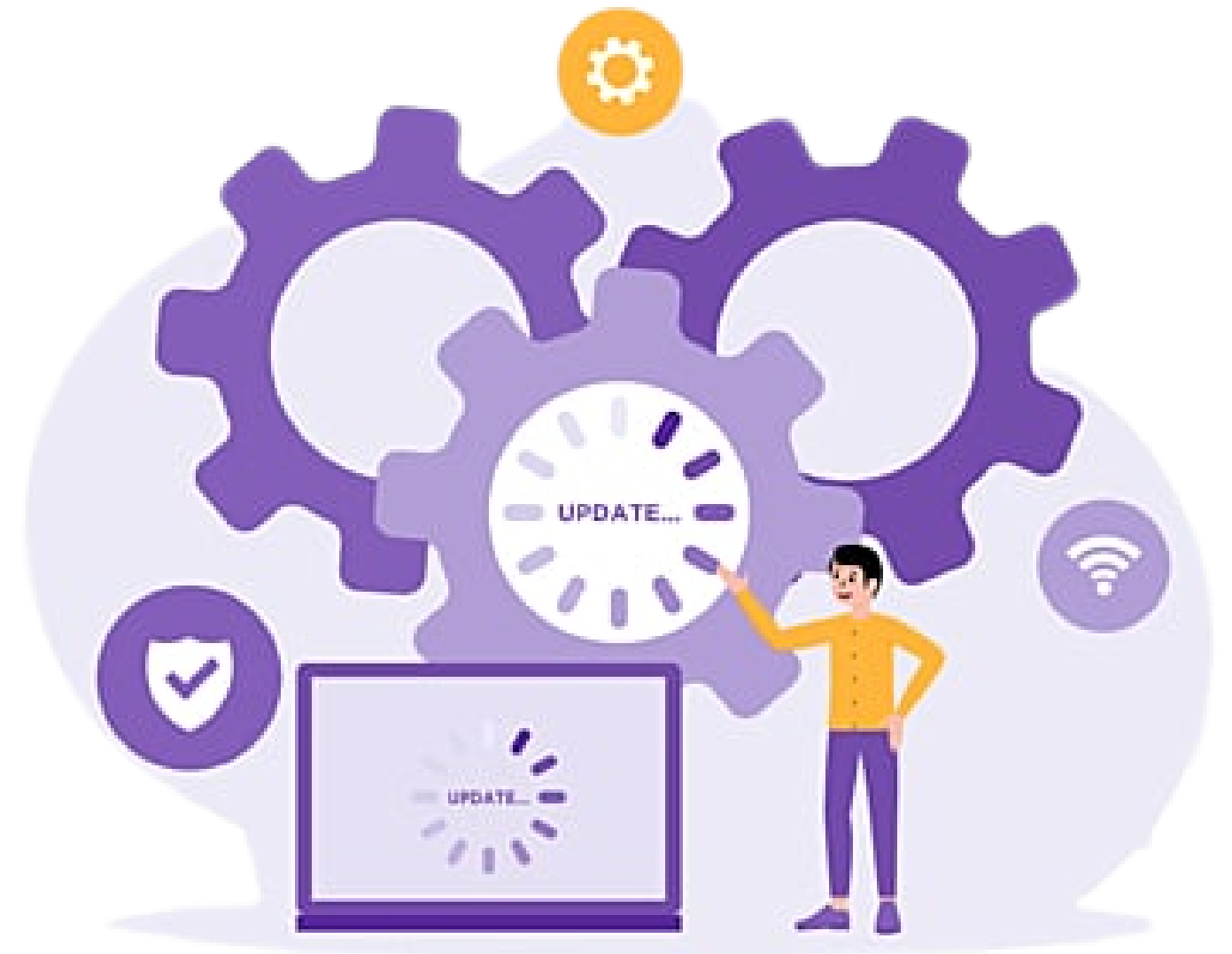
```
app = Flask(__name__) # app 변수에 플라스크 설정
@app.route('/') # 루트 설정
def home():
    global aiToken #변수 선언
    aiToken = 3 # Token 재설정
    msg_input.clear() # Chat 명령어 초기화
    basicAssistant() # 기본 assistant 추가
    weatherAssistant() # 날씨 assistant 추가
    return render_template('index.html')
```

시간: 분



# 추후 개선 사항

- 1 서버간의 시간 지연 => 로딩창 구현
- 2 기본적인 웹 디자인 => 디자인 개선
- 3 공유기능(카카오톡, 페이스북)
- 4 DB 저장
- 4 웹 도메인 설정



# 6. 팀 소개

lct\_mini\_project



팀장 : 최재광(Jay)

- 프로젝트 총괄
- 백엔드 개발
- 프론트엔드 개발



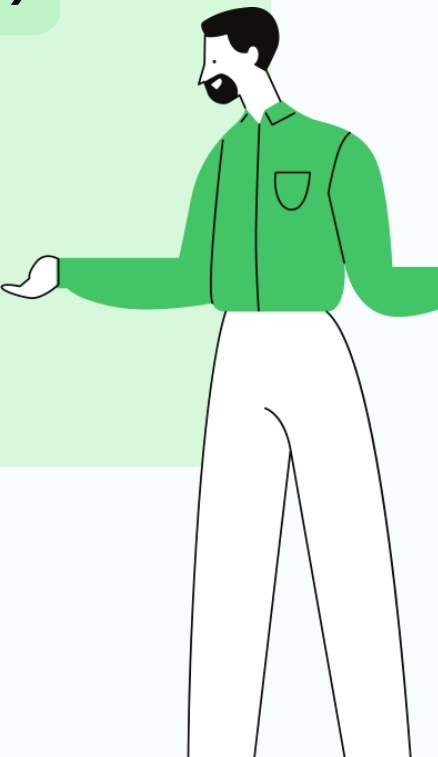
팀원 : 송솔(룰)

- 프로젝트 총괄
- 백엔드 개발
- 프론트엔드 개발



팀원 : 노슬기(Grace)

- 프로젝트 총괄
- 백엔드 개발
- 프론트엔드 개발





# Q&A

참여해 주셔서  
감사합니다.  
좋은 하루 보내세요.

