Personal Voice Assistant

**Speech Recognition:** personal voice assistant uses the speech recognition library to implement speech recognition functionality. The library provides a simple and intuitive interface for capturing audio input from the user and transcribing it into text.

To use the library, your assistant begins by waiting for the user to activate it with the wake word. Once the wake word is detected, the assistant begins capturing audio input using the Microphone class from the library. The library provides a number of options for adjusting the audio capture settings, such as adjusting the input volume or setting a timeout for the capture process.

**Sending Email:** personal voice assistant allows users to send emails using voice commands. The assistant prompts the user for the email recipient, subject, and body of the message, and uses the smtplib library to securely send the email using the user's email credentials. This feature provides a convenient and hands-free way for users to send emails quickly and efficiently.

**Alarm Function:** This function writes a user-specified time to a text file called "Alarmtext.txt" using the open function with the "a" mode, and then starts another Python script called "alarm.py" using the os.startfile function. The "alarm.py" script is responsible for setting and triggering the alarm at the specified time. This function allows users to set alarms using their voice commands, making it a convenient and hands-free way to manage their daily schedule.

**Current Time Function:** This conditional statement is triggered when the user asks for the current time using their voice commands. The code retrieves the current time using the datetime module in Python and formats it into a string using the %I:%M %p format code, which represents the current hour in 12-hour format with leading zeros, followed by a colon and the current minute, and finally the AM or PM indicator

**Write a Note Function:** This conditional statement is triggered when the user says "write a note" using their voice commands. The code asks the user what they want to write using the speak function and retrieves the user's input using the takeCommand function. The code then creates a new file called "rohan.txt" using the open function with the 'w' mode, which allows the file to be written to.

The code then asks the user if they want to include the current date and time in the note using the speak function and retrieves the user's input using the takeCommand function. If the user says "yes" or "sure", the code retrieves the current time using the datetime module in Python and formats it into a string using the %I:%M %p format code, which represents the current hour in 12-hour format with leading zeros, followed by a colon and the current minute, and finally the AM or PM indicator.

The code then writes the current time, along with the user's note, to the file. If the user says "no" or does not include the current date and time in the note, the code simply writes the user's note to the file.

**Show Note Function:** This conditional statement is triggered when the user says "show note" using their voice commands. The code reads the contents of the "rohan.txt" file using the open function with the 'r' mode and prints the contents to the console using the print function. The code also speaks the first 6 characters of the note out loud using the speak function.

By allowing users to create and view notes using their voice commands, this code demonstrates how your personal voice assistant can be used to help users stay organized and keep track of important information.

**Calculate Function**: This conditional statement is triggered when the user says "calculate" using their voice commands. The code imports two functions from a Python module called "Calculatenumbers": WolfRamAlpha and Calc.

The code then removes the words "calculate" and "jarvis" (the name of the personal voice assistant) from the user's input using the replace function, so that only the mathematical expression remains in the query. The Calc function is then called with the modified query as its argument.

The Calc function uses the eval function in Python to evaluate the mathematical expression and returns the result. This function can handle simple arithmetic operations, such as addition, subtraction, multiplication, and division, as well as more complex expressions involving brackets and mathematical functions.

Alternatively, if the user's query involves more complex calculations or queries that cannot be handled by the Calc function, the WolfRamAlpha function can be used to query the Wolfram Alpha computational knowledge engine, which can provide more detailed and comprehensive answers to a wide range of mathematical and scientific questions.

By providing users with the ability to perform calculations using their voice commands, this code demonstrates how your personal voice assistant can be used to perform complex calculations quickly and easily, without the need for manual input.

**Chrome Short Function:** This function takes a query as its argument and performs different actions based on the keywords included in the query. The function uses the pyautogui library to simulate keyboard and mouse inputs.

The function includes conditional statements for several common actions that can be performed in the Chrome browser, including selecting all text on a page, copying and pasting text, cutting text, pausing and resuming videos, and navigating forward and backward through videos.

The function also includes a conditional statement for switching between different windows using the alt and tab keys.

By providing users with the ability to control the Chrome browser using their voice commands, this code demonstrates how your personal voice assistant can be used to perform a variety of tasks quickly and efficiently, without the need for manual input.

**dictapp Dictionary:** This is a Python function that can be used with a voice assistant program to open and close applications and websites. The function uses a dictionary called "dictapp" to map certain keywords to their respective applications. The "openappweb" function is used to launch either a website or a specified application by utilizing this dictionary. If the query contains a ".com", ".co.in", or ".org" domain, the function will assume it is a website and open it using the webbrowser module. Otherwise, it will check the query for any of the keywords in the dictionary and open the corresponding application using the "os" module. The "closeappweb" function is used to close either an application or a tab in Chrome. If the query contains the words "one tab," "two tabs," "three tabs," "four tabs," or "five tabs," the function will close the specified number of tabs in Chrome. If the query contains any of the keywords in the dictionary, the function will close the corresponding application using the "os" module.

The **searchYoutube function** checks if the user wants to search on YouTube by checking if the query contains the word "youtube." If so, it removes any unnecessary words from the query and opens a YouTube search page in the default web browser. It then plays the first YouTube search result using the pywhatkit library and speaks a confirmation message to the user.

The **searchWikipedia function** checks if the user wants to search on Wikipedia by checking if the query contains the word "wikipedia." If so, it removes any unnecessary words from the query, searches for the query on Wikipedia using the wikipedia library, and returns a summary of the first two sentences of the Wikipedia article. It speaks the summary to the user.

**Keyboard controller:**

These functions are used to control the system's volume by pressing the media volume up or down key multiple times in a loop. The keyboard module is used to simulate the key presses. volumeup() increases the volume by simulating the media volume up key press 5 times with a small delay, and volumedown() decreases the volume by simulating the media volume down key press 5 times with a small delay.

**Latest News:**

This is a Python function that uses the News API to get the latest news articles from various categories. It first defines a dictionary with different categories and their corresponding API URLs. It then prompts the user to choose a category and searches for a matching URL from the dictionary. If a URL is found, it sends a GET request to the API and retrieves the top headlines for that category. It then loops through the articles and reads the title of each article out loud to the user, prompting them to press a key to continue or stop. Finally, it speaks "that's all" to indicate that the function is done.

**Remember function**: if the user's query includes the phrase "what do you remember". If it does, it will open a file called "Remember.txt" in read mode and read its contents. It will then speak a phrase informing the user that it remembers what they told it to remember, followed by the contents of the file.

**Another features**

Play songs of our system

Open any application of our system

Changing background of screen and lock screen

Lock window

shutdown system

clear recycle bin

restart

log off

camera/ take a photo

you tube shortcuts ( pause, resume. Forward, backward )