A

Minor Project on

Desktop Voice Assistant

Submitted during fourth semester in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

in

(Information Technology)

by

ABHISHEK CHAUDHARY & KESHAV KUMAR
(19001011005) & (19001011031)

Under supervision of

Ms. AAYUSHI BANSAL

Department of Computer Engineering

YMCA UNIVERSITY OF SCIENCE & TECHNOLOGY
FARIDABAD-121006

April 2021

Table of Content

CANDIDATE’S DECLARATION

This work which is being presented in this minor project work titled “Desktop Voice Assistant” in fulfillment of the requirement for the degree of bachelor of Technology in Information Technology and submitted to “J. C. Bose University of Science and Technology, YMCA, Faridabad”, is an authentic record of my own work carried out under the supervision of Ms Aayushi Bansal.

The work contained in this thesis work has not been submitted to any other University or Institute for the award of any other degree or diploma by me.

Abhishek Chaudhary & Keshav Kumar
19001011005 & 19001011031

Acknowledgement

In completing this project report on project titled
Desktop voice Assistant
we had to take the help and guideline of a few respected people, who deserve our greatest gratitude. The completion of this project report gives us much Pleasure. We would like to show our gratitude to
Aayushi bansal mam
for giving us a good guideline for project throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this project report. Many people, especially our classmates and friends themselves, have made valuable comments and suggestions on this proposal which gave us inspiration to improve our project. Here we thank all the people for their help directly and indirectly to complete this project report.

CERTIFICATE

This is to certify that the work project titled “**desktop Voice Assistant**” submitted by Abhishek & Keshav to “**J. C. Bose University of Science and Technology, YMCA, Faridabad**” for the award of the degree of bachelor of technology in Information Technology is a record of bonafide work carried out by them under my supervision. In my opinion, the thesis has reached the standards of fulfilling the requirements of the regulations to the degree.

Aayush

Ms. Aayushi

(Mentor)

Assistant Professor,

Department of Computer Engg,

J. C. Bose University of Science and Technology, YMCA, Faridabad

Introduction

In today’s era almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your finger tips. These days we aren’t even using fingers. We just speak of the task and it is done. There exist systems where we can say Text Dad, “I’ll be late today.” And the text is sent. That is the task of a Virtual Assistant.

It also supports specialized task such as booking a flight, or finding cheapest book online from various e-commerce sites and then providing an interface to book an order are helping automate search, discovery and online order operations.

Virtual Assistants are software programs that help you ease your day to day tasks, such as showing weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple’s Siri, Amazon’s Alexa and Microsoft’s Cortana. This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2020.Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses. This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities

Purpose Scope and Objective

Purpose

Purpose of virtual assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistants enable users to speak natural language voice commands in order to operate the device and its apps. There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it’s clear that we are moving towards less screen interaction.

Scope

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it’s not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand. They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

Objectives

- It can send emails on your behalf.
- It can play music for you.
- It can do Wikipedia searches for you.
- It is capable of opening websites like Google, Youtube, etc., in a web browser.
- It is capable of opening your code editor or IDE with a single voice command.

Software Used

Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5 th code as compared to other OOPs languages. Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For JIA, libraries used are speech recognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc. Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

Basic Concept Used And Working

The working of Virtual Assistant uses following principle:

1. **Natural Language Processing.**

   To Understand user's speech input.

   Automatic Speech Recognition To understand command according to user's input.

2. **Artificial Intelligence.**

   To learn things from user and to store all information and behavior and relations of user**.**

3. **Inter Process Communication.**

   To get important information from other software applications.

Implementation

Defining Speak Function

The first and foremost thing for an Virtual Assistant is that it should be able to speak. To make our virtual assistant talk, we have made a function called ***speak()*** . This function will take audio as an argument, and then it will pronounce it.

```
def speak(audio):
    pass
```

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the speak() function we made.For this we'll use pyttsx3 module.

What is pyttsx3?

- A python library that will help us to convert text to speech. In short, it is a text-to-speech library.
- It works offline, and it is compatible with any version of python.

```
import pyttsx3
```

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

voices= engine.getProperty('voices') #getting details of current voice

engine.setProperty('voice', voice[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

Writing speak() Function :

We have written speak function to convert text to speech

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

Writing main() function:

We have written a main() function, and inside this main() function, we have called speak function.

```
if __name__=="__main__" :

speak("Hi how are you")
```

Whatever is written inside this speak() function will be converted into speech.

Defining Wish me Function :

Now, we have defined a wishme() function, that will make our virtual assistant wish or greet the user according to the time of computer or pc. To provide current or live time to assistant, we need to import a module called ***datetime****.*

```
def wishme():
hour = int(datetime.datetime.now().hour)
```

Here, we have stored the current hour or time integer value into a variable named hour. Now, we will use this hour value inside an if-else loop.

Defining Take command Function :

The next most important thing for our virtual 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 virtual assistant will return a string output by taking microphone input from the user.

For this we have used **SpeechRecognition** module

```
def takeCommand():
    #It takes microphone input from the user and returns string output

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

Also used try and except block to handle errors

```
    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in') #Using google for voice recognition.
        print(f"User said: {query}\n")  #User query will be printed.

    except Exception as e:
        # print(e)
        print("Say that again please...")   #Say that again will be printed in case of improper voice

        return "None" #None string will be returned
    return query
```

Now, we have developed logic for different commands such as Wikipedia searches, playing music, etc.

To search something on Wikipedia :

Use Wikipedia module into program.

```
if __name__ == "__main__":
    wishMe()
    while True:
    # if 1:
        query = takeCommand().lower() #Converting user query into lower case
    # Logic for executing tasks based on query
    if 'wikipedia' in query:  #if wikipedia found in the query then this block will be executed
        speak('Searching Wikipedia...')
        query = query.replace("wikipedia", "")
        results = wikipedia.summary(query, sentences=2)
        speak("According to Wikipedia")
        print(results)
        speak(results)
```

In the above code, we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the speak function's help.
Defining Task 2: To open YouTube site in a web-browser need to use **webbrowser** module.

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

Here, we are using the elif loop to check whether Youtube is in the user's query. Let' suppose the user gives a command as " open youtube." So, open youtube will be in the user's query, and the elif condition will be true.

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

Defining Task 3: To open Google site in a web-browser
We are opening Google in a web-browser by applying the same logic that we used to open youtube.

Defining Task 4: To play music using **OS** 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]))
```

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, we can play any song of our choice.However, we can also play a random song with the help of a random module. Every time we command to play music, assistant will play any random song from the song directory.

Defining Task 5: To know the current time

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

In the above, code with using **datetime()** function and storing the current or live system 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.

Defining Task 6: To open the VS Code Program

```
elif 'open code' in query:
        codePath = "C:\\Users\\Abhi\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
        os.startfile(codePath)
```

Recapitulate:

1. First of all, we have created a wishme() function that gives the greeting functionality according to our system time.
2. After wishme() function, we have created a takeCommand() function, which helps our virtual assistant to take command from the user. This function is also responsible for returning the user's query in a string format.
3. We developed the code logic for opening different websites like google, youtube, and many more.
4. At last, we added functionality to send emails.
5. Developed code logic for opening VS Code or any other application.

Conclusion

An excellent virtual assistant will save time and money by doing the small tasks for you and doing them accurately and with high quality. If you handle the virtual assistant correctly, it will be a boom in your business. If you'd like to find out more about hiring a virtual assistant, please consider someone from VP Virtual Assistants. You can learn more by filling out a simple online form.