

# PERSONALIZED VIRTUAL VOICE ASSISTANT

## INTRODUCTION:

The key here is voice. A voice assistant is a digital assistant that uses voice recognition, speech synthesis, and natural language processing (NLP) to provide a service through a particular application.

## OBJECTIVES:

The virtual voice assistant is developed to perform simple personalized task like general interaction,jokes,open apps, news(feeds),play songs by using alogorithms and import all the necessary libraries implemeented us1ing python programming languages.

## ABSTRACT:

The aim of our research paper is to represent our project which makes to perform simple personalized task in daily life using voice commands .This project is developed by using integrating machine learning libraries in python. Python is most convenient and simple programming to language to perform ML task. Moreover python has the largest number of ML packages/libraries which makes programming in ML/AI more simple and convenient.The libraries used in our projects are speech\_recognition,pyttsx3,subprocess,pyjokes,pywhatkit,webbrowser,os,time,ecapture,wolframalpha which is responsed for doing voice commands, playing songs, tells the jokes and open the apps. The project is fitted inside the software module.The project is designed to perform day to day simple personalized task.

## PACKAGES REQUIRED:

- 1- Speech recognition — Speech recognition is an important feature used in house automation and in artificial intelligence devices. The main function of this library is it tries to understand whatever the humans speak and converts the speech to text.
- 2- pyttsx3 — pyttsx3 is a text to speech conversion library in python. This package supports text to speech engines on Mac os x, Windows and on Linux.
- 3- wikipedia — Wikipedia is a multilingual online encyclopedia used by many people from academic community ranging from freshmen to students to professors who wants to gain information over a particular topic. This package in python extracts data's required from Wikipedia.
- 4- datetime — This is an inbuilt module in python and it works on date and time.
- 5- os — This module is a standard library in python and it provides the function to interact with operating system.
- 6- time — The time module helps us to display time.
- 7- Web browser — This is an in-built package in python. It extracts data from the web.
- 8- Subprocess — This is a standard library use to process various system commands like to log off or to restart your PC.
- 9- Json- The json module is used for storing and exchanging data.
- 10- request- The request module is used to send all types of HTTP request. Its accepts URL as parameters and gives access to the given URL'S.

11- wolfram alpha — Wolfram Alpha is an API which can compute expert-level answers using Wolfram's algorithms, knowledge base and AI technology. It is made possible by the Wolfram Language.

#### IMPLEMENTATION:

```
In [11]: import speech_recognition as sr
import pytttsx3
import datetime
import wikipedia
import webbrowser
import os
import time
import subprocess
import wolframalpha
import json
import requests
```

#### SETTING UP THE SPEECH ENGINE:

The pytttsx3 module is stored in a variable name engine.

Sapi5 is a Microsoft Text to speech engine used for voice recognition.

The voice Id can be set as either 0 or 1,

0 indicates Male voice

1 indicates Female voice

```
In [12]: engine=pytttsx3.init('sapi5')
voices=engine.getProperty('voices')
engine.setProperty('voice',voices[1].id)
```

Define a function speak which converts text to speech. The speak function takes the text as its argument,further initialize the engine.

runAndWait: This function Blocks while processing all currently queued commands. It Invokes callbacks for engine notifications appropriately and returns back when all commands queued before this call are emptied from the queue.

```
In [13]: def speak(text):
engine.say(text)
engine.runAndWait()
```

#### INITIATE A FUNCTION TO GREET THE USER:

Define a function wishMe for the AI assistant to greet the user.

The now().hour function abstract's the hour from the current time.

If the hour is greater than zero and less than 12, the voice assistant wishes you with the message "Good Morning".

If the hour is greater than 12 and less than 18, the voice assistant wishes you with the following message "Good Afternoon".

Else it voices out the message "Good evening"

```
In [14]: def wishMe():
hour=datetime.datetime.now().hour
if hour>=0 and hour<12:
    speak("Hello,Good Morning")
    print("Hello,Good Morning")
elif hour>=12 and hour<18:
    speak("Hello,Good Afternoon")
    print("Hello,Good Afternoon")
else:
    speak("Hello,Good Evening")
    print("Hello,Good Evening")
```

## SETTING UP THE COMMAND FUNCTION FOR YOUR AI ASSISTANT :

Define a function takecommand for the AI assistant to understand and to accept human language. The microphone captures the human speech and the recognizer recognizes the speech to give a response.

The exception handling is used to handle the exception during the run time error and,the recognize\_google function uses google audio to recognize speech.

```
In [15]: def takeCommand():
r=sr.Recognizer()
with sr.Microphone() as source:
    print("Listening...")
    audio=r.listen(source)

    try:
        statement=r.recognize_google(audio,language='en-in')
        print(f"user said:{statement}\n")

    except Exception as e:
        speak("Pardon me, please say that again")
        return "None"
    return statement

print("Loading your AI personal assistant Nancy")
speak("Loading your AI personal assistant Nancy")
wishMe()
```

```
Loading your AI personal assistant Nancy
Hello,Good Evening
```

## THE MAIN FUNCTION:

The main function starts from here,the commands given by the user is stored in the variable statement.

```
In [ ]: if __name__=='__main__':

while True:
    speak("Tell me how can I help you now?")
    statement = takeCommand().lower()
    if statement==0:
        continue
```

```
Listening...
```

If the following trigger words are there in the statement given by the users it invokes the virtual assistant to speak the below following commands.

```
In [ ]:
```

```

if "good bye" in statement or "ok bye" in statement or "stop" in statement:
    speak('your personal assistant Nancy is shutting down,Good bye')
    print('your personal assistant Nancy is shutting down,Good bye')
    break

```

## FETCHING DATA FROM WIKIPEDIA:

The following commands helps to extract information from wikipedia. The wikipedia.summary() function takes two arguments, the statement given by the user and how many sentences from wikipedia is needed to be extracted is stored in a variable result.

```

In [ ]: if 'wikipedia' in statement:
        speak('Searching Wikipedia...')
        statement =statement.replace("wikipedia", "")
        results = wikipedia.summary(statement, sentences=3)
        speak("According to Wikipedia")
        print(results)
        speak(results)

```

## ACCESSING THE WEB BROWSERS — GOOGLE CHROME , G-MAIL AND YOU TUBE:

The web browser extracts data from web. The open\_new\_tab function accepts URL as a parameter that needs to be accessed.

The Python time sleep function is used to add delay in the execution of a program. We can use this function to halt the execution of the program for given time in seconds.

```

In [ ]: elif 'open youtube' in statement:
        webbrowser.open_new_tab("https://www.youtube.com")
        speak("youtube is open now")
        time.sleep(5)

        elif 'open google' in statement:
        webbrowser.open_new_tab("https://www.google.com")
        speak("Google chrome is open now")
        time.sleep(5)

        elif 'open gmail' in statement:
        webbrowser.open_new_tab("gmail.com")
        speak("Google Mail open now")
        time.sleep(5)

```

## PREDICTING TIME:

The current time is abstracted from datetime.now() function which displays the hour, minute and second and is stored in a variable name strTime.

```

In [ ]: elif 'time' in statement:
        strTime=datetime.datetime.now().strftime("%H:%M:%S")
        speak(f"the time is {strTime}")

```

## TO FETCH LATEST NEWS:

If the user wants to know the latest news , The voice assistant is programmed to fetch top headline news from Time of India by using the web browser function.

```

In [ ]: elif 'news' in statement:

```

```
news = webbrowser.open_new_tab("https://timesofindia.indiatimes.com/home/headl
speak('Here are some headlines from the Times of India,Happy reading')
time.sleep(6)
```

## SEARCHING DATA from WEB:

From the web browser you can search required data by passing the user statement (command) to the open\_new\_tab() function.

User: Hey Nancy, please search images of butterfly

The Voice assistant opens the google window & fetches butterfly images from web.

```
In [ ]: elif 'search' in statement:
        statement = statement.replace("search", "")
        webbrowser.open_new_tab(statement)
        time.sleep(5)
```

## SETTING YOUR AI ASSISTANT TO ANSWER GEOGRAPHICAL AND COMPUTATIONAL QUESTIONS:

Here we can use a third party API called Wolfram alpha API to answer computational and geographical questions. It is made possible by the Wolfram Language. The client is an instance (class) created for wolfram alpha. The res variable stores the response given by the wolfram alpha.

```
In [ ]: elif 'ask' in statement:
        speak('I can answer to computational and geographical questions and what ques
question=takeCommand()
app_id="Paste your unique ID here "
client = wolframalpha.Client('R2K75H-7ELALHR35X')
res = client.query(question)
answer = next(res.results).text
speak(answer)
print(answer)
```

It would be interesting to program your AI assistant to answer the following questions like what it can and who created it, isn't it?

```
In [ ]: elif 'who are you' in statement or 'what can you do' in statement or 'Tell me about you'
        speak('I am Nancy version 1 point 0 your personal assistant. I am programmed t
        'opening youtube,google chrome, gmail and stackoverflow ,predict time,ta
        'In different cities, get top headline news from times of india and you

        elif "who made you" in statement or "who created you" in statement or "who discove
        speak("I was built by Nithish, Sivaranjani, Sneha")
        print("I was built by Nithish, Sivaranjani, Sneha")
```

## DETECT WEATHER:

Program your AI assistant to detect weather we need to generate an API key from Open Weather map.

Open weather map is an online service which provides weather data. By generating an API ID in the official website you can use the APP\_ID to make your voice assistant detect weather of all places whenever required. The necessary modules needed to be imported for this weather detection is json and request module.

The city\_name variable takes the command given by the human using the takeCommand() function.

The get method of request module returns a response object. And the json methods of response object converts json format data into python format.

The variable X contains list of nested dictionaries which checks whether the value of 'COD' is 404 or not that is if the city is found or not.

The values such as temperature and humidity is stored in the main key of variable Y.

```
In [ ]: elif "weather" in statement:
        api_key="Apply your unique ID"
        base_url="https://api.openweathermap.org/data/2.5/weather?"
        speak("what is the city name")
        city_name=takeCommand()
        complete_url=base_url+"appid="+api_key+"&q="+city_name
        response = requests.get(complete_url)
        x=response.json()
        if x["cod"]!="404":
            y=x["main"]
            current_temperature = y["temp"]
            current_humidiy = y["humidity"]
            z = x["weather"]
            weather_description = z[0]["description"]
            speak(" Temperature in kelvin unit is " +
                    str(current_temperature) +
                    "\n humidity in percentage is " +
                    str(current_humidiy) +
                    "\n description  " +
                    str(weather_description))
            print(" Temperature in kelvin unit = " +
                    str(current_temperature) +
                    "\n humidity (in percentage) = " +
                    str(current_humidiy) +
                    "\n description = " +
                    str(weather_description))
```

USER: Hey Nancy ,I want to get the weather data

NANCY: What is the city name?

USER: Himachal Pradesh

NANCY: Temperature in kelvin unit is \*\*, **Humidity in percentage is** and Description is \_\_\_\_.

TO LOG OFF YOUR PC:

The subprocess.call() function here is used to process the system function to log off or to turn off your PC. This invokes your AI assistant to automatically turn off your PC.

```
In [ ]: elif "log off" in statement or "sign out" in statement:
        speak("Ok , your pc will log off in 10 sec make sure you exit from all applica
        subprocess.call(["shutdown", "/l"])

        time.sleep(3)
```

CONCLUTION:

We have provided a very effective solution to develop an intelligent system for personalized virtual voice assistant using python programming.