ARTIFICIAL INTELLIGENCE

LAB PROJECT SUBMISSION

Submitted to : Paluck mam

Submitted by : Abhishek Meena [102117072]

Problem statement :

We are focused on developing a voice assistant that can understand and respond to user queries and provide helpful and relevant information or assistance.

Description of problem :

Voice assistants have become increasingly popular in recent years due to the convenience and ease they offer users in performing various tasks. They are designed to interact with users using natural language and provide personalized responses based on the user's input.

Code and Explanation :

The following is my approach :

Firstly I imported the following modules :

```
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
```

Brief description of the imported modules :

1. **Pyttsx3** - pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.
2. **Speech_recognition** - Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. This article

aims to provide an introduction to how to make use of the SpeechRecognition library of Python. This is useful as it can be used on microcontrollers such as Raspberry Pi with the help of an external microphone.

3. **Datetime** - Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.
4. **Wikipedia** - Wikipedia is a multilingual online encyclopedia created and maintained as an open collaboration project by a community of volunteer editors using a wiki-based editing system.
5. **Webbrowser** - In Python, webbrowser module is a convenient web browser controller. It provides a high-level interface that allows displaying Web-based documents to users.
6. **os** - The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules.

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

What is sapi5?

- Microsoft developed speech API.
- Helps in synthesis and recognition of voice.

What Is VoiceId?

Voice id helps us to select different voices.

- voice[0].id = Male voice
- voice[1].id = Female voice

Defining Take command Function :

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system. So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.

```
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
```

Defining Task 1: To search something on Wikipedia

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. After successfully installing the Wikipedia module, import it into the program by writing an import statement.

```
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)
```

Defining Task 2: To open YouTube site in a web-browser

To open any website, we need to import a module called webbrowser. It is an in-built module, and we do not need to install it with a pip statement; we can directly import it into our program by writing an import statement.

```
elif 'open youtube' in query:
            webbrowser.open("youtube.com")
```

Defining Task 3: To open Google site in a web-browser

```
elif 'open google' in query:
            webbrowser.open("google.com")
```

Defining Task 4: To play music

In the above code, we first opened our music directory and then listed all the songs present in the directory with the os module's help. With the help of os.startfile, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module.

```
elif 'play music' in query:
            music_dir = 'D:\\Non Critical\\songs\\Favorite Songs2'
            songs = os.listdir(music_dir)
            print(songs)
            os.startfile(os.path.join(music_dir, songs[0]))
```

Defining Task 5: To know the current time

In the above, code we are using the datetime() function and storing the current or live system time into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into speech**.**

```
elif 'the time' in query:
            strTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"Sir, the time is {strTime}")
```

Defining Task 6: To open the Code Blocks Program

```
elif 'open code blocks' in query:
            codePath = "C:\Program Files\CodeBlocks\codeblocks.exe"
            os.startfile(codePath)
```

Defining Send email function :

To send an email, we need to import a module called smtplib.

What is smtplib?

Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and route emails between mail servers. An instance method called sendmail is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:

The sender: Email address of the sender.

The receiver: T Email of the receiver.

The message: A string message which needs to be sent to one or more than one recipient.

```
def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('iamabhishek404@gmail.com', 'xxxxxxx')
    server.sendmail('iamabhishek404 @gmail.com', to, content)
    server.close()
```

Calling sendEmail() function inside the main() function:

```
elif 'email to harry' in query:
            try:
                speak("What should I say?")
                content = takeCommand()
                to = "ameena_be21@thapar.edu"
                sendEmail(to, content)
                speak("Email has been sent!")
            except Exception as e:
                print(e)
                speak("Sorry my friend Abhishek. I am not able to send this
email")
```