AI BASED VIRTUAL VOICE ASSISTANT

A MINI PROJECT REPORT

Submitted by

JOITA GHOSH [RA2111003010664]
DHRUV SAINI[RA2111003010672]
KHUSHI MISHRA[RA2111003010673]

Under the guidance of

Dr. G Ramya

Assistant Professor, Department of Computer Science and Engineering

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2024

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report **"AI BASED VIRTUAL VOICE ASSISTANT"** is the bona fide work of **"JOITA GHOSH (RA2111003010664), DHRUV SAINI (RA2111003010672), KHUSHI MISHRA (RA2111003010673)"** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

(Signature)

Dr. G. Ramya
Assistant Professor
Dept of Computing Technologies
SRM Institute of Science and Technology

(Head Of Department)

Dr. M. Pushpalatha
Head and Professor
Dept of Computing Technologies
SRM Institute of Science and Technology

ABSTRACT

As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of the user. Speech recognition is the process of converting speech into text.

In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command.

In the current scenario, advancement in technologies is such that they can perform any task with the same effectiveness or can speak more effectively than us. By making this project, I realised that the concept of AI in every field is decreasing human effort and saving time.

Functionalities of this project include:

1. It can play music.
2. It can do Wikipedia searches for you.
3. It can open websites like Google, YouTube, etc., in a web browser.
4. It can give a weather forecast.
5. It can give desktop reminders of your choice.
6. It can have some basic conversation.

TABLE OF CONTENTS

INTRODUCTION

Artificial Intelligence, when used with machines, shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from humans. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python.

The instructions for the assistant can be handled as per the requirement of the user. In Python there is an API called Speech Recognition which allows us to convert speech into text. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command.

In the current scenario, advancement in technologies is such that they can perform any task with the same effectiveness or can speak more effectively than us. By making this project, we realised that the concept of AI in every field is decreasing human effort and saving time.

As the voice assistant is using Artificial Intelligence hence the results that it is providing are highly accurate and efficient. The libraries and packages used to make this assistant focuses on the time complexities and reduces time.

LITERATURE SURVEY

Author	Title	Dataset	Methods	Remarks
Satya Prakash Yadav Amit Gupta Mahaveer Singh Naruka Sansar Singh Chauhan	Voice-Based Virtual-Controlled Intelligent Personal Assistants	No specific dataset mentioned.	- NLP utilisation - Software integration - IoT architecture incorporation - Cloud-based systems development - Voice recognition and synthesis	Improved customer contacts - Privacy concerns - Evolution of voice recognition - Future prospects - Continued research into automation
Mr. Akash S, Mr. Neeraj Jayaram, Dr. Jesudoss A.	Desktop-based Smart Voice Assistant using Python Language Integrated with Arduino.	No specific dataset mentioned.	Python-Arduino integration, speech recognition, API calls for news/weather, system calls for desktop access, content	Challenges like imperfect voice recognition persist, yet Python's versatility enables seamless

			extraction, IoT via serial modules.	hardware-AI integration, enhancing functionality.
Ridwan Sanjaya , Christin Wibhowo	Arduino, Raspberry Pi, and Smartphone Usage Comparison for Voice-based Virtual Assistant	No specific dataset mentioned.	Compared voice features on Arduino, Raspberry Pi, and smartphones for borderline personality disorder assistance	Raspberry Pi and smartphones recommended; smartphones are cost-effective and user-friendly.
Dr. A. Jagan, Domakonda Nandini, Dhulipalla Naga Parvathi, Mohammad Ghouse	Personal Voice Assistant Using Computer Vision	No specific dataset mentioned.	Utilising Python algorithms for speech recognition, text-to-speech, and natural language processing to create a voice assistant operating without internet connections	The proposed model aims to enhance task management and scheduling efficiency, offering hands-free operation and significant potential for daily productivity improvement.

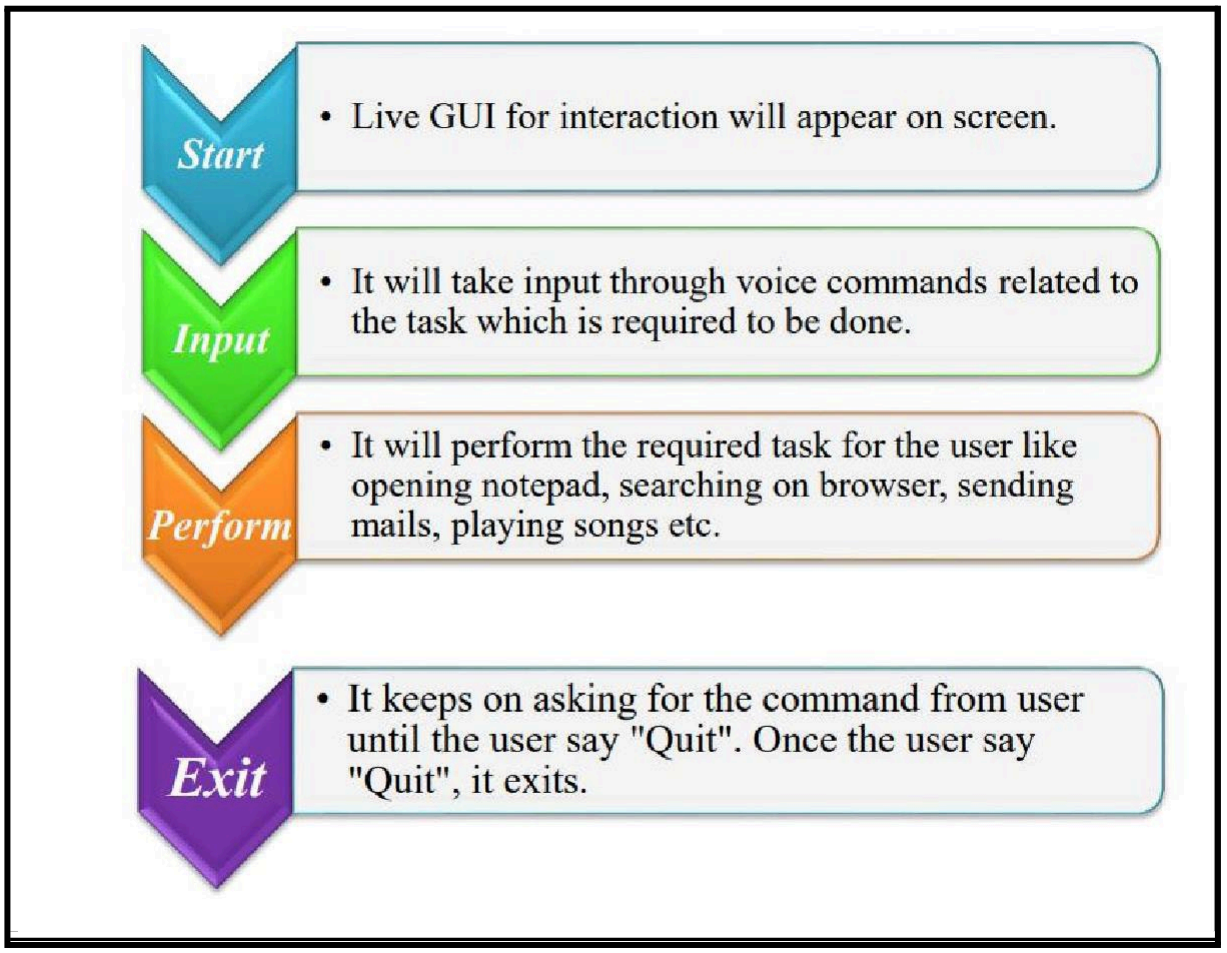
Pasha, Hanamkonda Susanna				
D Rajkumar Pillay, Saravanan A, Dr. Binda M B, Dr. Archana Raja, Dr. Manam Vamsi Krishna, Dr. Pankaj Saxena	AI-based Virtual Personal Assistant Design for Communication	VoiceAssistantDB	Develop AIVPA accepting voice/text inputs, utilising an algorithm for input processing, task execution, and data storage	AIVPA outperforms existing algorithms, demonstrating superior performance in input recognition, rejection, accuracy, precision, recall, and F1-Score, with higher accuracy and shorter processing time.

SYSTEM ARCHITECTURE AND DESIGN

- The IDE used in this project is PyCharm. All the python files were created in PyCharm and all the necessary packages were easily installable in this IDE.
- For this project following modules and libraries were used:
    1. **pyttsx3**: It is a python library which converts text to speech.
    2. **Speech Recognition**: It is a python module which converts speech to text
    3. **Datetime:** This library provides us the actual date and time.
    4. **Wikipedia:** It is a python module for searching anything on Wikipedia.
    5. **Smtplib**
    6. **pywhatkit**: It is a python library to send WhatsApp messages at a particular time with some additional features.
    7. **pyjokes**
    8. **pyPDF2**
    9. **pyautogui:** It is a Python library for graphical user interface.
    10. **pyQt**
    11. **Webbrowser:** It provides interface for displaying web- based documents to users

```
import pyttsx3
import speech_recognition as sr
import datetime
import pyaudio
import wikipedia
import webbrowser
import requests
from bs4 import BeautifulSoup
import pywhatkit
import pyautogui
```

DATA-FLOW DIAGRAM



- The data in this project is nothing but user input, whatever the user says, the assistant performs the task accordingly. The user input is nothing specific but the list of tasks which a user wants to get performed in human language i.e English

METHODOLOGY

1. Project Scope and Objectives:

- The assistant is designed to perform various tasks through voice commands, leveraging Python as the primary programming language and integrating with relevant APIs and libraries.
- The project aims to create a user-friendly and capable assistant that can enhance productivity and convenience by executing tasks such as playing music, conducting web searches, accessing weather forecasts, setting reminders, and engaging in basic conversations.

2. Research and Select Technologies:

- We conducted thorough research on over 10 papers based on AI voice assistant technology before settling on the best five as given in our literature survey.
- Python was chosen as the primary language and we used pyttsx3, Speech Recognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc.libraries.

3. Setting Up Development Environment:

- We installed the necessary Python packages and dependencies for speech recognition, text processing, web interaction, and task automation.
- Configured development environment with required tools like text editors, version control systems, and virtual environments for managing depende

4. Design System Architecture:

- Designed a modular architecture for the virtual voice assistant system, considering scalability, maintainability, and ease of integration for future enhancements.

5. Implement Speech Recognition:

- Utilising the SpeechRecognition API along with few other speech recognition libraries to capture audio input from the user and convert it into text.

6. Integrate Functionalities:

- Implementing modules to perform various functionalities such as playing music, searching Wikipedia, accessing weather forecasts, opening websites, and setting reminders.
- Integrate APIs for accessing external services like music streaming platforms, search engines, and weather forecast providers.

7. Test and Debug:

- Conducting comprehensive testing to ensure the correctness, reliability, and performance of the virtual voice assistant across different scenarios and user inputs.
- Debugging issues related to speech recognition accuracy, natural language understanding, and task execution.

8. Deploy and Evaluate:

- Deploying the virtual voice assistant system on appropriate platforms such as desktop computers.
- Gathering user feedback and evaluating the effectiveness, usability, and satisfaction of the virtual voice assistant in real-world scenarios.

.

9. Maintain and Update:

- Establishing procedures for maintaining the virtual voice assistant system, including monitoring performance, handling user inquiries, and addressing technical issues.
- Continuously update the system with new functionalities, improvements in speech recognition and compatibility with evolving technologies and APIs.

10. Document and Disseminate:

- Documenting the development process, system architecture, and technical details of the virtual voice assistant project for future reference and knowledge sharing.

CODING AND TESTING

```
import pyttsx3
import speech_recognition as sr
import datetime
import webbrowser
import wikipedia
import requests
from bs4 import BeautifulSoup
import pywhatkit
import pyautogui
import time

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


def wishMe():
    hour = datetime.datetime.now().hour
    if 0 <= hour < 12:
        speak("Good Morning!")
    elif 12 <= hour < 18:
        speak("Good Afternoon!")
    else:
        speak("Good Evening!")
    speak("I am David, how may I help you?")
```

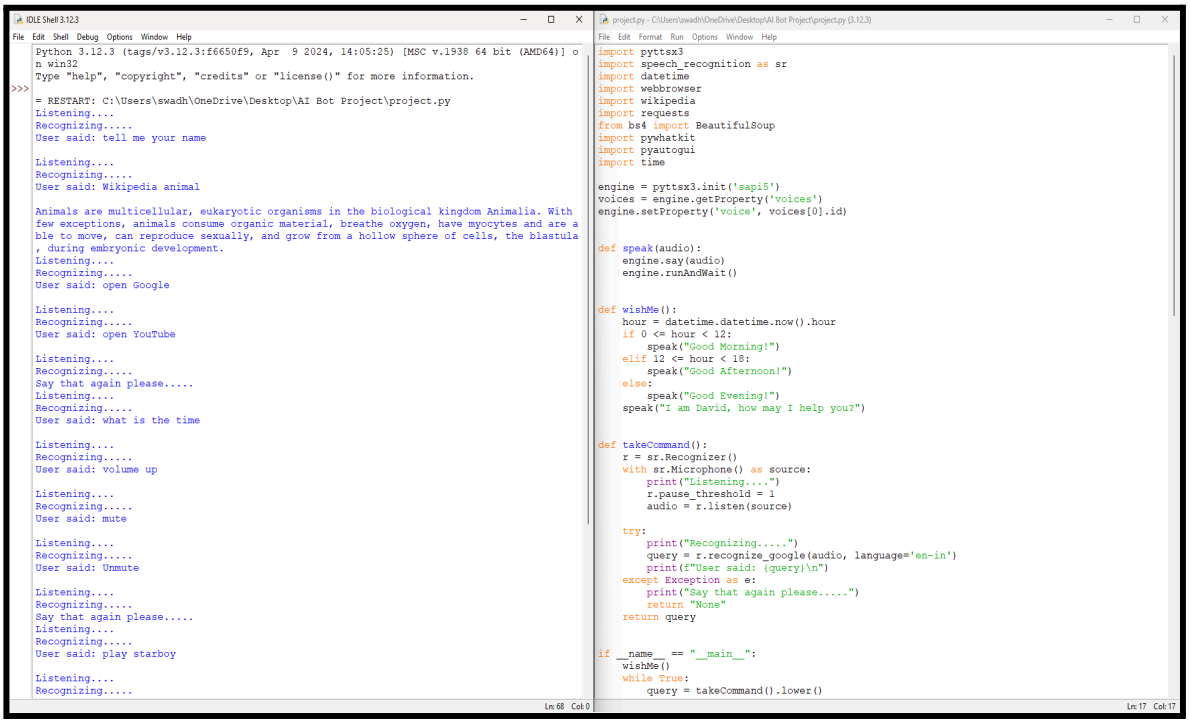
```
def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening....")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing.....")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")
    except Exception as e:
        print("Say that again please.....")
        return "None"
    return query


if __name__ == "__main__":
    wishMe()
    while True:
        query = takeCommand().lower()

        if 'wikipedia' in query:
            speak('Searching Wikipedia....')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'search' in query:
            query = query.replace("search", "")
            keywords = "+".join(query.split())

webbrowser.open(f"https://www.google.com/search?q={keywords}
```

```
")
            time.sleep(2)  # Delay after opening the browser

        elif any(word in query for word in ['name', 'about']):
            speak("Hi! My name is David. I am an AI-based Virtual Voice
Assistant capable of performing all your daily tasks!")

        elif 'open youtube' in query:
            webbrowser.open("https://www.youtube.com")

        elif 'open google' in query:
            webbrowser.open("https://www.google.com")

        elif 'open stackoverflow' in query:
            webbrowser.open("https://www.stackoverflow.com")

        elif 'time' in query:
            currentTime =
datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"The time is {currentTime}")

        elif 'date' in query:
            currentDate =
datetime.datetime.now().strftime("%D:%M:%Y")
            speak(f"The date is {currentDate}")

        elif any(word in query for word in ['up', 'increase']):
            pyautogui.press("volumeup")

        elif any(word in query for word in ['down', 'decrease']):
            pyautogui.press("volumedown")

        elif 'mute' in query:
            pyautogui.press("volumemute")

        elif 'play' in query:
```

```
    query = query.replace("play", "")
    keywords = "+".join(query.split())
    speak(f"Playing... {query}")
    pywhatkit.playonyt(keywords)

elif 'temperature' in query:
    city = "Chennai"
    search = f"temperature in {city}"
    url = f"https://www.google.com/search?q={search}"
    r = requests.get(url)
    soup = BeautifulSoup(r.text, "html.parser")
    try:
        temperature = soup.find("div", class_="BNeawe").text
        speak(f"Current {search} is {temperature}")
    except AttributeError:
        speak(f"Sorry, I couldn't find the temperature for {city}")

elif 'screenshot' in query:
    im = pyautogui.screenshot()
    speak("Taking screenshot")
    im.save("ss.jpg")

elif 'picture' in query:
    pyautogui.press("super")
    pyautogui.typewrite("camera")
    pyautogui.press("enter")
    time.sleep(2)  # Giving time to open camera app
    speak("Smile!")
    pyautogui.press("enter")

elif any(word in query for word in ['quit', 'exit']):
    speak("Have a nice day!")
    exit()
```

RESULTS

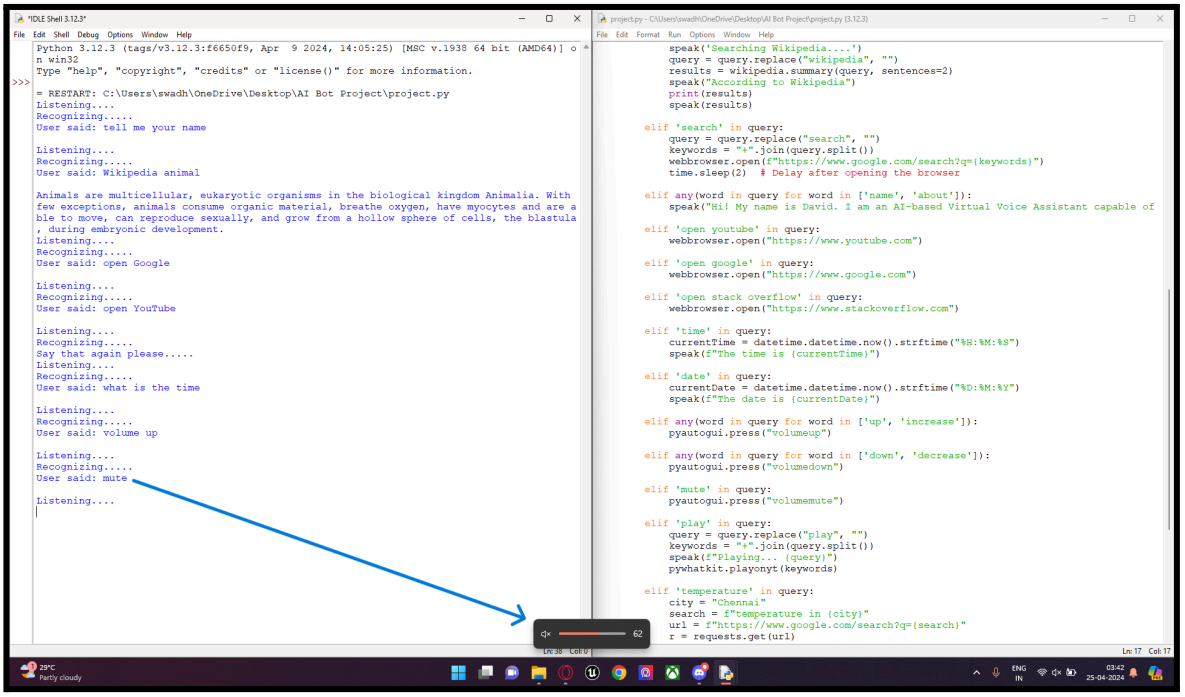
1. INTRODUCTION



```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] o
n win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\swadh\OneDrive\Desktop\AI Bot Project\project.py
Listening....
Recognizing.....
User said: tell me your name

Listening....
Recognizing.....
User said: Wikipedia animal

Animals are multicellular, eukaryotic organisms in the biological kingdom Animalia. With
few exceptions, animals consume organic material, breathe oxygen, have myocytes and are a
ble to move, can reproduce sexually, and grow from a hollow sphere of cells, the blastula
, during embryonic development.
Listening....
Recognizing.....
User said: open Google

Listening....
Recognizing.....
User said: open YouTube

Listening....
Recognizing.....
Say that again please.....
Listening....
Recognizing.....
User said: what is the time

Listening....
Recognizing.....
User said: volume up

Listening....
Recognizing.....
User said: mute

Listening....
Recognizing.....
User said: Unmute

Listening....
Recognizing.....
Say that again please.....
Listening....
Recognizing.....
User said: play starboy

Listening....
Recognizing.....
```

```
import pyttsx3
import speech_recognition as sr
import datetime
import webbrowser
import wikipedia
import requests
from bs4 import BeautifulSoup
import pywhatkit
import pyautogui
import time

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


def wishMe():
    hour = datetime.datetime.now().hour
    if 0 <= hour < 12:
        speak("Good Morning!")
    elif 12 <= hour < 18:
        speak("Good Afternoon!")
    else:
        speak("Good Evening!")
    speak("I am David, how may I help you?")


def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening....")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing.....")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")
    except Exception as e:
        print("Say that again please.....")
        return "None"
    return query


if __name__ == "__main__":
    wishMe()
    while True:
        query = takeCommand().lower()
```

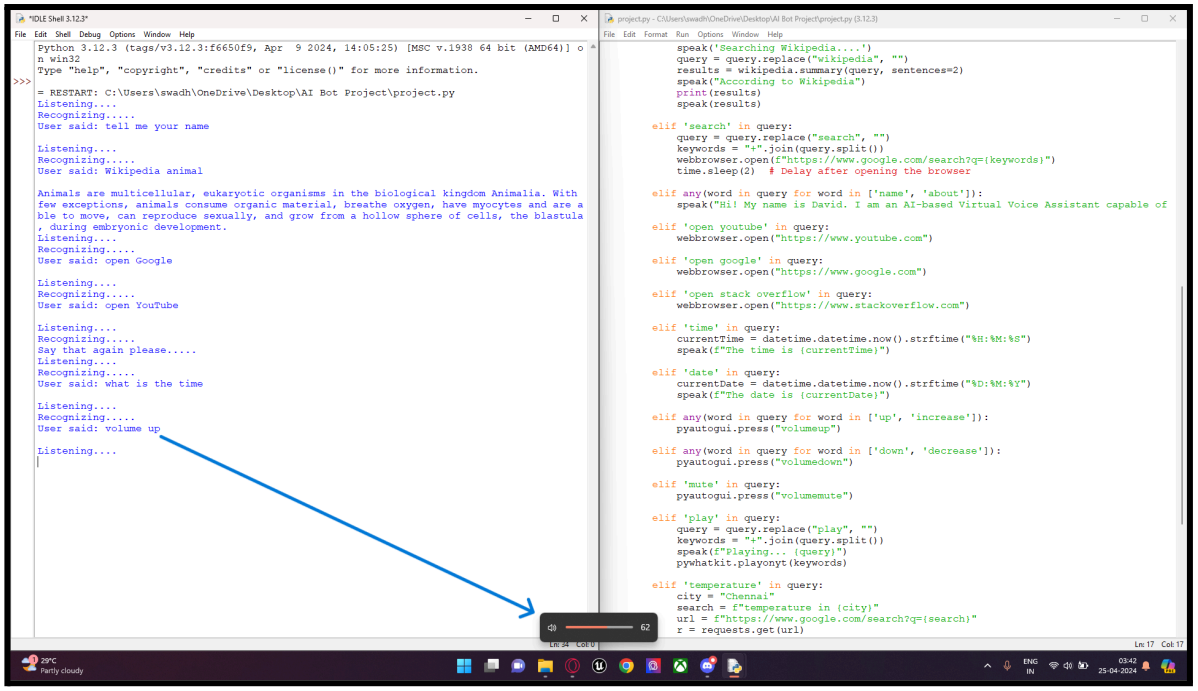
2. MUTE



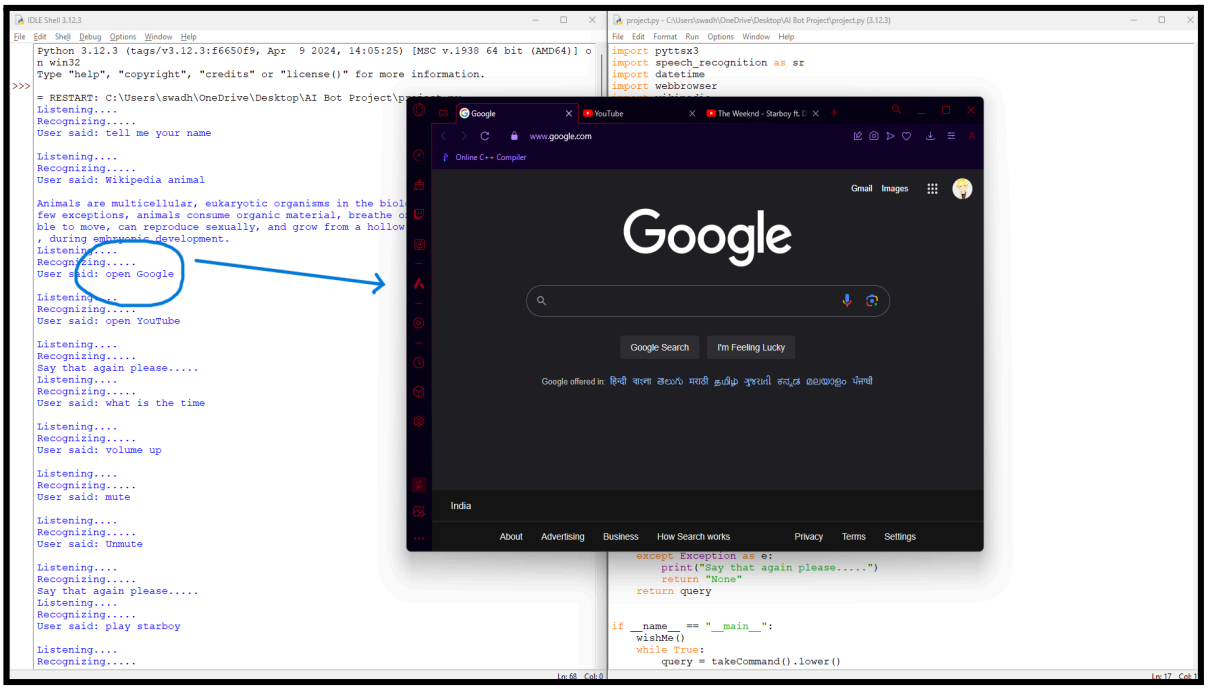
```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] o
n win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\swadh\OneDrive\Desktop\AI Bot Project\project.py
Listening....
Recognizing.....
User said: tell me your name

Listening....
Recognizing.....
User said: Wikipedia animal

Animals are multicellular, eukaryotic organisms in the biological kingdom Animalia. With
few exceptions, animals consume organic material, breathe oxygen, have myocytes and are a
ble to move, can reproduce sexually, and grow from a hollow sphere of cells, the blastula
, during embryonic development.
Listening....
Recognizing.....
User said: open Google

Listening....
Recognizing.....
User said: open YouTube

Listening....
Recognizing.....
Say that again please.....
Listening....
Recognizing.....
User said: what is the time

Listening....
Recognizing.....
User said: volume up

Listening....
Recognizing.....
User said: mute

Listening....
```

```
            speak('Searching Wikipedia....')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'search' in query:
            query = query.replace("search", "")
            keywords = "+".join(query.split())
            webbrowser.open(f"https://www.google.com/search?q={keywords}")
            time.sleep(2)  # Delay after opening the browser

        elif any(word in query for word in ['name', 'about']):
            speak("Hi! My name is David. I am an AI-based Virtual Voice Assistant capable of

        elif 'open youtube' in query:
            webbrowser.open("https://www.youtube.com")

        elif 'open google' in query:
            webbrowser.open("https://www.google.com")

        elif 'open stack overflow' in query:
            webbrowser.open("https://www.stackoverflow.com")

        elif 'time' in query:
            currentTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"The time is {currentTime}")

        elif 'date' in query:
            currentDate = datetime.datetime.now().strftime("%D:%M:%Y")
            speak(f"The date is {currentDate}")

        elif any(word in query for word in ['up', 'increase']):
            pyautogui.press("volumeup")

        elif any(word in query for word in ['down', 'decrease']):
            pyautogui.press("volumedown")

        elif 'mute' in query:
            pyautogui.press("volumemute")

        elif 'play' in query:
            query = query.replace("play", "")
            keywords = "+".join(query.split())
            speak(f"Playing... {query}")
            pywhatkit.playonyt(keywords)

        elif 'temperature' in query:
            city = "Chennai"
            search = f"temperature in {city}"
            url = f"https://www.google.com/search?q={search}"
            r = requests.get(url)
```

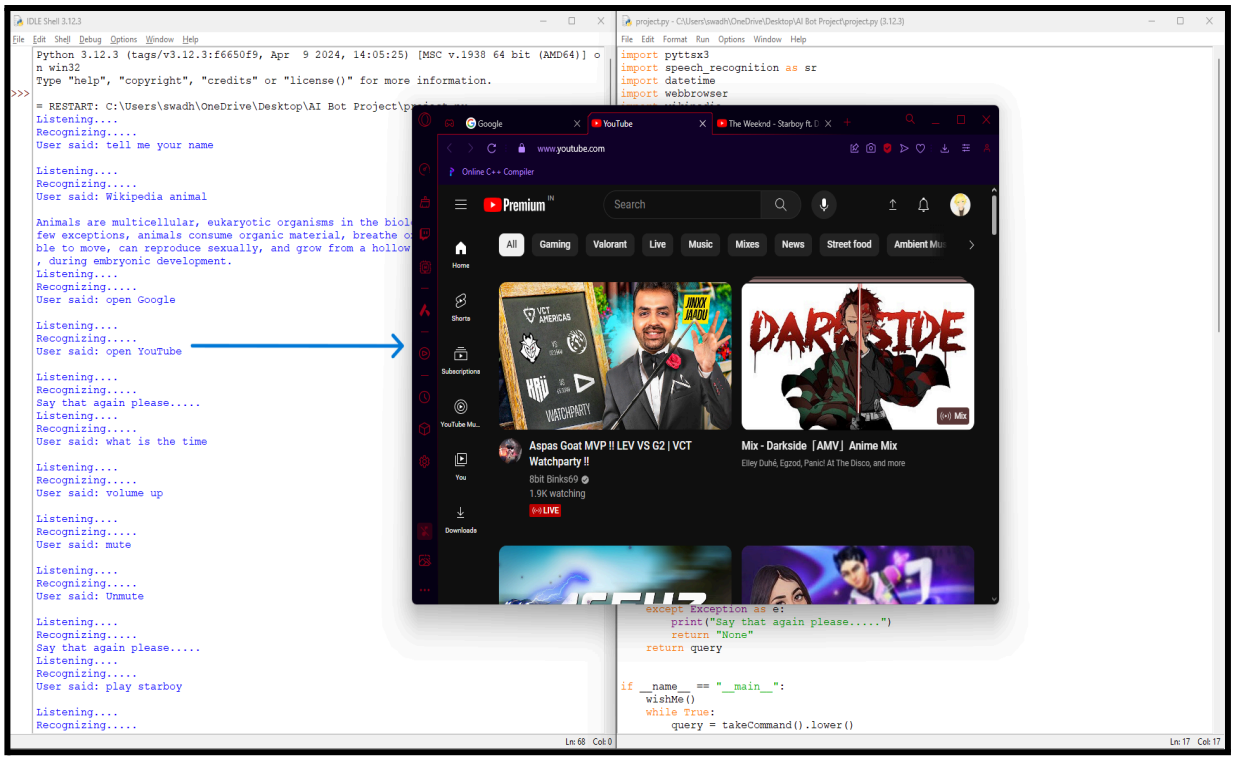
3. VOLUME UP



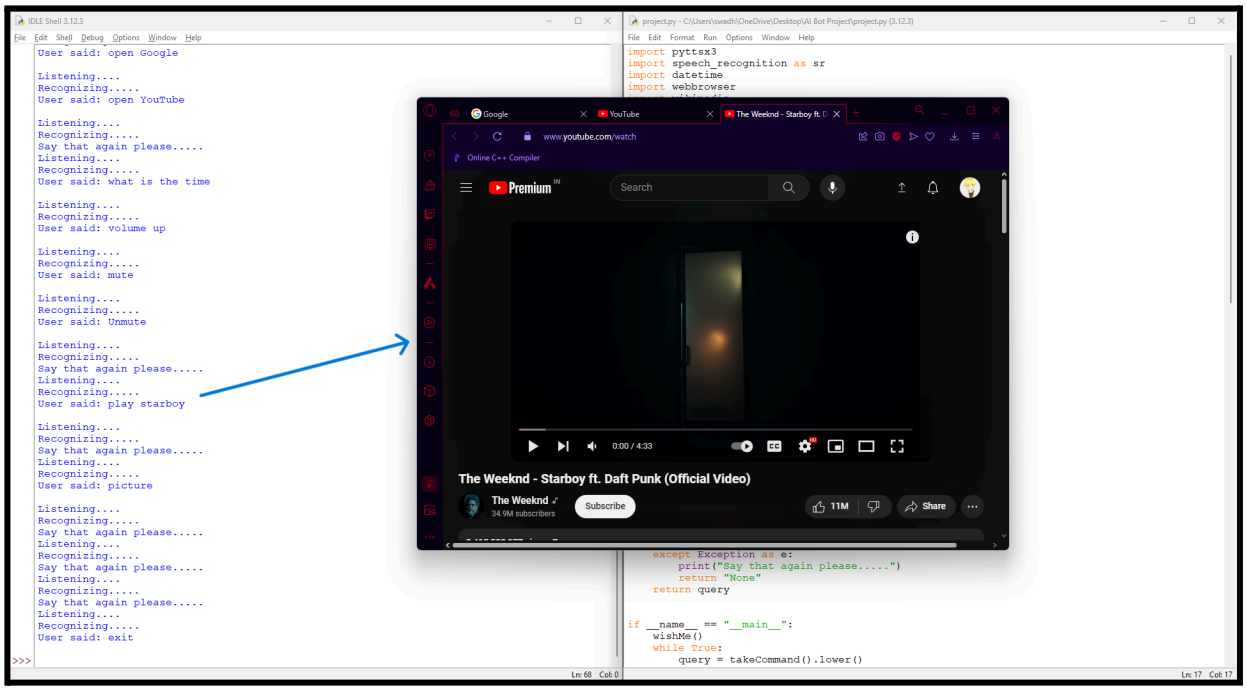
4. OPEN GOOGLE



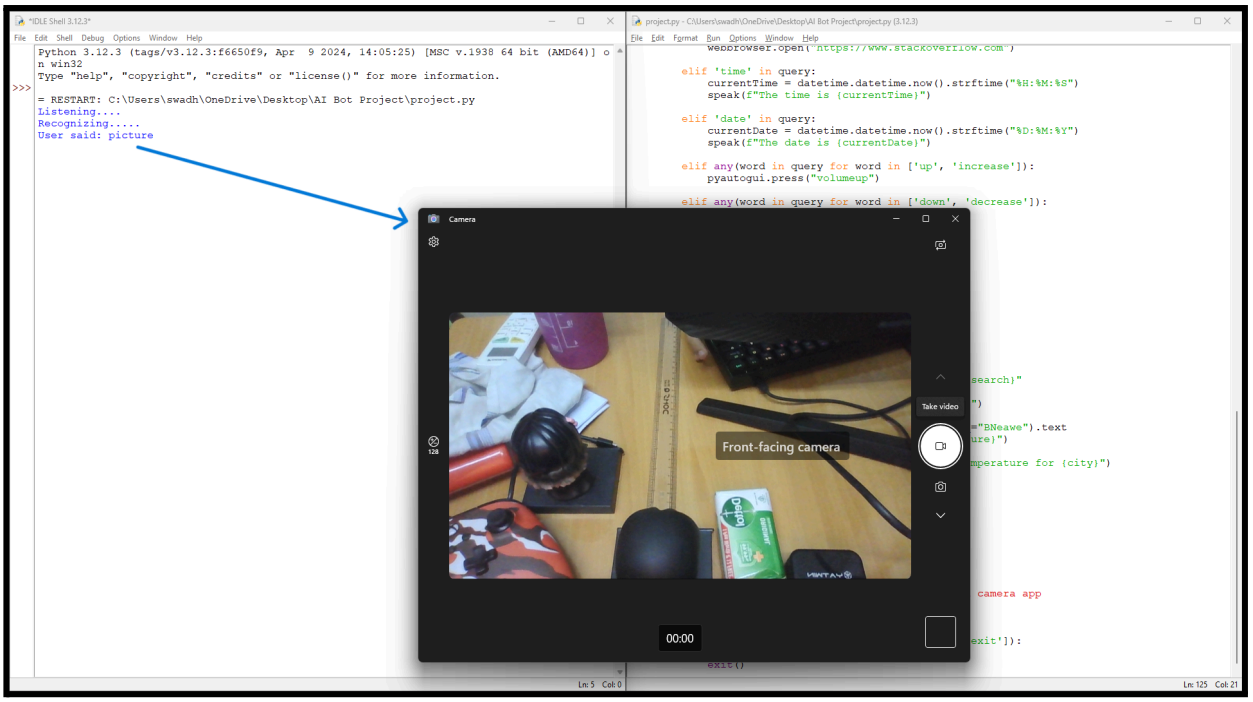
5. OPEN YOUTUBE



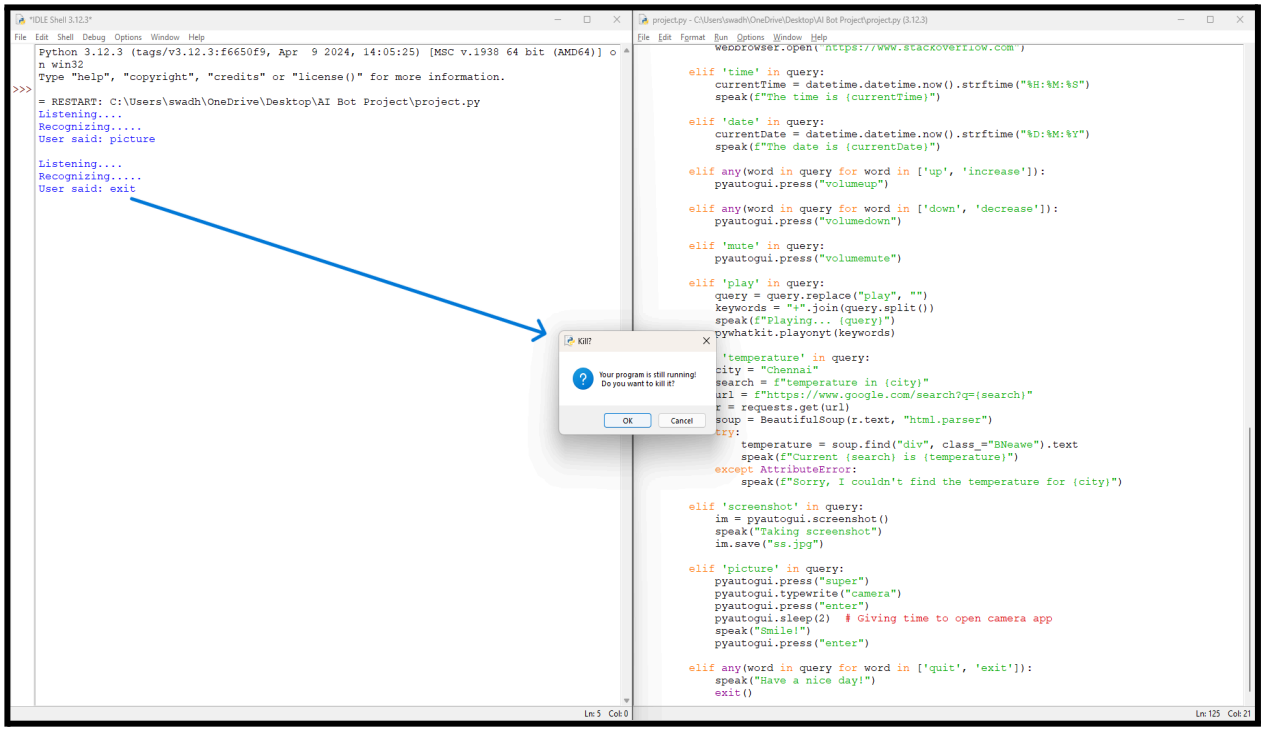
6. PLAYING MUSIC



7. TAKE PICTURE



8. EXIT



CONCLUSION

In conclusion, the development of the AI virtual voice assistant represents a significant milestone in leveraging artificial intelligence technology to enhance user interactions and streamline daily tasks. Through the integration of speech recognition and task execution modules, the assistant offers a seamless and intuitive interface for users to interact with technology using voice commands.

Throughout the project, we have successfully achieved the objectives outlined in the project scope. The assistant is capable of performing a wide range of tasks, including playing music, conducting web searches, sending emails, accessing weather forecasts, setting reminders, and engaging in basic conversations. Its ability to understand natural language and adapt to user preferences enhances its usability and effectiveness in assisting users with their needs.

With further advancements in AI technology and ongoing refinements to the assistant's capabilities, we anticipate even greater opportunities to enhance productivity, convenience, and efficiency for users across various domains.

Therefore, the AI virtual voice assistant project underscores the transformative power of AI technology in simplifying human-computer interaction and empowering users to accomplish more with less effort. As we continue to push the boundaries of innovation, we look forward to the continued evolution and impact of AI-powered virtual assistants in shaping the future of human-machine collaboration.

REFERENCES

- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9885725
- https://ieeexplore.ieee.org/document/10369477
- https://ieeexplore.ieee.org/document/10067741
- https://ieeexplore.ieee.org/document/9788267
- https://ieeexplore.ieee.org/document/10141447