from openai import OpenAI

import requests

openai\_client = OpenAI()

def speech\_to\_text(audio\_binary):

    # Set up Watson Speech-to-Text HTTP Api url

    base\_url = "https://sn-watson-stt.labs.skills.network"

    api\_url = base\_url+'/speech-to-text/api/v1/recognize'

    # Set up parameters for our HTTP reqeust

    params = {

        'model': 'en-US\_Multimedia',

    }

    # Set up the body of our HTTP request

    body = audio\_binary

    # Send a HTTP Post request

    response = requests.post(api\_url, params=params, data=audio\_binary).json()

    # Parse the response to get our transcribed text

    text = 'null'

    while bool(response.get('results')):

        print('speech to text response:', response)

        text = response.get('results').pop().get('alternatives').pop().get('transcript')

        print('recognised text: ', text)

        return text

def text\_to\_speech(text, voice=""):

    # Set up Watson Text-to-Speech HTTP Api url

    base\_url = "https://sn-watson-tts.labs.skills.network"

    api\_url = base\_url + '/text-to-speech/api/v1/synthesize?output=output\_text.wav'

    # Adding voice parameter in api\_url if the user has selected a preferred voice

    if voice != "" and voice != "default":

        api\_url += "&voice=" + voice

    # Set the headers for our HTTP request

    headers = {

        'Accept': 'audio/wav',

        'Content-Type': 'application/json',

    }

    # Set the body of our HTTP request

    json\_data = {

        'text': text,

    }

    # Send a HTTP Post request to Watson Text-to-Speech Service

    response = requests.post(api\_url, headers=headers, json=json\_data)

    print('text to speech response:', response)

    return response.content

def openai\_process\_message(user\_message):

    # Set the prompt for OpenAI Api

    prompt = "Act like a personal assistant. You can respond to questions, translate sentences, summarize news, and give recommendations."

    # Call the OpenAI Api to process our prompt

    openai\_response = openai\_client.chat.completions.create(

        model="gpt-3.5-turbo",

        messages=[

            {"role": "system", "content": prompt},

            {"role": "user", "content": user\_message}

        ],

        max\_tokens=4000

    )

    print("openai response:", openai\_response)

    # Parse the response to get the response message for our prompt

    response\_text = openai\_response.choices[0].message.content

    return response\_text