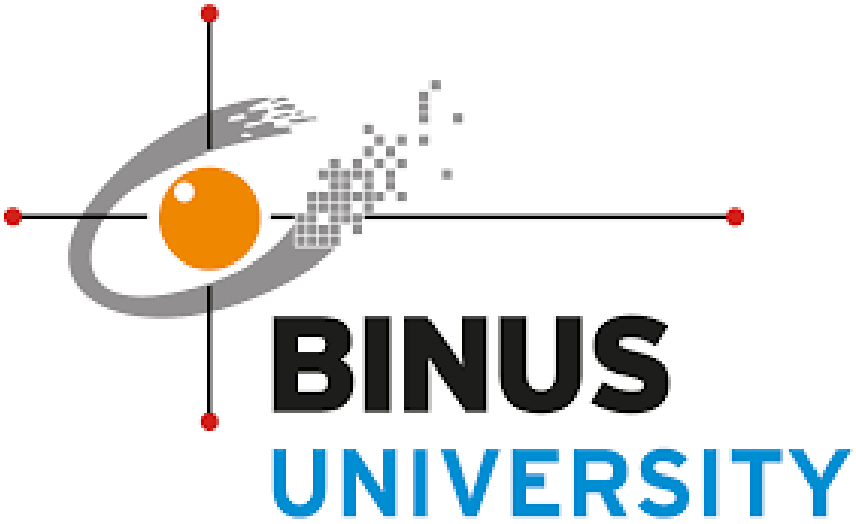
AlGORITHM AND PROGRAMMING

FINAL PROJECT REPORT

VOICE ASSISTANT



By

Ivandito Rakaputra - 2602119303

Lecturer:

Jude Joseph Lamug Martinez, MCS

School of Computing and Creative Arts

BINUS International University

I. Introduction

For this final project, the students in Computer Science are asked to make a comprehensive application that solves a problem, beyond what was taught in the class over the semester. Students are also expected to solve the problem as long as they are given enough time to finish their tasks.

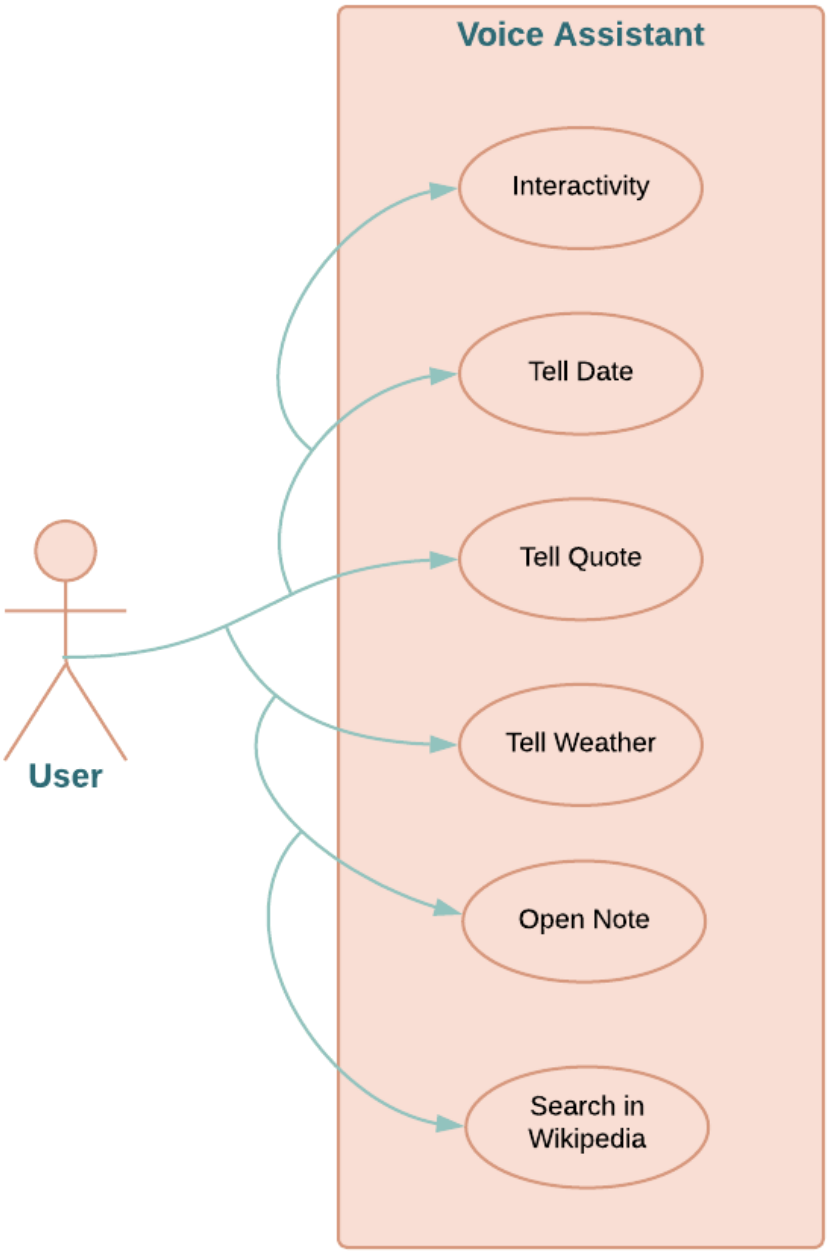
During last Christmas holiday, I was thinking about what I will make for this final project. Finally, I chose to make a simple voice assistant for the final project. When I thought about making this voice assistant, I had 0 experience to make it, from what libraries were used, then the contents of this voice assistant, and others. Therefore, during the time I was working on this project, I studied the libraries that I used and their implementation in the coding.

First of all, I plan to make this voice assistant just to be a robot that can answer some specific simple questions based on the maker putting the questions in the code. So, the robot will be triggered to answer the question when the question is in accordance with what is written in the code. But besides that, the robot can also answer questions based on existing data. It means that the robot can answer questions according to the libraries I use (the use of libraries will be explained later). This project started from my thought that I want someone who can help me to entertain and make my work easier. But that “someone” is not human, but a robot that can work tirelessly.

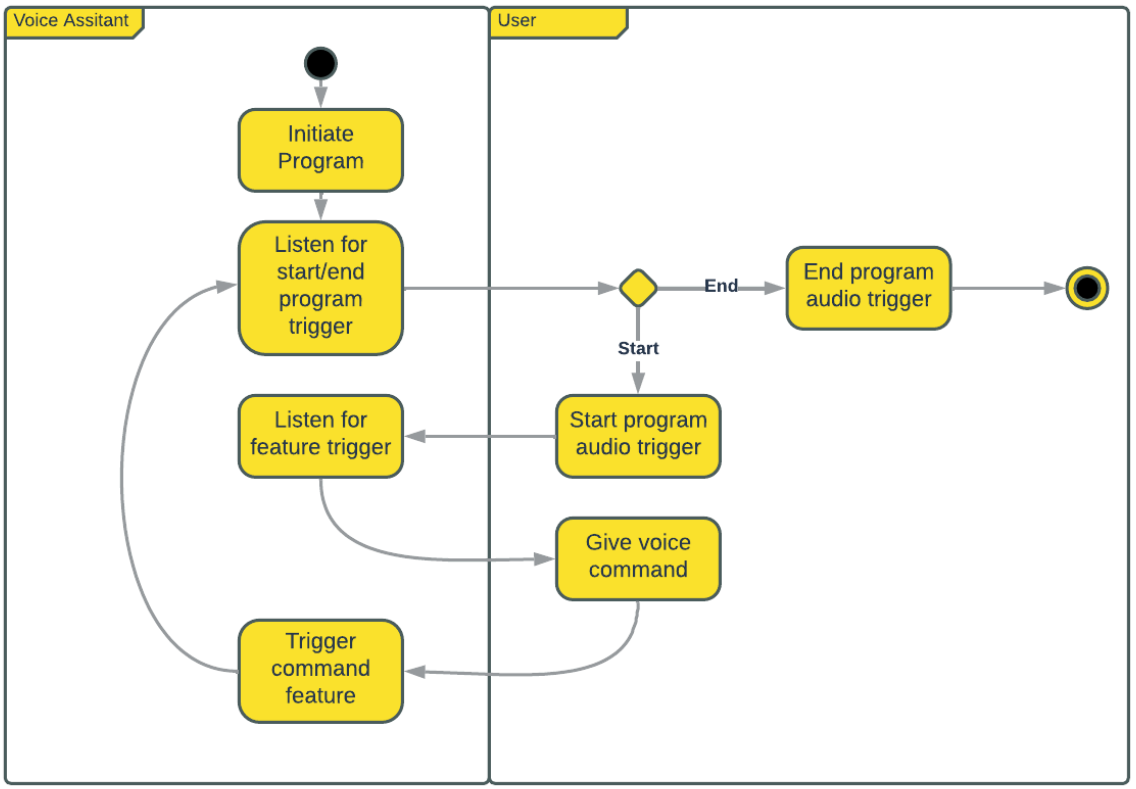
So, the goal of this project for me is to provide an illustration that in the future voice assistants will be very much needed to help humans make their work easier, starting with this small project, where this voice assistant can interact with humans.

II. Solution Design (UML Diagram)

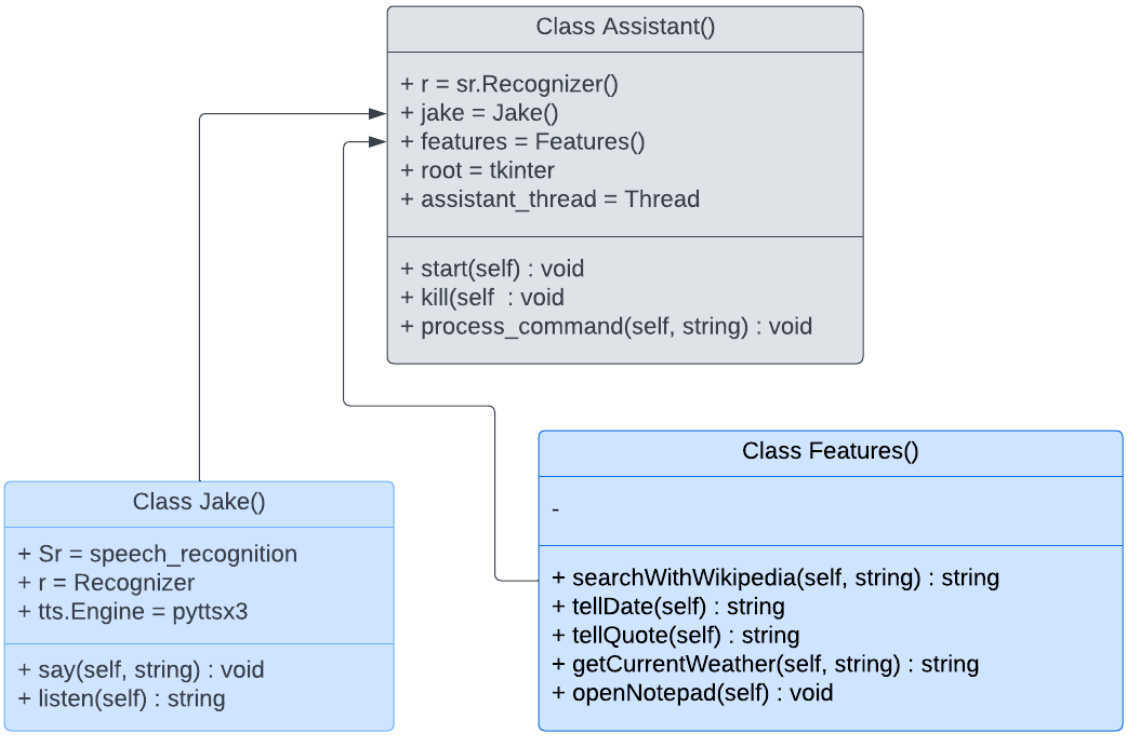
1. Use Case Diagram



2. Activity Diagram



3. Class Diagram



III. Algorithm and Data Structure.

1. Modules

So, to help work on this final project, I use these modules as follows:

I. speech_recognition

I use these modules to machine's to be able to listen to spoken words and identify them. I used this speech_recognition module in Python to convert the spoken words into text, make a query or give a reply.

II. pyttsx3

pyttsx3 is a text-to-speech conversion module in Python. I use this tool which converts the entered text into speech .

III. wikipedia

I use these modules to make it easy to access and parse data from Wikipedia.

IV. datetime

I use these modules to manipulate date and time object data.

V. quote

I use these modules to generate quotes.

VI. tkinter

Tkinter is the standard GUI module for Python. I use this module because tkinter provides a fast and easy way to create GUI applications.

VII. threading

I use these modules to run multiple threads (tasks, function calls) at the same time.

VIII. os

I use these modules because they provide the facility to establish the interaction between the user and the operating system.

IX. requests

I use these modules for making HTTP requests in Python.

2. Algorithm

The algorithm that is used to make this program is Speech Recognition Algorithm. A speech recognition algorithm or voice recognition algorithm is used in speech recognition technology to convert voice to text. It works by hearing human sound in a normal environment. This normal environment is referred to as an analog environment. A computer can't work with analog data, so it needs digital data. This is why the first piece of equipment needed is an analog to digital converter, which is a microphone. These Speech recognition systems have several advantages, which is efficiency where we can generate everything faster (documents,etc), and handsfree communication, where we can ask a robot to do the work instead of people.

3. Essential Algorithms

```
1   import speech_recognition as sr
2   from tkinter import *
3   from tkinter import ttk
4   from jake import Jake
5   from features import Features
6   import threading
7   import os
```

```
import speech_recognition as sr
import pyttsx3
```

```
import os
import wikipedia
import datetime
from quote import quote
import requests
```

```python
class Assistant():

    """
    Main Class of this Project
    """

    def __init__(self):
        self.r = sr.Recognizer()
        self.jake = Jake()
        self.features = Features()

        #create a tkinter root
        self.root = Tk()

        #create window title
        self.root.title("Jake Voice Assistant")
        #create the UI of jake
        self.label = ttk.Label(text="😈", font=("Arial", 120, "bold"))
        self.label.pack()

        #create and start a separate thread for self.start method
        self.assistant_thread = threading.Thread(target=self.start)
        self.assistant_thread.start()

        #create a window listener for exit event
        self.root.protocol("WM_DELETE_WINDOW", self.kill)

        #to start tkinter
        self.root.mainloop()
```

```python
                #if the commands didn't follow
                except:
                    print("Error")

        except sr.RequestError as e:
            print("Could not request results from Google Speech Recognition service; {0}".format(e))
```

```python
def start(self):

    """
    Starts the Voice Assistant
    """

    #create infinite loop for voice assistant to always listen.
    while True:
        print("Listening...")
        recognized_audio_str = self.jake.listen()

    #Introduction of Jake  , the smart voice assistant.

        try:
            #trigger exit program
            if "goodbye" in recognized_audio_str:
                self.jake.say("Thank you, goodbye!")
                self.kill()

            #trigger jake's features
            if "hey jake" in recognized_audio_str:
                self.label.config(foreground="red")

                #Respond of jake
                print("What can I do for you?")
                self.jake.say("What can I do for you?")
                recognized_audio_str = self.jake.listen()
                try:

                    #process command from recognized audio
                    self.process_command(recognized_audio_str)
```

```python
#to stop program
def kill(self):
    print("Goodbye :)")
    self.root.destroy()
    os._exit(1)
```

```
#list of conditions to trigger features
def process_command(self, command_str):
    if "tell me about yourself" in command_str:
        self.jake.say(
            "I am jake, a voice assistant created by my creator, Ivan. I was made to assist him doing his "
            "final project.")
        print(
            "I am jake, a voice assistant created by my creator, Ivan. I was made to assist him doing his "
            "final project.")
        return
```

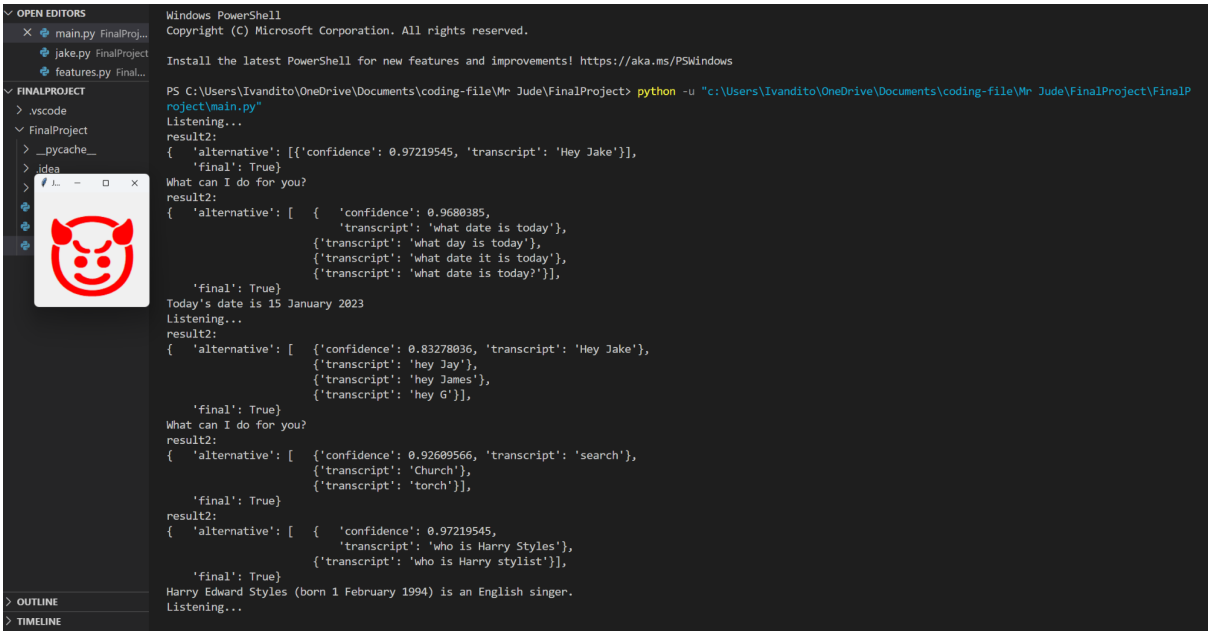
```
class Jake ():

    def __init__(self):
        self.ttsEngine = pyttsx3.init()
        self.r = sr.Recognizer()
        self.sr = sr

    def say(self, strToSay):
        """
        The respond of Jake
        """
        #use text to speech engine to output audio
        self.ttsEngine.say(strToSay)
        self.ttsEngine.runAndWait()

    def listen(self):
        """
        Listen for audio from user microphone
        """

        # Use microphone as audio source

        try:
            with sr.Microphone() as source:
                audio = self.r.listen(source)

                # use google's service to process audio input

                recognized_audio_str = self.r.recognize_google(audio)
                recognized_audio_str = str.lower(recognized_audio_str)
                return recognized_audio_str
```

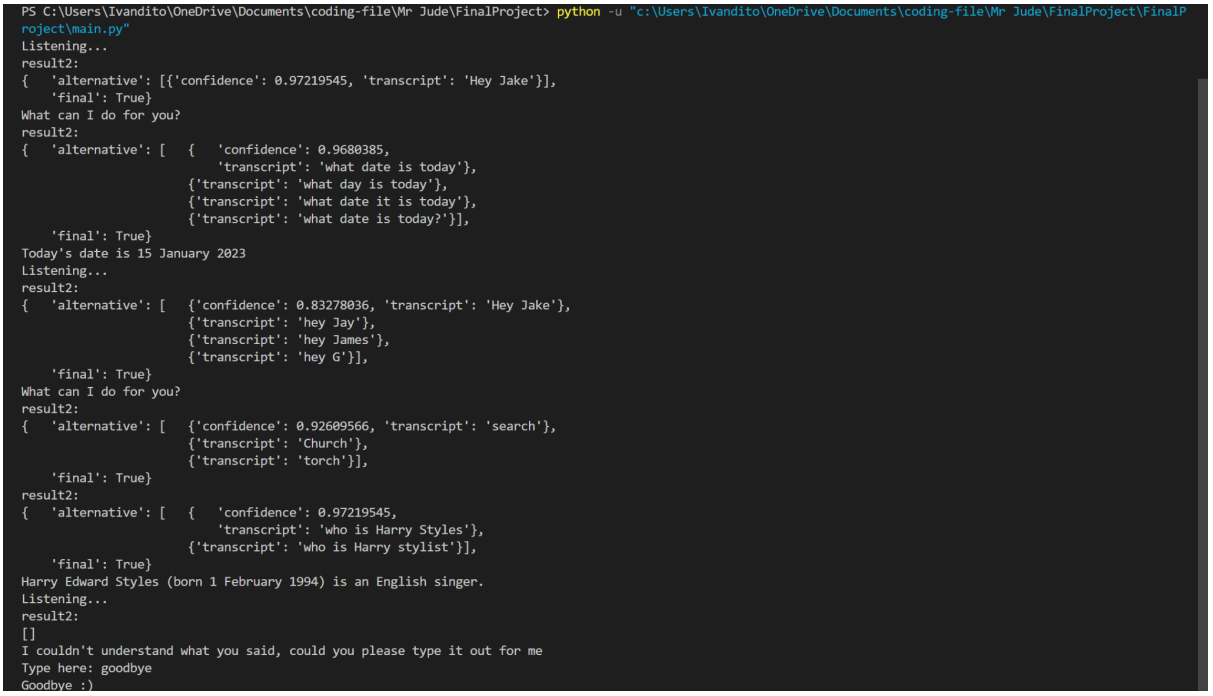
```
    except:
    #if the audio couldn't understand the voice, the program ask to type it.
        message = "I couldn't understand what you said, could you please type it out for me"

        print(message)
        self.say(message)

        #ask user for input
        command_str = input("Type here: ")
        return command_str
```

```
    #Special Features inside Jake (using external libraries)


    if "what date is today" in command_str:
        today = self.features.tellDate()
        self.jake.say(today)
        print(today)
        return

    if "search" in command_str:
        self.jake.say("What do you want to know?")
        recognized_audio_str = self.jake.listen()

        search_result = self.features.searchWithWikipedia(recognized_audio_str)
        self.jake.say(search_result)
        print(search_result)
        return
```

```
class Features():
    """
        This class contains all the features that Jake
        can do.
    """

    #search wikipedia for a topic
    def searchWithWikipedia(self, topic_to_search):
        searchResult = wikipedia.summary(topic_to_search, sentences=1)
        return searchResult


    def tellDate(self):
        #get current time
        now = datetime.datetime.now()

        #convert time to date, month, and year.
        date = now.strftime("%d")
        month = now.strftime("%B")
        year = now.strftime("%Y")

        #format date result
        date_result = "Today's date is " + date + " " + month + " " + year
        return date_result
```

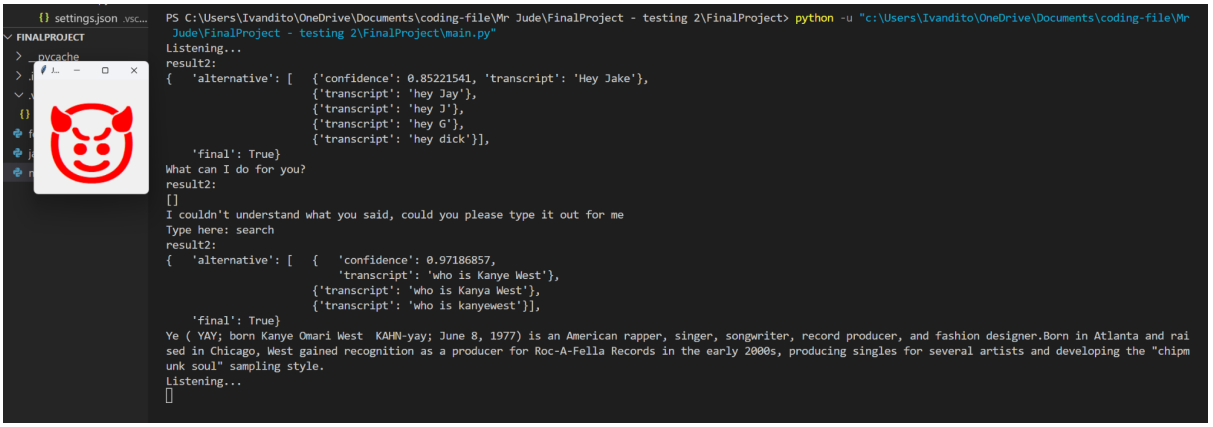
4. Screenshots of the application



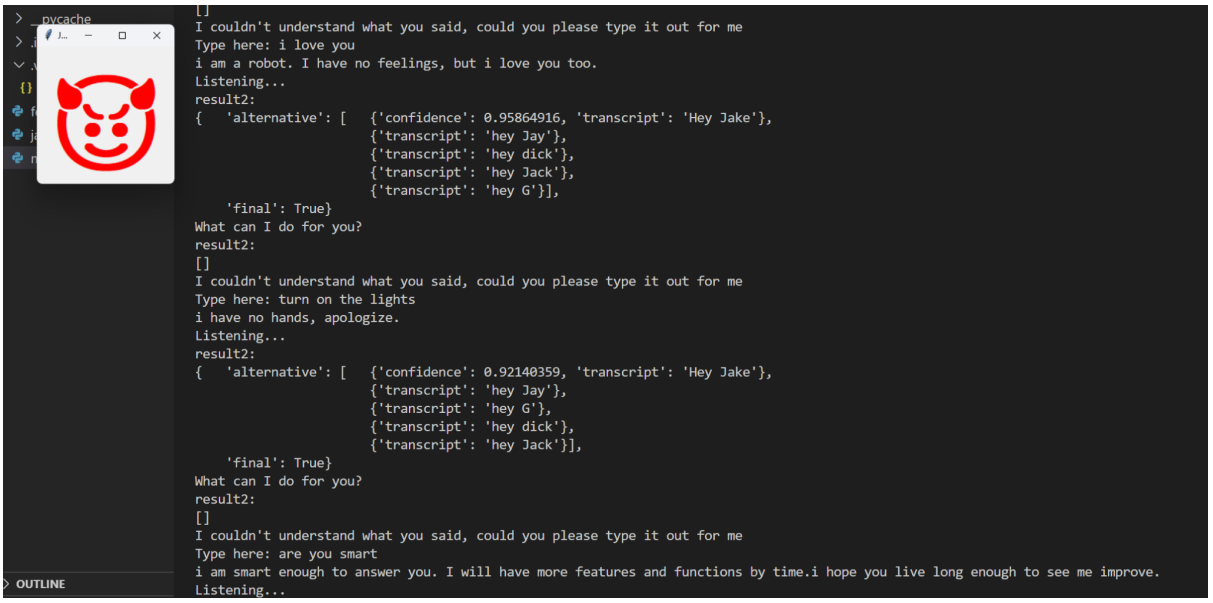


5. Screenshots of the error

The first error: when the program is stuck on listening, the user has to exit the program and rerun it.



The second error: when the program couldn't hear the audio from the user, the program asked to type the questions.



IV. Conclusion

1. Lesson Learned

I learned a lot from making this project. I ran into a lot of errors in the code but in the end, I was able to troubleshoot and fix them. I learned a lot of modules that I had never heard of before such as pyttsx3, tkinter, and others. Basically, making this voice assistant looks easy, but in fact it's not really. The error that is passed is not only from the code, but also other factors such as the environment sound which is louder than the user's voice. On the big occasion of this final project, I am very proud of what I have created. In the future, I will understand more about opening other modules and I will develop it into a voice assistant that can help us in our daily lives.

2. Video

https://youtu.be/fj1B-wwbp9c

3. Github Link

https://github.com/IvanditoRakaputra/Final-Project-Algopro

4. References

https://itchronicles.com/artificial-intelligence/speech-recognition-algorithms/

https://youtu.be/F62wb_jfUUw