|  |  |  |
| --- | --- | --- |
| **Tech Saksham**  Final Project Report  **Python** |  |  |

**“PERSONAL VOICE ASSISTANT”**

**“KIET BATCH-1”**

|  |  |
| --- | --- |
| **ROLL NO** | **NAME** |
| 19JN1A0513 | ADAPA VIJAYA DURGA NARISHMA |
| 19JN1A0559 | LANKALAPALLI SAILAJA |
| 19JN1A05D8 | ADAPA JANAKI RATNA MAHESWARI |
| 19JN1A0549 | DULLA NAGA SAI DEEPIKA |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | Mr. Raja |
|  |  | Master Trainer |

**ABSTRACT**

Digitization brings new possibilities to ease our daily life activities by the means of assistive technology. Amazon Alexa, Apple Siri, Microsoft Cortana, Samsung Bixby, to name only a few were successful in the age of smart personal assistants (spas).A voice assistant is defined a digital assistant that combines artificial intelligence, machine learning Speech Recognition, Natural Language Processing (NLP), Speech Synthesis and various actuation mechanisms to sense and influence the environment. We use different NLP techniques to convert Speech to text (STT), then process the text, convert Text to Speech (TTS), add various functionalities. However, SPA research seems to be highly fragmented among different disciplines, such as computer science, human-computer-interaction and information systems, which leads to ‘reinventing the wheel approaches’ and thus impede progress and conceptual clarity.

In this paper, we present an exhaustive, integrative literature review to build a solid basis for future research. Hence, we contribute by providing a consolidated, integrated view on prior research and lay the foundation for an SPA classification scheme.

Instead of pattern recognition we use NLP techniques to recognize the text which is context based. Operates online as well as offline. Data is Stored in Application itself, reduces Time and Space Complexity.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Table of Contents** | **Page No.** |
| 1 | Chapter 1: Introduction | 1-6 |
| 2 | Chapter 2: Services and Tools Required | 7-9 |
| 3 | Chapter 3: Project Architecture | 10 |
| 4 | Chapter 4: Architecture Blocks Detail Working | 11 |
| 5 | Conclusion | 12 |
| 6 | References | 13 |
| 7 | Code | 14-22 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Overview**
  2. **Feature**
  3. **Advantages**
  4. **Scope**
  5. **Future Work**
  6. **OVERVIEW**

The overall system design consists of following phases:

1. Data collection in the form of speech.

2. Voice analysis and conversion to text

3. Data storage and processing

4. Generating speech from the processed text output

In first phase, the data is collected in the form of speech and stored as an input for the next phase for processing. In second phase, the input voice is continuously processed and converted to text using speech to text (STT). In next phase the converted text is analyzed and processed using Python Script and NLP techniques to identify the response to be taken against the command. Finally once the response is identified, output is generated from simple text to speech conversion using text to speech (TTS).

Problems with Existing System:

Despite the various benefits provided by speech recognition, the system is also plagued with limitations. By implication the development of speech recognition applications also inherits these limitations. The existing Voice Assistants use pattern recognition techniques of python which lack in the context, Lack of accuracy, and misinterpretations, Time, costs and productivity, User accents. They operate only on online mode. They store the data in database servers which leads to increase in Time and Space Complexity. Some of them use cloud to store the data which leads to security issues. Background noise interference is also another daunting problem with speech recognition software.

Proposed System:

The proposed system of voice assistant will solve some issues of existing system as well introduce new features for better quality and usage. So, let’s have a brief of the new updated version of the voice assistant. Instead of pattern recognition technique which has been used in previous models, we use Natural Language Processing (NLP) techniques to recognize the text which is context based rather the usual pattern based. This Operates in online as well as offline mode. System application runs on offline mode, whereas web based operations run on online mode. Data is Stored in Application itself, rather than cloud which reduces Time and Space Complexity. It even reduces the economic cost due to reducing high bundles of data usage.

Process Logic:

Personal Voice Assistant is developed as a desktop application with the help of Natural Language Processing which helps to send messages and use various built-in systems based and web based applications using voice commands.

The Voice Assistant performs basic operations such as controlling computer tasks and operations, asking for temperature, humidity, date, time, and year. Adding, reading and deleting notes using voice commands and playing YouTube videos on demand.

The above tasks can be performed using certain methodologies in which

each technique has its own functionality and different operations to be performed. Each technique has different process logic to be executed.

Techniques to be implemented:

1. Speech Recognition

2. NLP

3. Threading

4.Scraping

* 1. **FEATURE SELECTION**

Smartphones and tablets are equipped with technology that allows you to operate them hands-free, by using your voice. These features work in different ways, one works as a “personal assistant”, performing full tasks for you, i.e. send a text message, check stuff online, or tell you the weather. The other one allows you to select and open apps, dictate text in text fields, send, tap, scroll, and other tasks. It works as a “replacement” of your fingers but requires you to indicate the exact action to complete.

## **1. Voice Control or Voice Access**

You can control your device by telling it what to do, in a step-by-step process with any element shown on the screen; those actions include “select”, “open”, “type”, “send”, “scroll”, “go back” and “go home” among other commands. This feature is useful for people with conditions that affect their upper limb mobility or dexterity and it is intended to replace your physical interaction with the touch screen, allowing you to fully control of your phone.

Once you activate voice access or voice control, your device will assign numbers to your apps and settings, and then you can tell your phone which app to select and open by using simple commands i.e. “open number five” so your device opens the app tagged with number five. Once the app opens, new numbers will appear on the elements of that app so you can continue repeating this process until you have completed a full task. Some of these tasks are combined with the ones from the personal assistant of your device for increased accessibility. The following video might provide further explanation to this access feature:

To install Voices Access (Android) or Voice Control (IOS)

* iOS: Go to Settings > Accessibility > Select Voice Control >Set up Voice control.
* Android (version 5 and above): Download Voice Access from the Gooogle Play store. Then Go to Settings > Accessibility > tap on Voice Access>Tap the On/Off Switch.

## **2.** **Personal or Virtual Assistants**

Personal Assistants allow you to control you smart device using voice commands and speech recognition, useful for individuals with vision, mobility, and cognitive disabilities. Personal assistants can complete full tasks with just one command, such as telling you the weather, looking up stuff on the internet, setting alarms, making phone calls. The personal assistant will usually confirm each task by telling you that the task has been initiated or completed.

Personal Assistants come built-in to your devices and they have their own names:  “Siri” for iOS devices, “Hey Google” for Android, and “Cortana” for Windows 10 devices.

**1.3 ADVANTAGES**

* They Can Speak Multiple Languages
* They are Easy to Use with GPS Tracker Software
* They Provide Instant Access to Information
* Easy to Manage Timesheet with Voice Assistant
* We can Use Them Anywhere, Anytime
* They Make Common Grammatical Errors
* They Incur High Costs
* Fast response
* Easy for children
* Hands-free
* Minimal Effort
  1. **SCOPE**

Modern voice assistants can do almost anything you might be able to think of. You can use them to play music, answer your questions, make phone calls, and even control parts of your home, such as the lights or thermostat. And they are so convenient to use too - whether you want to start the coffee machine or check the news, the assistant will report back to you in seconds.

72% of voice assistant owners admit that their devices quickly became an essential part of their daily routines and that they wouldn’t want to return to “regular” ways of controlling their appliances. What’s more, 55% of teenagers say that they use voice search every day.

Since Apple introduced Siri and soon after that Google and Amazon introduced their own voice assistants, the capabilities and popularity of voice assistants and voice search have been growing steadily.

Alexa, for example, already has  100,000 skills available through which you can control your home devices, order pizza, play trivia games, listen to the news and weather, or shop online. Google meanwhile boasts that its assistant has more  than one million actions available and supports over 100 languages.

As for Siri,  around 500 million people use it on their phones. In fact, the latest voice

* 1. **FUTURE WORK**

Technological advances are making voice assistants more capable, particularly in AI, natural language processing (NLP), and machine learning. To build a robust speech recognition experience, the artificial intelligence behind it must become better at handling challenges such as accents and background noise. And, as consumers are becoming increasingly more comfortable and reliant upon using voice to talk to their phones, cars, smart home devices, etc., voice technology will become a primary interface to the digital world and with it, expertise for voice interface design and voice app development will be in greater demand.

Based on the survey we recommend that the application should be developed which accomplishes the desire of different users. The main reason that the

user wants to use the voice assistant is to make their life easier, so by implementing the below mentioned features the user can be facilitated.

1. Developing for different languages and different accents.

2. Portability for any environment.

3. Voice authentication technology can be implemented for more security.

4. Chatbot implementation requires corpus.

5. Dialogue flow needs stack with neurals

6. Deploy on web using flask or Django

7. Deploy on cloud uses amazon ec2, Heroku.

8.NLP features such as finding entities, topic modelling.

**CHAPTER 2**

**SERVICES AND TOOLS REQUIRED**

**2.1 Functional Requirements**

**2.2 Non-Functional Requirements**

**2.3 Domain Requirements**

**2.4 Hardware & Software Requirements**

**2.1 FUNCTIONAL REQUIREMENTS**

FR 1.1: Detect Position My Guide should be able to detect continuously the current position of the user inside Politecnico di Torino hallways.

FR 1.2: Update Route My Guide is able to update the route (shortest) continuously according to detected position of user and dead ends. If encountered with a dead end the system can re-calibrate a new route to the end destination.

FR 1.2: Route Guidance The mobile application communicates the route to the user through voice commands. It notifies of upcoming turns through hallways, doors, and upon arrival to end destination.

FR 3.1: Generating Route According to the detected position of user inside a hallway and the specified end destination by the user (inputted as text or voice ), the system generates a route between the 2 points according to the shortest path.

FR 4: Switch MyGuide Cane ON/OFF User is able to activate/deactivate the vibrating cane anytime during the usage of the system depending on the personal preference of the user.

**2.2 NON-FUNCTIONAL REQUIREMENTS**

1. Supporting Technologies MyGuide implementation should be feasible using technologies that are accessible to the end-users.

2. Device Software Compatibility The mobile interfaces must be compatible with Android

3. Obstacle detection The system allows for obstacle detection within the range of 1 meters.

4. Language The language should be localized to the preference of the user.

5. Time Response My Guide must perform in a proper time constraint that reflects average walking speed, motion and obstacles in the environment.

6. Multi User System My Guide is able to consider the presence of more than one user in the same environment. All the features of the system should operate properly for all users.

**2.3 DOMAIN REQUIREMENTS**

Quickly and accurately transcribe audio to text in more than 100 languages and varients. Customize models to enhance accuracy for domain-specific terminology.

Get more values from spoken audio by enabling search or analytics on transcribed text or facilitating actio-- all in the preferred programming language.

And also having High quality transcription , customizable models, flexible deployment, production ready.

**2.4 HARDWARE AND SOFTWARE REQUIREMENTS**

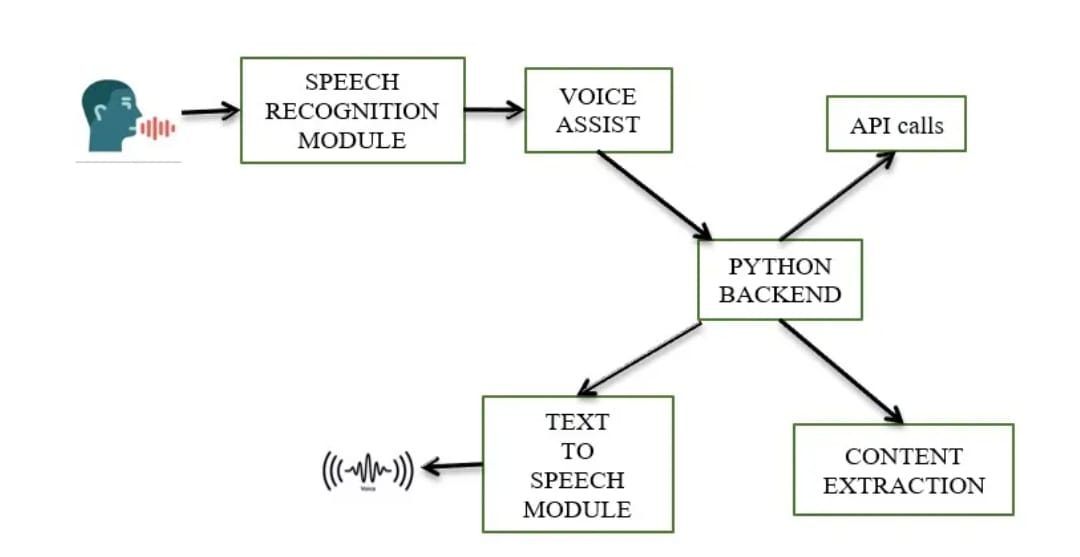
Software requirements: Python (Pycharm)

Hardware requirements: Windows 8 and above, intel core i3 and above, RAM: 4GB or more

**CHAPTER 3**

**PROJECT ARCHITECTURE**

**3. Architecture**

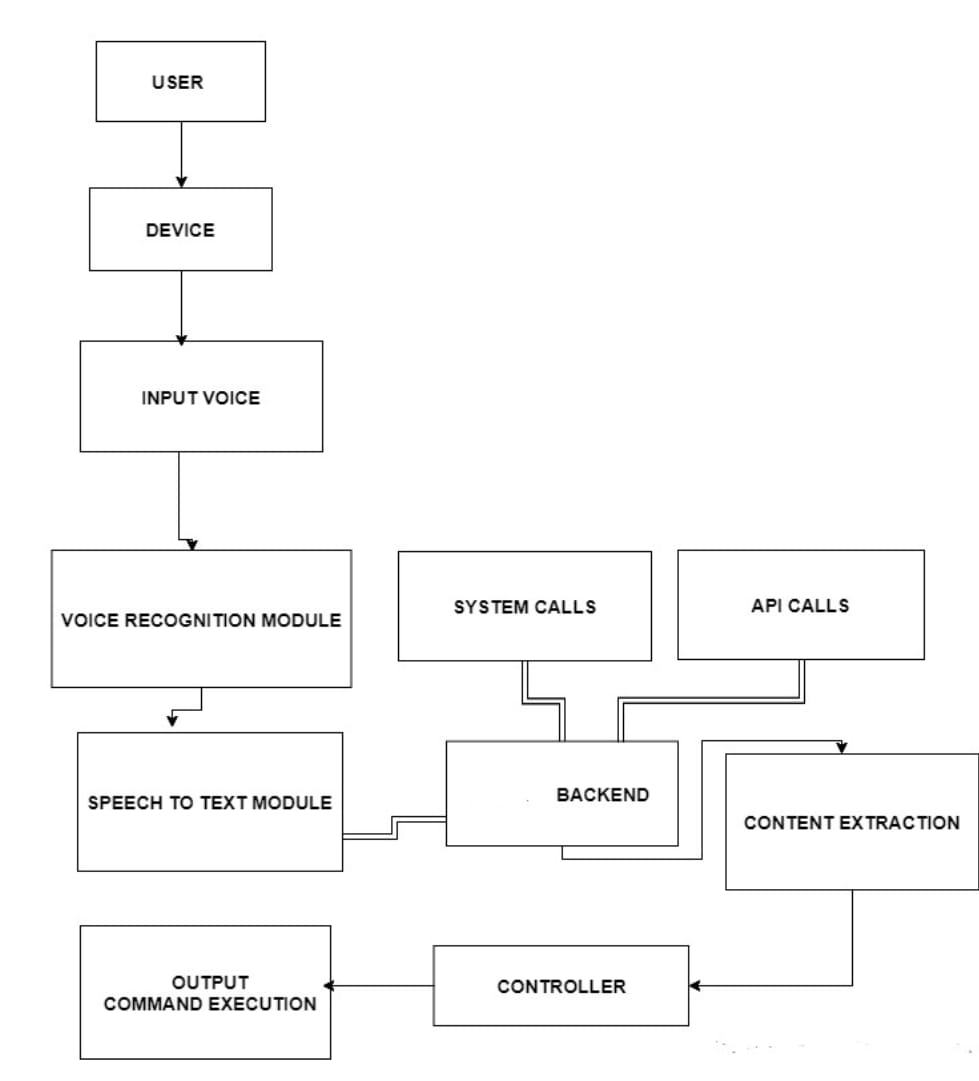
****

|  |  |  |
| --- | --- | --- |
|  |  |  |

**CHAPTER 4**

**ARCHITECTURE BLOCKS DETAIL WORKING**

**4. Blocks**

****

**CONCLUSION**

Voice Controlled Personal Assistant System will use the Natural language processing and can be integrated with Machine learning techniques to achieve a smart assistant that can perform action on various applications and will make human life comfortable. The system will have the following phases: Data collection in the form of voice; Voice analysis and conversion to text; Data storage and processing; generating speech from the processed text output. This application will also make life easier for those who are physically disabled and every common user who is fascinated by voice recognition. Academically, raising awareness for systems like this for students can give them better understanding of topics like Artificial Intelligence, Neural Networks, Natural Language Processing, Machine Learning and Human Computer Interaction and also how to improve user experience in application development. The formulated solution is able to process voice commands offline allowing users to cut down on the cost of data bundles. This also helps to make it faster in comparison to alternative applications like Apple’s Siri, Google assistant, etc. Moreover, the solution is capable of carrying out a variety of tasks with ease such as telling the date and time, playing music/videos, making phone calls ,finding weather, temperature, googling information etc.. This paper can also act as a prototype for many advanced applications.

**REFERENCES**

* DOUGLAS O’SHAUGHNESSY, SENIOR MEMBER, IEEE, “Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis” proceedings of THE IEEE, VOL. 91, NO. 9, SEPTEMBER 2003
* Nil Goksel-Canbek2 Mehmet EminMutlu,”On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant” International Journal of Human Sciences, 13(1), 592-601. doi:10.14687/ijhs.v13i1.3549.
* Easwara Moorthy, A., Vu, K.-P.L.: Privacy Concerns for Use of Voice Activated Personal Assistant in the Public Space. International Journal of Human-Computer Interaction 31, 307–335 (2015)
* Tsiao, J.C.-S., Tong, P.P., Chao, D.Y.: NaturalLanguage Voice-Activated Personal Assistant, United States Patent (10), Patent No.: US 7,216,080 B2 (45), 8 May 2007
* Lopez, G., Quesada, L., Guerrero, L.A.: Alexa vs Siri vs Cortana vs Google assistant: a comparison of speech-based natural user interfaces. Conference Paper, January 2018
* Kepuska, V., Bohouta, G.: Next generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). In: IEEE Conference (2018)
* Gong, L.: San Francisco, CA (US) United States US 2003.01671.67A1 (12) Patent Application Publication c (10) Pub. No.: US 2003/0167167 A1 Gong (43) Pub. Date: 4 September 2003 for Intelligent Virtual Assistant
* Sumitkumar Sarda, Yash Shah, Monika Das, Nikita Saibewar, ShivprasadPatil,”VPA: Virtual Personal Assistant” Published in 2017 International Journal of Computer Applications (0975 – 8887) Volume 165 – No 1.
* Grabianowski, E. (2011). How Speech Recognition Works. Retrieved February 2016, from How Stuff Works:

<http://electronics.howstuffworks.com/gadgets/h>

* Nagesh Singh Chauhan,“Build Your First Voice Assistant”,March,2019

<https://towardsdatascience.com/build-your-firstvoiceassistant>

**CODE**

**import pyttsx3**

**import requests**

**import speech recognition as sr**

**import datetime**

**import os**

**from requests import get**

**import wikipedia as wiki**

**import webbrowser**

**import pywhatkit as pk**

**import sys**

**import time**

**import pyjokes**

**import pyautogui**

**engine = pyttsx3.init('sapi5')**

**voices = engine.getProperty('voices')**

**#print(voices[0].id)**

**engine.setProperty('voices', voices[1].id)**

**engine.setProperty('rate',200)**

**#text to speech**

**def speak(audio):**

**engine.say(audio)**

**print(audio)**

**engine.runAndWait()**

**# To convert voice into text**

**def takecommand():**

**r = sr.Recognizer()**

**with sr.Microphone() as source:**

**print("Listening...")**

**r.pause\_threshold = 1**

**audio = r.listen(source, timeout=4, phrase\_time\_limit=7)**

**try:**

**print("Recognizing...")**

**query = r.recognize\_google(audio, language='en-in')**

**print(f"user said: {query}")**

**except Exception as e:**

**speak("Say that again please...")**

**return "none"**

**query = query.lower()**

**return query**

**#to wish**

**def wish():**

**hour = int(datetime.datetime.now().hour)**

**tt = time.strftime("%I:%M %p")**

**if hour>=0 and hour<12:**

**speak(f"Good Morning Boss, its {tt}")**

**elif hour>12 and hour<18:**

**speak(f"Good Afternoon Boss, its {tt}")**

**else:**

**speak(f"Good Evening Boss, its {tt}")**

**speak(" Team hash it out. please tell me how can i help you")**

**main\_page = requests.get(main\_url).json()**

**# print(main\_page)**

**articles = main\_page["articles"]**

**# print(articles)**

**head = []**

**day=["first", "second", "third", "fourth", "fifth"]**

**for ar in articles:**

**head.append(ar["title"])**

**for i in range(len(day)):**

**# print(f"today's {day[i]} news is:", head[i])**

**speak(f"today's {day[i]} news is: {head[i]}")**

**if \_name\_ == "\_main\_":**

**wish()**

**while True:**

**# if 1:**

**query = takecommand().lower()**

**# logic building**

**if "open command prompt" in query:**

**os.system("start cmd")**

**elif "ip address" in query:**

**ip = get('https://api.ipify.org').text**

**speak(f"Your IP address is {ip}")**

**elif "who is" in query:**

**query = query.replace("who is ", "")**

**result = wiki.summary(query, sentences=2)**

**print(result)**

**speak(result)**

**elif "wikipedia" in query:**

**speak("searching wikipedia...")**

**query = query.replace("wikipedia", "")**

**result = wiki.summary(query, sentences=2)**

**speak("According to wikipedia")**

**print(result)**

**speak(result)**

**elif "open google" in query:**

**speak("Sir, what should i search on google")**

**cm = takecommand().lower()**

**webbrowser.open(f"{cm}")**

**elif "open youtube" in query:**

**webbrowser.open("www.youtube.com")**

**elif "play" in query:**

**query = query.replace("play", "")**

**speak("playing " + query)**

**pk.playonyt(query)**

**elif "open facebook" in query:**

**webbrowser.open("www.facebook.com")**

**# to find a joke**

**elif "tell me a joke" in query:**

**joke = pyjokes.get\_joke()**

**speak(joke)**

**# to close application**

**elif "close command prompt" in query:**

**speak("okay sir closing command prompt")**

**os.system("taskkill /f /im cmd.exe")**

**# to change window**

**elif "switch the window" in query:**

**pyautogui.keyDown("alt")**

**pyautogui.press("tab")**

**time.sleep(1)**

**pyautogui.keyUp("alt")**

**elif "tell me news" in query:**

**speak("please wait sir, feteching the latest news")**

**news()**

**# To find my location**

**elif "where i am" in query or "where we are" in query or "what is my location" in query:**

**speak("wait sir, let me find")**

**try:**

**ipAdd = requests.get('https://api.ipify.org').text**

**print(ipAdd)**

**url = 'https://get.geojs.io/v1/ip/geo/' + ipAdd + '.json'**

**geo\_requests = requests.get(url)**

**geo\_data = geo\_requests.json()**

**# print (geo\_data)**

**city = geo\_data['city']**

**# state = geo\_data['state']**

**country = geo\_data['country']**

**speak(f"sir i am not sure, but i think we are in {city} city of {country} country")**

**except Exception as e:**

**speak(" sorry sir, Due to network issue i am not able to find our location.")**

**pass**

**# To take screen shot**

**elif "take screenshot" in query or "take a screenshot" in query:**

**speak("sir, please tell me the name for this screenshot file")**

**name = takecommand().lower()**

**speak("please sir hold screen for few seconds , i am taking screenshot")**

**time.sleep(3)**

**img = pyautogui.screenshot()**

**img.save(f"{name}.png")**

**speak("it is done sir, the screenshot is saved in main folder. now i am ready for next command")**

**elif "hello" in query or "hey" in query:**

**speak("hello sir, may i help you something..")**

**elif "how are you" in query:**

**speak("i am fine sir, what about you.")**

**elif "fine" in query or "good" in query or "also good" in query:**

**speak("that's great to hear from you.")**

**elif "thank you" in query or "thanks" in query:**

**speak("it's my pleasure sir.")**

**elif "you can sleep" in query or "sleep now" in query:**

**speak("ok sir i am going to sleep .")**

**elif "no thanks" in query or "no" in query:**

**speak("thanks for using me sir, have a great day.")**

**sys.exit()**

**speak("sir, do yo have any other work")**

**Footer**

**© 2023 GitHub, Inc.**

**Footer navigation**

**Terms**

**Privacy**

**Security**

**Status**

**Docs**

**Contact GitHub**