A Project/Dissertation Review Report

on

VIRTUAL ASSISTANT:
A VOICE BASED SENDING EMAIL USING A.I

Submitted in partial fulfilment of the
requirement for the award of the degree of

B. Tech/ CSE

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Submitted To

Ms. Heena Kheera

Submitted By

1)ANKIT KUMAR   2) MD SHAYAN RAZA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
November, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"VIRTUAL ASSISTANT: A VOICE BASED SENDING EMAIL USING A.I"** in partial fulfilment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of Sep 2021 to Dec 2021, under the supervision of **MR. SHUBHAM KUMAR** Asst. Professor, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

1. ANKIT KUMAR (19SCSE1010278)
2. MD SHAYAN RAZA (19SCSE1010843)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

MR. SHUBHAM KUMAR
Asst. Professor,
GALGOTIAS UNIVERSITY

CERTIFICATE

The Final Project Viva-Voce examination of 1. ANKIT KUMAR (19SCSE1010278) 2. MD SHAYAN RAZA (19SCSE1010843) has been held on _________________ and their work is recommended for the award of Bachelor of Technology in the School of Computing Science and Engineering of Galgotias University, Greater Noida

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Abstract

The advancement in technology over time has been unmeasurable. From the first digital computer built by Eniac having a clock speed of 100KHz to Summit developed by the US Department of Energy has a performance of 148.6 Peta Flops, we have come a long way in technological advancement. In such an era of advancement if people are still struggling to interact with their machine using various input devices, then it's not worth it. For this reason, many voice assistants were developed and are still being improved for better performance and efficiency. The main task of a voice assistant is to minimize the use of input devices like keyboard, mouse, touch pens, etc. This will reduce both the hardware cost and space taken by it. The Most famous application is of iPhone "SIRI" which helps the end user to communicate end user mobile with voice and it also responds to the voice commands of the user. Same kind of application is also developed by the Google that is "Google Voice Search " which is used for in Android Phones. It is named as Personal Assistant with Voice Recognition Intelligence, which takes the user input inform of voice or text and process it and returns the output in various forms like action to be performed or the search result is dictated to the end user. In addition, this proposed system can change the way of interactions between end user and the mobile devices.

Keywords:

Desktop Assistant, Python, Machine Learning, Text to Speech, Speech to Text, Language Processing, Voice Recognition, Artificial Intelligence, Internet of Things (IOT), Virtual Assistant

TABLE OF STUDENT DATA:

S No	**NAME OF THE STUDENT**	**ADMISSION NO**
1.	ANKIT KUMAR	19SCSE1010278
2.	MD SHAYAN RAZA	19SCSE1010843

TABLE OF FACULTY DATA:

S No	**NAME OF FACULTY**	
1.	MR. SHUBHAM KUMAR	GUIDE
2.	MS. HEENA KHERA	REVIEWER

TABLE OF CONTENT

CHAPTER-1
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 fingertips. 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 ecommerce 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. For my project the wake word is JOKER. We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and Microsoft's Cortana. For this project, wake word was chosen JOKER.

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. JOKER is effortless to use. Call the wake word 'JOKER' followed by the command. And within seconds, it gets executed. Voice searches have dominated over text search. 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 2022.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.

CHAPTER-2

Literature Survey

Problem Statement

Artificial Intelligence personal assistants have become plentiful over the last few years. Applications such as Siri, Bixby, Ok Google and Cortana make mobile device users' daily routines that much easier. We may be asking ourself how these functions. Well, the assistants receive external data (such as movement, voice, light, GPS readings, visually defined markers, etc.) via the hardware's sensors for further processing - and take it from there to function accordingly. Not too long ago, building an AI assistant was a small component of developers' capacities; however, nowadays, it is quite a realistic objective even for novice programmers. To create a simple personal AI assistant, one simply needs dedicated software and around an hour of working time. It would take much more time, though, to create something more advanced and conceptually innovative. Nonetheless, well thought-out concepts can result in a great base for a profitable startup. We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms. But to our surprise, there's no such complete virtual assistant available for Core Windows platform consisting of 70% of the users. So, this is actually a major problem for users where there could be internet instability, server problems and places where internet is not accessible.

EXISTING SYSTEM

The current voice assistant system basically existing on Windows OS is the Cortana which is completely online based system and requires high speed fast internet and also a regular Microsoft account for login and other existing system is Ok-Google voice assistant which is browser dependent.

PROPOSED SYSTEM

The work is initialized with analyzing the audio commands given by the user via microphone. This can be anything like retrieving any information, operating computer's files, etc. Tests are conducted by programming according to books and online resources, with the goal to find best practices and a more advanced understanding of Voice Assistant. Fig.1 shows the detailed workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relatable script for execution. But the difficulties don't end there. Even with tons of hours of input, other factors aside can play a big role in whether or not the software can understand you basically. Background noise can easily eliminate a speech recognition device off the track. This is because it does not inherently have the ability to classify the ambient sounds it "hears" of a dog barking or a helicopter flying overhead, from your voice. Developers have to program that ability into the machine; they conduct data collection of these ambient sounds and "tell" the device to filter them out accordingly. Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes in most of the conditions.

CHAPTER-3

Project Design

3.1 Data Flow Diagram:

A data flow diagram shows **the way information flows through a process or system**. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. ... Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text.

a) DFD Level 0

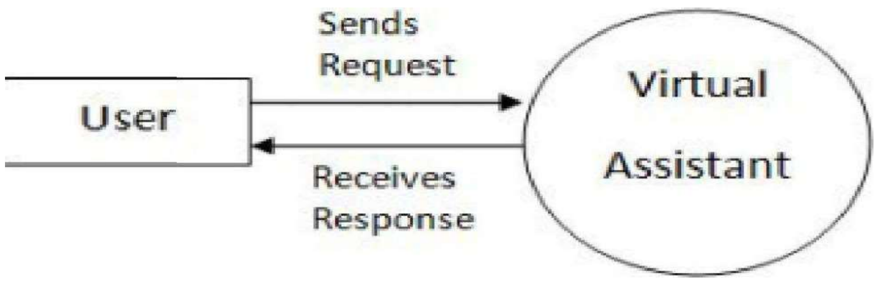


Fig: Data Flow Diagram L0

b) DFD Level 1

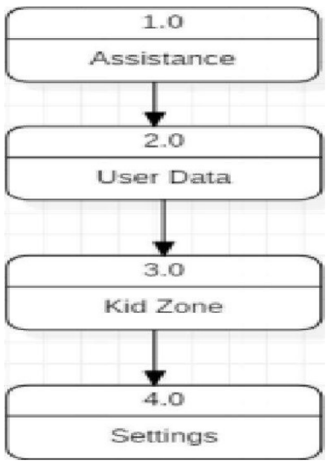


Fig: Data Flow Diagram L

3.2 ER Diagram

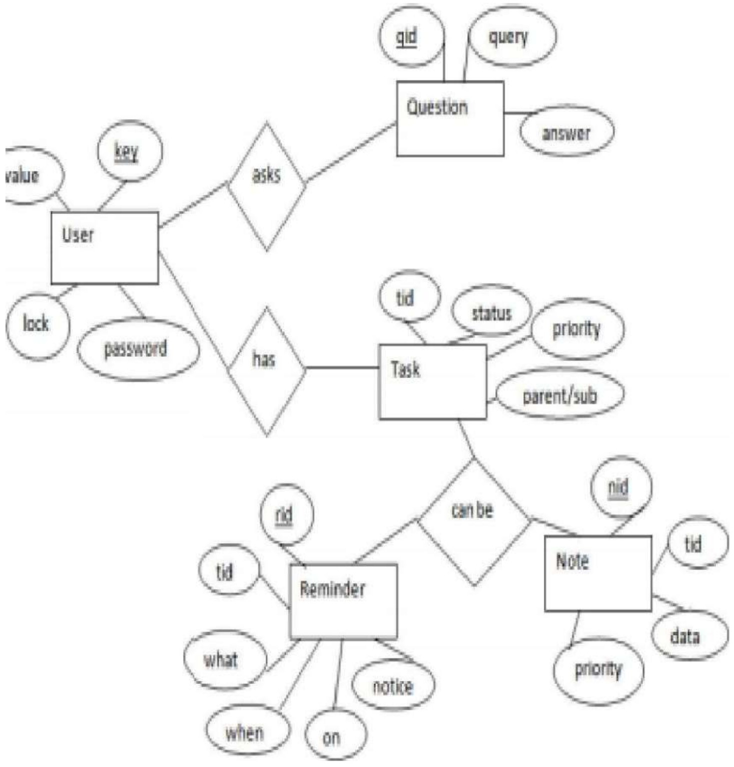


Figure 3.2 ER Diagram

The above diagram shows entities and their relationship for a virtual assistant system. We have a user of a system who can have their keys and values. It can be used to store any information about the user. Say, for key "name" value can be "User1". For some key's user might like to keep secure. There he can enable lock and set a password (voice clip).
Single user can ask multiple questions. Each question will be given ID to get recognized along with the query and its corresponding answer. User can also be having n number of tasks. These should have their own unique id and status i.e. their current state. A task should also have a priority value and its category whether it is a parent task or child task of an older task.

3.3 Use Case Diagram

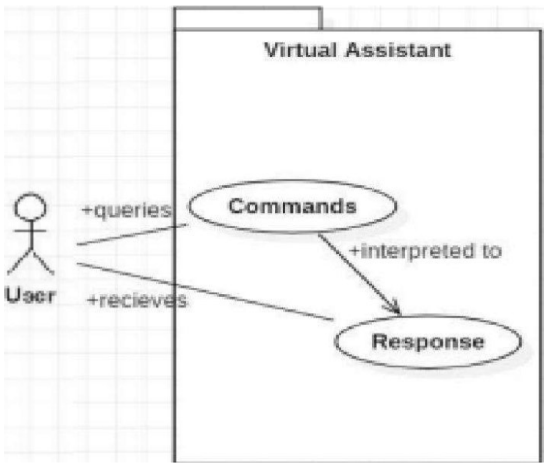


Fig: Use Case Diagram

In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

CHAPTER 4

Modules Description

Is it an A.I.?

Artificial intelligence (AI) is the ability of a computer or a robot controlled by a computer to do tasks that are usually done by humans because they require human intelligence and discernment.

Many people will argue that the virtual assistant that we have created is not an A.I, but it is the output of a bunch of the statement. But, if we look at the fundamental level, the sole purpose of A.I develop machines that can perform human tasks with the same effectiveness or even more effectively than humans.

It is a fact that our virtual assistant is not a very good example of A.I., but it is an A.I.!

4.1 Survey of Technology

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 1/5th 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 (Al), Machine Learning (ML), natural language processing, Data science etc. Python has a lot of libraries for every need of this project. For JOKER, libraries used are speech recognition to recognize voice, Pyttsx3 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.

Quepy
Quepy is a python framework to transform natural language questions to queries in a database query language. It can be easily customized to different kinds of questions in natural language and database queries. So, with little coding you can build your own system for natural language access to your database.

SQLite
SQLite is a capable library, providing an in-process relational database for efficient storage of small-to-medium sized data sets. It supports most of the common features of SQL (Structured Query Language) with few exceptions. Best of all, most Python users do not need to install anything to get started working with SQLite, as the standard library in most distribution ships with the sqlite3 module.

SQLite runs embedded in memory alongside your application, allowing you to easily extend SQLite with your own Python code. SQLite provides quite a few hooks, a reasonable subset of which are implemented by the standard library database driver

Pyttsx3

Pyttsx3 stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

Installation:

```
pip install pyttsx3
```

In case of errors:

- No module named win32com.client
- No module named win32
- No module named win32api

Then, install pypiwin32 by typing the below command in the terminal:

```
pip install pypiwin32.
```

After successfully installing pyttsx3, import this module into your program.

Speech Recognition
This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

sapi5

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

VoiceId

- Voice id helps us to select different voices.
- voice[0].id = Male voice
- voice[1].id = Female voice

speak() Function :

We made a function called speak() at the starting of this tutorial. Now, we will write our speak() function to convert our text to speech.

```
def speak(audio):
engine.say(audio)
engine.runAndWait() #Without this command, speech will not be audible
to us.
```

4.2 DEFINING FUNCTIONS:

Our main() function:

We will create a main() function, and inside this main() Function, we will call our speak function.

```
if __name__=="__main__" :
speak("Welcome professor")
```

Whatever we will write inside this speak() function will be converted into speech.

Defining Wish me Function:

Now, we will make a **wishme()** function that will make our J.O.K.E.R. wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program by:

```
import datetime
```

Now, let's start defining the **wishme()** function:

```
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 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.

Before defining the takeCommand() function, we need to install a module called **speechRecognition.** Install this module by:

```
pip install speechRecognition
```

After successfully installing this module, import this module into the program by writing an import statement.

```
import speechRecognition as sr
```

Let's start coding the takeCommand() function :

```
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)
```

We have successfully created our takeCommand() function. Now we are going to add a try and except block to our program to handle errors effectively.

```
  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
```

Defining Task 1: To search something on Wikipedia

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module :

```
pip install wikipedia
```

After successfully installing the Wikipedia module, import it into the program by writing an import statement.

```
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 user's search query 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

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.

Code:

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

Here, we are using an elif loop to check whether YouTube is in the user's query. Let' suppose the user gives a command as "J.O.K.E.R., open YouTube." So, open YouTube will be in the user's query, and the elif condition will be true.

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

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

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

To play music, we need to import a module called os. Import this module directly with an import statement.

```
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, 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. Every time you command to play music, J.O.K.E.R 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 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.

Defining Task 7: To send Email

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.

Defining Send email function :

We will create a **sendEmail()** function, which will help us send emails to one or more than one recipient.

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

In the above code, we are using the SMTP module, which we have already discussed above.

Note: Do not forget to *'enable the less secure apps'* feature in your Gmail account. Otherwise, the sendEmail function will not work properly.

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

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

We are using the try and except block to handle any possible error while sending emails.

Recapitulate

1. First of all, we have created a **wishme()** function that gives the greeting functionality according to our A.I system time.
2. After wishme() function, we have created a **takeCommand()** function, which helps our A.I 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 stack overflow.
4. Developed code logic for opening application.
5. At last, we added functionality to send emails.

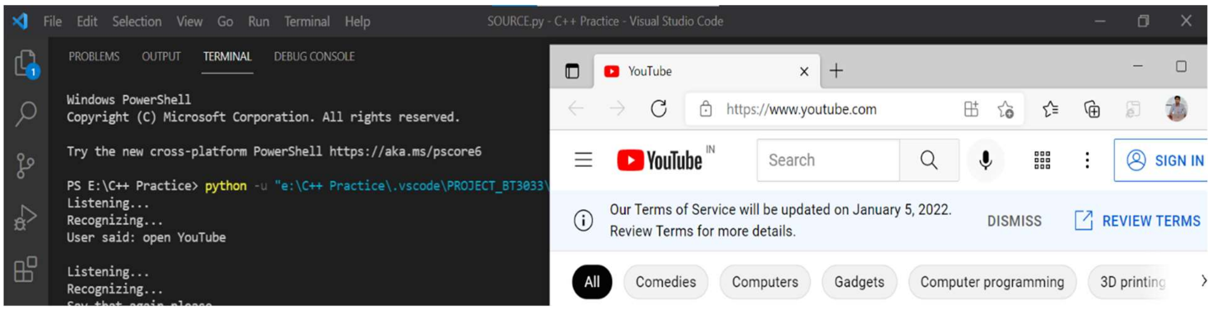
4.3 Source code:

```python
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import wikipedia #pip install wikipedia
import webbrowser
import os
import smtplib

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[1].id)


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


def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")

    elif hour>=12 and hour<18:
        speak("Good Afternoon!")

    else:
        speak("Good Evening!")

    speak("I am Joker Sir Please tell me how may I help you")

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)

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

    except Exception as e:
        # print(e)
        print("Say that again please...")
        return "None"
    return query

def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('project.bt3033@gmail.com','Project@3033')
    server.sendmail('project.bt3033@gmail.com', to, content)
    server.close()

if __name__ == "__main__":
    wishMe()
    while True:
    # if 1:
        query = takeCommand().lower()

        # Logic for executing tasks based on query
        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)

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

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

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


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

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

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

        elif 'send email' in query:
            try:
                speak("What should I say?")
                content = takeCommand()
                to = "project.bt3033@gmail.com"
                sendEmail(to, content)
                speak("Email has been sent!")
            except Exception as e:
                print(e)
                speak("Sorry my friend joker. I am not able to send this
email")
```

5.Result

Glimpse of User Interface

5.1 Opening Youtube

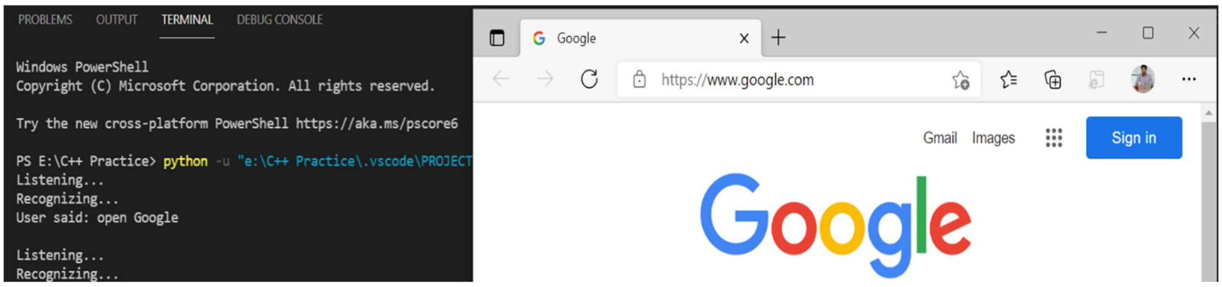


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\C++ Practice> python -u "e:\C++ Practice\.vscode\PROJECT_BT3033\
Listening...
Recognizing...
User said: open YouTube

Listening...
Recognizing...
Say that again please
```

File Edit Selection View Go Run Terminal Help
SOURCE.py - C++ Practice - Visual Studio Code
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
YouTube
https://www.youtube.com
Search
SIGN IN
DISMISS
REVIEW TERMS
All Comedies Computers Gadgets Computer programming 3D printing

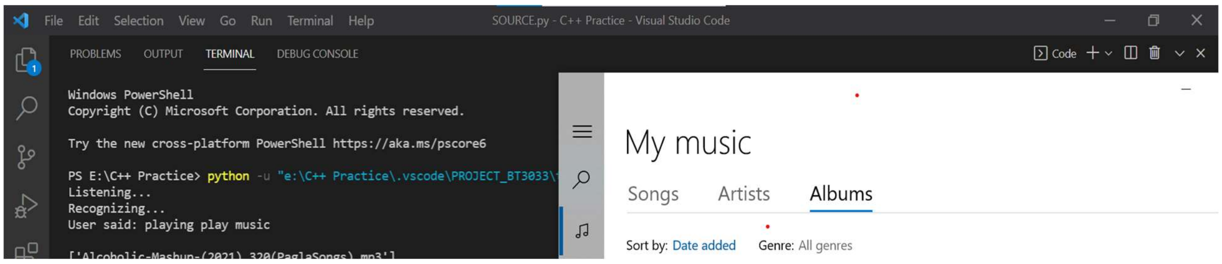
Our Terms of Service will be updated on January 5, 2022.
Review Terms for more details.

5.2 Opening Google



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\C++ Practice> python -u "e:\C++ Practice\.vscode\PROJECT
Listening...
Recognizing...
User said: open Google

Listening...
Recognizing...
```

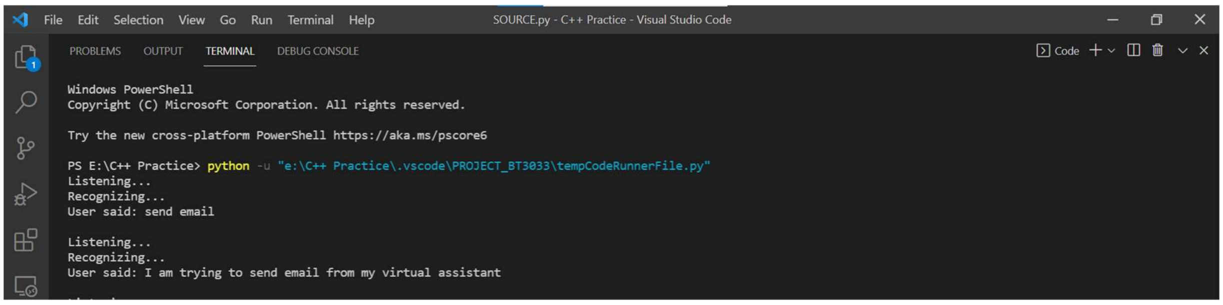
5.3 Playing Music



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\C++ Practice> python -u "e:\C++ Practice\.vscode\PROJECT_BT3033\
Listening...
Recognizing...
User said: playing play music

['Alcoholic-Mashup-(2021) 320(PaglaSongs).mp3']
```

File Edit Selection View Go Run Terminal Help
SOURCE.py - C++ Practice - Visual Studio Code
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Code
My music
Songs Artists Albums
Sort by: Date added
Genre: All genres

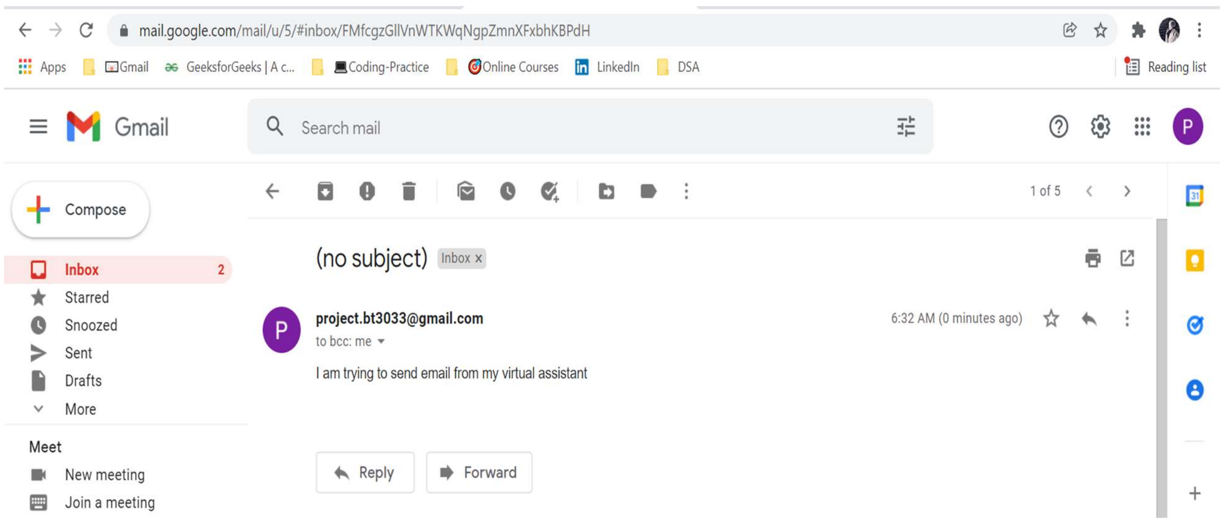
5.4 Sending email



File Edit Selection View Go Run Terminal Help
SOURCE.py - C++ Practice - Visual Studio Code
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\C++ Practice> python -u "e:\C++ Practice\.vscode\PROJECT_BT3033\tempCodeRunnerFile.py"
Listening...
Recognizing...
User said: send email

Listening...
Recognizing...
User said: I am trying to send email from my virtual assistant
```

Received email:



mail.google.com/mail/u/5/#inbox/FMfcgzGllVnWTKWqNgpZmnXFxbhKBPdH
Apps Gmail GeeksforGeeks | A c... Coding-Practice Online Courses LinkedIn DSA Reading list
Gmail
Search mail
Compose
Inbox 2
Starred
Snoozed
Sent
Drafts
More
Meet
New meeting
Join a meeting
1 of 5
Reply
Forward

(no subject) Inbox x

project.bt3033@gmail.com
to bcc: me
6:32 AM (0 minutes ago)

I am trying to send email from my virtual assistant

Future Scope

When voice technology began to emerge in 2011 with the introduction of Siri, no one could have predicted that this novelty would become a driver for tech innovation. Now, a decade later, it's estimated that every 1 in 4 U.S. adults own a smart speaker (i.e., Google Home, Amazon Echo) and eMarketer forecasts that nearly 92.3 percent smartphone users will be using voice assistants by 2023.

Brands such as Amazon, and Google are continuing to fuel this trend as they compete for market share. Voice interfaces are advancing at an exponential rate in all industries, with notable growth in healthcare to banking, as companies are racing to release their own voice technology integrations to keep pace with consumer demand.

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 has to become better at handling challenges such as accents and background noise.

CONCLUSION

Voice-Controlled Devices uses Natural Language Processing to process the language spoken by the human and understand the query and process the query and respond to the human with the result. The understanding of the device means Artificial Intelligence needs to be integrated with the device so that the device can work in a smart way and can also control IoT applications and devices and can also respond to query which will search the web for results and process it. It is designed to minimize the human efforts and control the device with just human Voice. The device can also be designed to interact with other intelligent voice-controlled devices like IoT applications and devices, weather reports of a city from the Internet, send an email to a client, add events on the calendar, etc. The accuracy of the devices can be increased using machine learning and categorizing the queries in particular result sets and using them in further queries. The accuracy of the devices is increasing exponentially in the last decade. The devices can also be designed to accept commands in bilingual language and respond back in the same language queried by the user. The device can also be designed to help visually impaired people.

The voice assistant’s technology is more than affordable and offers many benefits to its users. Having a personal assistant with access to the unlimited knowledge stored on the internet - isn't this what mankind dreams about?

There can be no doubt that voice assistants are, and will continue to become, a great feat of human ingenuity and they are already creeping into our lives in some shape or form. With the eventual roll-out of 5G and the improvement in machine learning voice assistants may be setting themselves up to be a tool we cannot live without.

However, before we get to that stage, there are hurdles to cross which include heavy investment, improvement in the technology.

References

1. https://www.wikipedia.org/
2. https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyaudio
3. https://docs.python.org/3/library/smtplib.html
4. https://pypi.org/project/pyttsx3/
5. https://docs.python.org/3/library/webbrowser.html
6. https://www.w3schools.com/python/python_datetime.asp
7. https://www.youtube.com/results?search_query=code+with+harry+