

MINI PROJECT REPORT

On

VOICE ASSISTANT

Submitted by-

Akshat Dwivedi
181500062
Prakhar Srivastava
181500471
Simran Gupta
181500713
Sristi Shukla
181500727

Department of Computer Engineering & Applications
Institute of Engineering & Technology



GLA University
Mathura- 281406, INDIA
2020

CERTIFICATES OF ALL MEMBERS IN THE GROUP

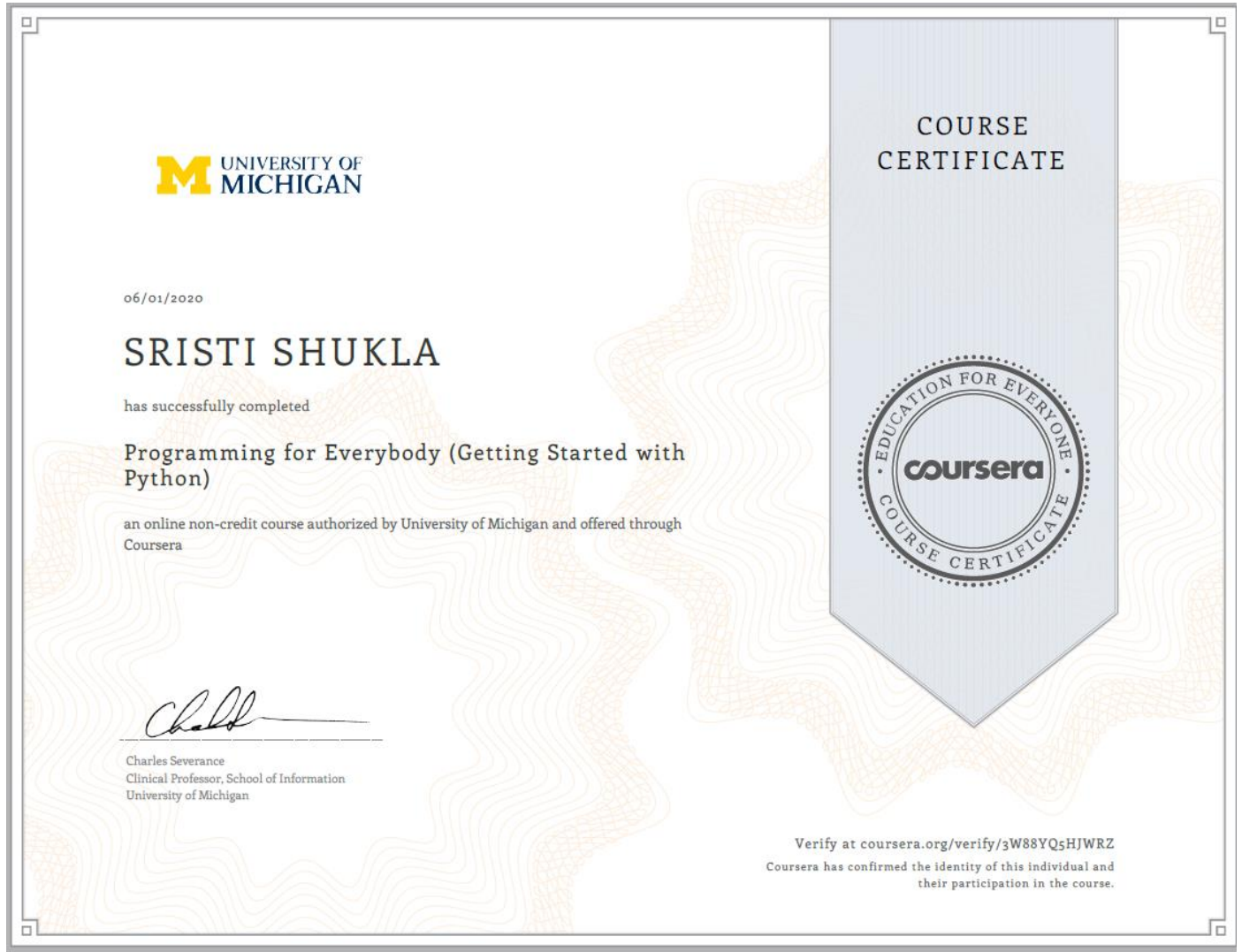
1. AKSHAT DWIVEDI



2. PRAKHAR SRIVASTAVA



3.SRISTI SHUKLA



4.SIMRAN GUPTA



Summary of the Project Work

In this project I have learned to work on virtual assistant to query voice commands of a program user so to do this I have learnt the interrelation between using python programming. I developed a project on Voice Assistant for a general store. And have learnt many uses of python in our daily life for making our life simpler and managing our data.

The project entitled Voice Assistant was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The entire code is error free. Also the project helped us understanding about the development phases of a project and software development life cycle. I learned how to test different features of a project. There is a scope for further development in our project to a great extent. A number of features can be added to this project in future.

ACKNOWLEDGEMENT

The project work in this report is an outcome of continuous work over a period and drew intellectual support from various sources. I would like to articulate our profound gratitude and to all those people who extended their wholehearted co-operation and have helped us in completing this project successfully. I am thankful to my teammates for helping and assisting me in making the project successful. We would also like to thank our parents & other fellow mates for guiding and encouraging me throughout the duration of the project.



Department of Computer Engineering and Applications

GLA University, Mathura

17 km Stone NH#2, Mathura-Delhi Road, P.O. Chaumuha

Mathura, 281406

DECLARATION

I hereby declare that the project work entitled “[Voice Assistant](#)” submitted to the GLA University Mathura, is a record of an original work done by me under the guidance of [Mr. Amir khan](#).

Signature of Candidate:

Name of team members: Akshat Dwivedi , Prakhar Srivastava , Simran Gupta , Srishti Shukla

Roll No: 181500062 , 181500471 , 181500713 , 181500727

Course: Computer Science and Engineering

Year: III

Semester: V



ABSTRACT

To call any technology that makes our lives easier by one name is almost impossible. There are a variety of terms that refer to agents that can perform tasks or services for an individual, and they are almost interchangeable — but not quite. They differ mainly based on how we interact with the technology, the app, or a combination of both.

Here are some basic definitions, similarities, and differences:

Intelligent Personal Assistant: This is software that can assist people with basic tasks, usually using natural language. Intelligent personal assistants can go online and search for an answer to a user's question. Either text or voice can trigger an action. **Automated Personal Assistant:** This term is synonymous with intelligent personal assistant. **Smart Assistant:** This term usually refers to the types of physical items that can provide various services by using smart speakers that listen for a wake word to become active and perform certain tasks. Amazon's Echo, Google's Home, and Apple's Home Pod are types of smart assistants. **Virtual Digital Assistants:** These are automated software applications or platforms that assist the user by understanding natural language in either written or spoken form. Technology is constantly advancing and changing, and the voice assistant market will progress along with it. In April 2015, the research firm Gartner predicted that by the end of 2018, 30 percent of interactions with technology would be through "conversations" with smart machines, many of them by voice. Tractica is a market intelligence firm that focuses on human interaction with technology. Their reports say unique consumer users for virtual digital assistants (which they define as automated software applications or platforms that assist the human user through understanding natural language in written or spoken form) will grow from more than 390 million worldwide users in 2015 to 1.8 billion by the end of 2021. The growth in the business world is expected to increase from 155 million users in 2015 to 843 million by 2021. With that kind of projected growth, revenue is forecasted to grow from \$1.6 billion in 2015 to \$15.8 billion in 2021. According to Global Market Insights, Inc., between 2016 and 2024, the market share for the technology will grow at an annual rate of almost 35 percent. More and more sectors of the economy, like healthcare and the automotive industry, are finding uses for the speech recognition technology in addition to those found in devices like smart speakers and phones.

Table of Content

1. About project.....	9
○ Introduction	
○ Benefits / Drawbacks	
○ Application area	
○ Definition of Project	
○ Objective	
○ Project Scope	
○ Software Required	
3. Introduction to python.....	13
• Modules used	
• About python 3.9	
4. Project Implementation.....	15
6. References.....	25

Introduction:

Speech is the most basic means of adult human communication. The basic goal of speech processing is to provide an interaction between a human and a machine. Speech recognition allows the machine to catch the words, phrases and sentences. Natural language processing to allow the machine to understand what we speak, and speech synthesis to allow the machine to speak. An intelligent virtual assistant (IVA) or intelligent personal assistant (IPA) is a software agent that can perform tasks or services for an individual based on commands or questions. Sometimes the term "Chabot" is used to refer to virtual assistants generally or specifically accessed by online chat. In some cases, online chat programs are exclusively for entertainment purposes. Some virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal (spoken?) commands.^[1]As of 2017, the capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market and a strong emphasis on both email and voice user interfaces. Apple and Google have large installed bases of users on smartphones. Microsoft has a large installed base of Windows-based personal computers, smartphones and smart speakers. Amazon has a large install base for smart speakers.^[2]Conversica has over 100 million engagements via its email and sms interface Intelligent Virtual Assistants for business. In this project we make use of the above feature to create an Automatic Speech Recognition System to make our systems make use of our voice commands to perform real life tasks. For example sending an E-mail, telling What time is it, Searching for something on Wikipedia and opening files present on our Systems etc. We were motivated to create this project through inbuilt AI like Microsoft Cortana, Google Assistant and we are quite interested in creating our own AI Voice Recognition System using Python as our primary language. We were also intrigued to work on this project because of its many future prospects and applications.

1.2 Benefits / Drawbacks

● Benefits

In-car systems

- Simple voice commands may be used to initiate phone calls, select radio stations or play music from a compatible smartphone, MP3 player or music-loaded flash drive.

Military

- In these programs, speech recognizers have been operated successfully in fighter aircraft, with applications including: setting radio frequencies, commanding an autopilot system, setting steer-point coordinates and weapons release parameters, and controlling flight display.

Usage in education and daily life User:

- Speech recognition can allow students with learning disabilities to become better writers. By saying the words aloud, they can increase the fluidity of their writing, and be alleviated of concerns regarding spelling, punctuation, and other mechanics of writing.

Keep your home safe

- You can connect your virtual assistant to your smart home security devices to make it easier to keep your property safe. For example, you can connect the Blink security camera system to any Amazon Alexa device and once the Blink 'skill' is enabled on your Alexa app you can give voice commands such as 'Alexa, ask Blink to arm my home security system'.

Save money on your energy bills

- Virtual assistants can control smart home devices, including next-gen lighting options. You could hook up a smart thermostat such as the Nest to Google Home or Alexa and tell your VPA when you're leaving the house, so that your heating gets turned off. Tests suggest that UK users can reduce their energy usage by as much 16.5 per cent, saving cash in the long run.

What's Causing the Shift Towards Voice?

The main driver of the shift towards voice user interfaces is the changing user demands. 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. The mass adoption of artificial intelligence in users' everyday lives is also fueling the shift towards voice applications. The number of IoT devices such as smart thermostats, appliances, and speakers are giving voice assistants more utility in a connected user's life. Smart speakers are the number one way we are seeing voice being used, however, it only starts there. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years. Applications of this technology are seen everywhere, so where will it take us in 2020 and beyond? We provide a high-level overview of the potential that voice has and 7 key predictions we think will take off in the coming years.

Application Area

1.Streamlined Conversations

Both Google and Amazon recently announced that both assistants will no longer require the use of repeated "wake" words. Previously both assistants were dependent on a wake word (Alexa or Ok, Google) to initiate a new line of conversation. For example, one would have to ask "Alexa, what's the current temperature at the hallway thermostat?" and then have to say, "Alexa" again before requesting that the voice assistant to "set the hallway thermostat to 23 degrees." It would be more convenient and natural for the user to say, "Alexa, what's the current temperature at the hallway thermostat?" and then simply say "set my hallway thermostat to 23 degrees," without requiring the wake word again, and now that's possible. Consumers use voice assistants in specific locations, usually while multitasking, and can either be alone or amongst a group of people when using them. Having devices that can decipher these contextual factors make a conversation more convenient and efficient with these devices, but it also shows that developers behind the technology are aiming to provide a more user-centric experience.

2.Compatibility and Integration

When it comes to integrating voice technology with other products, Amazon has been ahead of the game. Those who use Alexa will be familiar with the fact that the voice assistant is already integrated into a vast array of products including Samsung's Family Hub refrigerators. Google has finally caught on and has announced Google Assistant Connect. The idea behind this technology is for manufacturers to create custom devices that serve specific functions and are integrated with the Assistant. In 2020, we will see a greater interest in the development of voice-enabled devices. This will include an increase in mid-level devices: devices that have some assistant functionality but aren't full-blown smart speakers. Instead, they communicate with your smart speaker, display or even perhaps your phone over Bluetooth where the processing happens on those devices. Amazon is already well on its way with an Alexa-enabled wall clock.

3.Search Behaviors Will Change

Voice search has been a hot topic of discussion. Visibility of voice will undoubtedly be a challenge. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface unless it is connected to the Alexa or Google Assistant app. Search behaviors, in turn, will see a big change. In fact, if tech research firm Juniper Research is correct, voice-based ad revenue could reach \$19 billion by 2022, thanks in large part to the growth of voice search apps on mobile devices. Brands are now experiencing a shift in which touchpoints are transforming to listening points, and organic search will be the main way in which brands have visibility. comScore data even reveals that 50% of all search will be via voice tech by 2020. As voice search grows in popularity, advertising agencies and marketers expect Google and Amazon will open their platforms to additional forms of paid messages.

4.Individualized Experiences

Voice assistants will also continue to offer more individualized experiences as they get better at differentiating between voices. Google Home is able to support up to six user accounts and detect unique voices, which allows Google Home users to customize many features. Users can ask "What's on my calendar today?" or "tell me about my day?" and the assistant will dictate commute times, weather, and news information for individual users.

It also includes features such as nicknames, work locations, payment information, and linked accounts such as Google Play, Spotify, and Netflix. Similarly, for those using Alexa, simply saying “learn my voice” will allow users to create separate voice profiles so the technology can detect who is speaking for more individualized experiences.

5.Voice Push Notifications

We’ve previously discussed the method of using user-centric push notifications as a means to re-engage users with your app, voice technology presents a unique means of distributing push notifications. As a way to increase user engagement and retention, push notifications simply remind users of the app and display relevant messaging to the user. Now that both Google Assistant and Amazon’s Alexa allow the user to enable spoken notifications for any third-party app that has the compatibility, users can hear notifications rather than read them. These notifications are generally related to calendar appointments or new content from core features.

6.Touch Interaction

CES 2019 continued to prove that voice and visual displays are merging into one seamless experience. This year Google showcased what is being called the E Ink screen. This display can show the weather, local traffic information, or calendar events. The push to bring visual and voice capabilities together allow users to further interact with the assistant.

7.Security Will Be a Focus

Forty-one percent of voice assistant users are concerned about trust and privacy according to a report from Microsoft. With news from Google I/O and Amazon’s re:MARS conferences announcing that assistants will essentially be able to plan an entire evening, for example, find local movie times, buy tickets, book a restaurant reservation and schedule an Uber, concerns regarding payments and sensitive information are valid. Voice payments, in particular, will become more secure and convenient for users to make purchases. Speaker verification and ID will also become paramount as part of the voice assistant experience with more security being built around the user.

Why Adopt A Mobile Voice Strategy?

Mobile phones are already personalized, more so than any website. Additionally, there is very little screen space on mobile, making it more difficult for users to search, or navigate. With larger product directories and more information, voice applications enable consumers to use natural language to eliminate or reduce manual effort, making it a lot faster to accomplish tasks. Rogers has introduced voice commands to their remotes allowing customers to quickly browse and find their favorite shows or the latest movies with certain keywords, for example, an actor's name. Brands need to focus on better mobile experiences for their consumers and voice is the way to do so. Users are searching for quicker and more efficient ways of accomplishing tasks and voice is quickly becoming the ideal channel for this. Whether that's finding out information, making a purchase, or achieving a task, voice is the new mobile experience. It's clear that brands are racing to figure out their voice strategy. With over 100 million Alexa devices being sold alone, there's a reason why businesses are looking to catch up.

8.Voice User Interface (VUI) Will Continue to Advance

Even with just that handful of simple scenarios, it's easy to see why voice assistants are shaping up to become the hubs of our connected homes and increasingly connected lives. Voice technology is becoming increasingly accessible to developers. For example, Amazon offers Transcribe, an automatic speech recognition (ASR) service that enables developers to add speech-to-text capability to their applications. Once the voice capability is integrated into the application, users can analyze audio files and in return, receive a text file of the transcribed speech. Google has made moves in making Assistant more ubiquitous by opening the software development kit through Actions, which allows developers to build voice into their own products that support artificial intelligence. Another one of Google's speech-recognition products is the AI-driven Cloud Speech-to-Text tool which enables developers to convert audio to text through deep learning neural network algorithms. This is only the beginning of voice technology as we will see major advancements in the user interface in the years to come. With the advancements in VUI, companies need to start educating themselves on how they can best leverage voice to better interact with their customers.

Department of CEA,
GLAU Mathura

How are you solving their pain points with voice? Will voice enhance the user experience or frustrate the user. In 2020, voice-enabled apps will not only accurately understand what we are saying, but how we are saying it and the context in which the inquiry is made. However, there are still a number of barriers that need to be overcome before voice applications will see mass adoption. 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. 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.

Voice Is the Future of Brand Interaction and Customer Experience

Advancements in a number of industries are helping digital voice assistants become more sophisticated and useful for everyday use. Voice has now established itself as the ultimate mobile experience. A lack of skills and knowledge make it particularly hard for companies to adopt a voice strategy. There is a lot of opportunity for much deeper and much more conversational experiences with customers.

About Python 3.9

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is named after a TV Show called "Monty Python's Flying Circus" and not after Python-the snake. Python 3.0 was released in 2008. Although this version is supposed to be backward

Why to Learn Python 3?

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is a **MUST** for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python:

Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Characteristics of Python

Following are important characteristics of python –

It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

4. Project Implementation

Implementation Details:

Pytttsx3 module:

pytttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

Speech recognition module:

Library for performing speech recognition, with support for several engines and APIs, online and offline.

OS module:

The OS module in python provides functions for interacting with the operating system. OS, comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The `*os*` and `*os.path*` modules include many functions to interact with the file system.

Smtplib module:

This script simply checks whether a website is up or not. If it is up then it will send an email about this, if it is down then it will keep checking and when the site will be up, it will send an email and terminate.

Datetime module:

a module named `datetime` can be imported to work with the date as well as time. 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.

Keyboard module :

Python provides a library named `keyboard` which is used to get full control of the keyboard. It's a small Python library which can hook global events, register hotkeys, simulate key presses and much more.

Wikipedia module:

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.

In this article, we will see how to use Python's Wikipedia module to fetch a variety of information from the Wikipedia website

```
1 import pytttsx3
2 import datetime
3 import webbrowser
4 import smtplib
5 import keyboard
6 import os
7 import wikipedia
8 import speech_recognition as sr
```

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

Speak function take input as audio in audio format from microphone

```
1 def wishMe():
2     hour = int(datetime.datetime.now().hour)
3     if hour>=0 and hour<12:
4         speak("Good Morning!!!")
5
6     elif hour>=12 and hour<17:
7         speak("Good Afternoon!!!")
8
9     else:
10        speak("Good Evening!!!!")
11
12    speak("I am Rex Sir. Please tell me how may I help you")
```

Wishme function will automatically called on the run time and return output as audio
Automatically wishes condition based on global time


```
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")
        speak(query)

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

Take command take audio or command given to compiler stored in a variable query and Return it and that particular variable is further used in upcoming condition to perform Some task depend on condition user want
And also try exception rule is used whether speech is recognized accurately otherwise it Will show message to say again to user to give again clear voice input

```
1 def sendEmail(to, content):
2     server = smtplib.SMTP('smtp.gmail.com', 587)
3     server.ehlo()
4     server.starttls()
5     server.login('ur gmail ', 'password')
6     server.sendmail('ur gmail', to, content)
7     server.close()
8
```

This send Email function is main definition of program to send a email to which person And what content user want to send by just calling by name and content of email This function must be hided from other because it content user privacy email and Password through which user send his mail from mentioned email address

```

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

```

AS our voice recognition system open with wish by calling function wishme()
 And above we explain take command return a voice into a string format stored in query Variable

```

1-      if 'wikipedia for' in query:
2-          speak('Searching Wikipedia...')
3-          query = query.split(" ")
4-          location = query[2:]
5-          location=' '.join(location)
6-          webbrowser.open('https://en.wikipedia.org/wiki/'+ location)
7-          text = wikipedia.summary(query)
8-          print(text)
9-          speak(text)
10-
11-      elif 'open youtube' in query:
12-          query = query.split(" ")
13-          c=query[1:]
14-          location=' '.join(c)
15-          webbrowser.open('https://www.youtube.com/')
16-
17-
18-      elif 'open google' in query:
19-          webbrowser.open("google.com")
20-
21-      elif 'open stackoverflow' in query:
22-          webbrowser.open("stackoverflow.com")
23-
24-      elif 'open gla website' in query:
25-          webbrowser.open("www.gla.ac.in")
26-      elif 'how are you' in query:

```

```

1-          speak('i am fine what can i do for you')
2-      elif 'tell me about yourself' in query:
3-          speak('i am rex your personal virtual assistant created to help you')
4-      elif 'play ' in query:
5-          query=query.split(" ")
6-          location = query[1:]
7-          location=' '.join(location)
8-          webbrowser.open("https://www.youtube.com/results?search_query=" + location)
9-          m.position = (395,306)
10-          for _ in range(10000):
11-              for i in range(10000):
12-                  pass
13-          m.press(Button.left)
14-          m.release(Button.left)
15-          #m.position = (855,618)
16-          for _ in range(1000):
17-              for i in range(1000):
18-                  pass
19-          keyboard.press_and_release('f')

```

Given above condition user want to perform some task by system
 Above condition is checked in elseif condition with the help of query variable
 If query matched with condition than particular task is performed

```

elif 'email to akshat' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "akshat.dwivedi_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")
elif 'email to prakhar' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "prakhar.srivastava_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
        continue
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")

```

```

elif 'email to srishti' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "sristi.shukla_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
        continue
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")
elif 'email to amir sir' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "amir.khan@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
        continue
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")

```

Here is above condition for query name mentioned people to whom email is send
 Using try exception condition and using content also what message is to be send to a
 particular person with their proper email address with a message email has been sent

Project Code

```
import pytsx3
import datetime
import webbrowser
import smtplib
import keyboard
import os
import wikipedia
import speech_recognition as sr
from pynput.mouse import Button, Controller

m = Controller()
engine = pytsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].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 < 17:
        speak("Good Afternoon!!!!")

    else:
        speak("Good Evening!!!!!!")

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

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("User said: %s\n"%query)
```

```

    if 'how are you' not in query and 'tell me about yourself' not in query and "sing for me"
not in query and 'I love you' not in query and 'good job rex':
    speak(query)

```

```

except Exception as e:
    speak("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('ur gmail', 'password')
    server.sendmail('ur gmail', to, content)
    server.close()

```

```

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

        # Logic for executing tasks based on query
        if 'wikipedia for' in query:
            speak('Searching Wikipedia...')
            query = query.split(" ")
            location = query[2:]
            location=' '.join(location)
            webbrowser.open('https://en.wikipedia.org/wiki/'+ location)
            text = wikipedia.summary(query)
            print(text)
            speak(text)

        elif 'open youtube' in query:
            query = query.split(" ")
            c=query[1:]
            location=' '.join(c)
            webbrowser.open('https://www.youtube.com/')

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

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

        elif 'open gla website' in query:
            webbrowser.open("www.gla.ac.in")
        elif 'how are you' in query:
            speak('i am fine what can i do for you')
        elif 'tell me about yourself' in query:
            speak('i am rex your personal virtual assistant created to help you')
        elif 'play ' in query:
            query=query.split(" ")
            location = query[1:]
            location=' '.join(location)
            webbrowser.open("https://www.youtube.com/results?search_query=" + location)

```

```

m.position = (395,306)
for _ in range(10000):
    for i in range(10000):
        pass
    m.press(Button.left)
    m.release(Button.left)
#m.position = (855,618)
for _ in range(1000):
    for i in range(1000):
        pass
    keyboard.press_and_release('f')

elif "where is" in query:
    query = query.split(" ")
    location = query[2:]
    location=' '.join(location)
    speak("Hold on Sir, I will show you where " + location + " is.")
    webbrowser.open('https://www.google.com/maps/place/'+ location)
    m.position = (318,98)
    for _ in range(10000):
        for i in range(10000):
            pass
        m.press(Button.left)
        m.release(Button.left)
elif "record audio" in query:
    r = sr.Recognizer()
    with sr.Microphone() as source:
        speak("Say something!")
        audio = r.listen(source)

# Speech recognition using Google Speech Recognition
data = ""
try:
    # Uses the default API key
    # To use another API key: `r.recognize_google(audio,
key="GOOGLE_SPEECH_RECOGNITION_API_KEY")`
    data = r.recognize_google(audio)
    speak("i recorded: " + data)
    speak("do you want to save this audio")
    with sr.Microphone() as s:
        ad=r.listen(s)
        y=r.recognize_google(ad)
        if "yes" in y:
            speak("file has been successfully saved")
except sr.UnknownValueError:
    print("Google Speech Recognition could not understand audio")
except sr.RequestError as e:
    print("Could not request results from Google Speech Recognition service;
{0}".format(e))

elif 'open playlist' in query:
    music_dir = "C:\\Users\\This PC\\Desktop\\new\\"
    songs = os.listdir(music_dir)
    print(songs)
    os.startfile(os.path.join(music_dir, songs[0]))

```

```

elif 'time' in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak("Sir, the time is %s"%strTime)
elif 'date' in query:
    strDate = datetime.datetime.now().strftime("%d:%B:%Y")
    speak("Sir, the date is %s"%strDate)
elif 'open code' in query:
    codePath = "C:\\Users\\This PC\\Desktop\\rex.py"
    os.startfile(codePath)

elif 'email to akshat' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "akshat.dwivedi_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")
elif 'email to prakhar' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "prakhar.srivastava_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
        continue
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")
elif 'email to simran' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "simran.gupta_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
        continue
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")
elif 'email to srishti' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "sristi.shukla_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
        continue
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")
elif 'email to amir sir' in query:
    try:
        speak("What should I say?")
        content = takeCommand()

```



```

    to = "amir.khan@gla.ac.in"
    sendEmail(to, content)
    speak("Email has been sent!")
    continue
except Exception as e:
    print(e)
    speak("Sorry my friend harry bhai. I am not able to send this email")

elif 'email to pratyush' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "pratyush.khare_cs18@gla.ac.in"
        sendEmail(to, content)
        speak("Email has been sent!")
        continue
    except Exception as e:
        print(e)
        speak("Sorry my friend harry bhai. I am not able to send this email")

elif 'tell me about' in query:
    try:
        speak('this is what i found')
        query=query.split(" ")
        content=query[3:]
        webbrowser.open('https://www.google.com/search?q=' + str(*content))
    except Exception as e:
        speak("sorry")
elif 'good job rex' in query:
    speak('thank you sir')
elif 'i love you' in query:
    speak('I am a Machine. Find another partner for you')
elif "shutdown" in query:
    speak("do you wish to shutdown your computer")
    speak("press enter to shutdown!")
    os.system("shutdown /s /t 1")
elif 'open word' in query:
    speak("opening Microsoft word")
    os.startfile('C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Word.lnk')
elif 'open excel' in query:
    speak("opening Microsoft Excel")
    os.startfile('C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Excel.lnk')
elif 'open powerpoint' in query:
    speak("opening Microsoft powerpoint")
    os.startfile('C:\\ProgramData\\Microsoft\\Windows\\Start
Menu\\Programs\\PowerPoint.lnk')
else:
    try:
        if "none" not in query:
            speak('this is what i found')
            webbrowser.open('https://www.google.com/search?q=' + str(query))
    except Exception as e:
        speak("sorry")

```

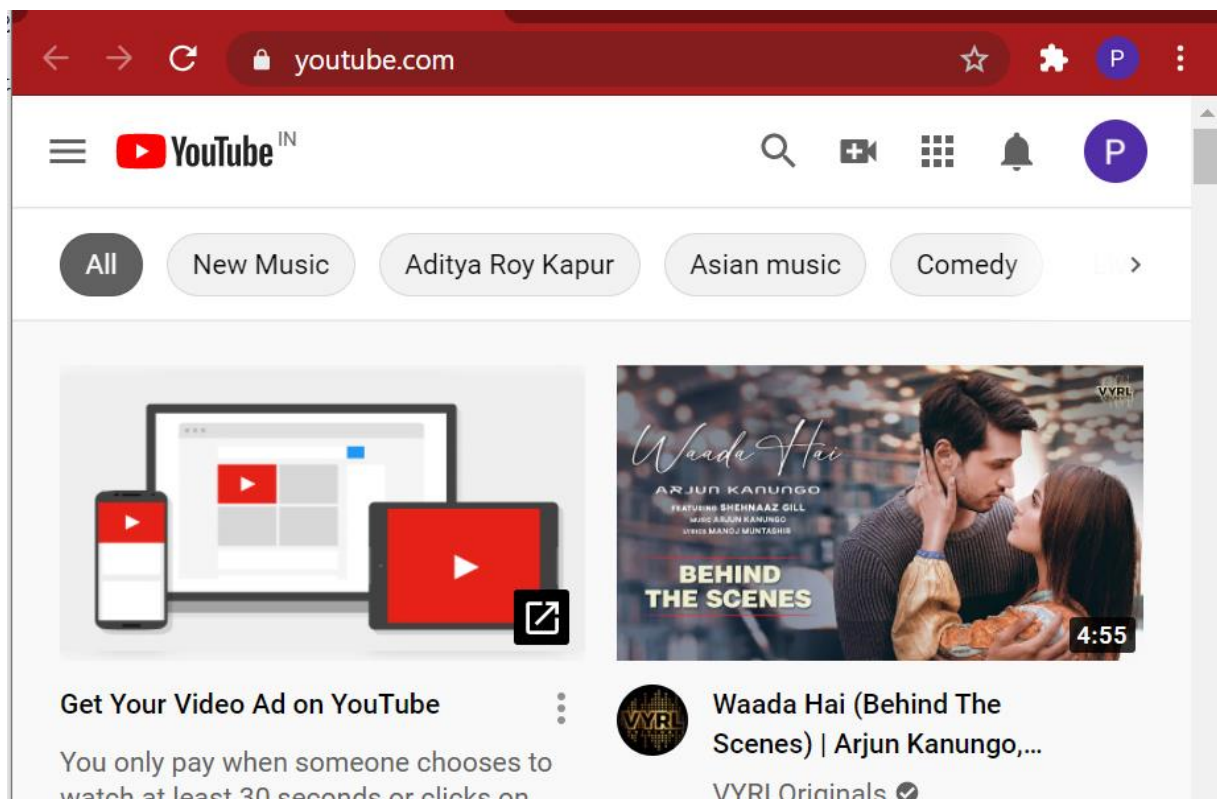
OUTPUT

FOR OPENING YOUTUBE

Command –

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.192 D64] on win32
Type "help", "copyright", "credits" or "license()" for more informat
>>>
===== RESTART: C:\Users\HP\Desktop\VoiceAssistant (1) (1).py
Listening...
Recognizing...
User said: open YouTube
```

Output –



For Someone's Wikipedia

Command-

Listening...
 Recognizing...
 User said: Wikipedia for Mahatma Gandhi

Mohandas Karamchand Gandhi (; 2 October 1869 – 30 January 1948), also known as Mahatma Gandhi, was an Indian lawyer, anti-colonial nationalist, and political ethicist, who employed nonviolent resistance to lead the successful campaign for India's independence from British rule, and in turn inspired movements for civil rights and freedom across the world. The honorific *Mahātmā* (Sanskrit: "great-souled", "venerable"), first applied to him in 1914 in South Africa, is now used throughout the world. Born and raised in a Hindu family in coastal Gujarat, western India, Gandhi trained in law at the Inner Temple, London, and was called to the bar at age 22 in June 1891. After two uncertain years in India, where he was unable to start a successful law practice, he moved to South Africa in 1893 to represent an Indian merchant in a lawsuit. He went on to stay for 21 years. It was in South Africa that Gandhi raised a family, and first employed nonviolent resistance in a campaign for civil rights. In 1915, aged 45, he returned to India. He set about organising peasants, farmers, and urban labourers to protest against excessive land-tax and discrimination. Assuming leadership of the Indian National Congress in 1921, Gandhi led nationwide campaigns for easing poverty, expanding women's rights, building religious and ethnic amity, ending untouchability, and above all for achieving *Swaraj* or self-rule. The same year Gandhi adopted the Indian loincloth, or short dhoti and, in the winter, a shawl, both woven with yarn hand-spun on a traditional Indian spinning wheel, or *charkha*, as a mark of identification with India's rural poor. Thereafter, he lived modestly in a self-sufficient residential community, ate simple vegetarian food, and undertook long fasts as a means of self-purification and political protest. Bringing anti-colonial nationalism to the common Indians, Gandhi led them in challenging the British-imposed salt tax with the 400 km (250 mi) Dandi Salt March in 1930, and later in calling for the British to Quit India in 1942. He was imprisoned for many years, upon many occasions, in both South Africa and India.

Gandhi's vision of an independent India based on religious pluralism was challenged in the early 1940s by a new Muslim nationalism which was demanding a separate Muslim homeland carved out of India. In August 1947, Britain granted independence, but the British Indian Empire was partitioned into two dominions, a Hindu-majority India and Muslim-majority Pakistan. As many displaced Hindus, Muslims, and Sikhs made their way to their new lands, religious violence broke out, especially in the Punjab and Bengal. Eschewing the official celebration of independence in Delhi, Gandhi visited the affected areas, attempting to provide solace. In the months following, he undertook several fasts unto death to stop religious violence. The last of these, undertaken on 12 January 1948 when he was 78, also had the indirect goal of pressuring India to pay out some cash assets owed to Pakistan. Some Indians thought Gandhi was too accommodating. Among them was Nathuram Godse, a Hindu nationalist, who assassinated Gandhi on 30 January 1948 by firing three bullets into his chest. Gandhi's birthday, 2 October, is commemorated in India as Gandhi Jayanti, a national holiday, and worldwide as the International Day of Nonviolence. Gandhi is commonly, though not formally, considered the Father of the Nation in India, and was commonly called *Bapu* (Gujarati: endearment for father, papa).

Output-

The screenshot shows a web browser window displaying the Wikipedia article for Mahatma Gandhi. The browser's address bar shows the URL `en.wikipedia.org`. The Wikipedia logo and navigation links are visible on the left. The main content area displays the article title "Mahatma Gandhi" and the beginning of the text, which describes his role in India's independence movement. A right-hand sidebar contains a portrait of Gandhi and its caption. The Windows taskbar is visible at the bottom of the screen.

To Send email for specific user

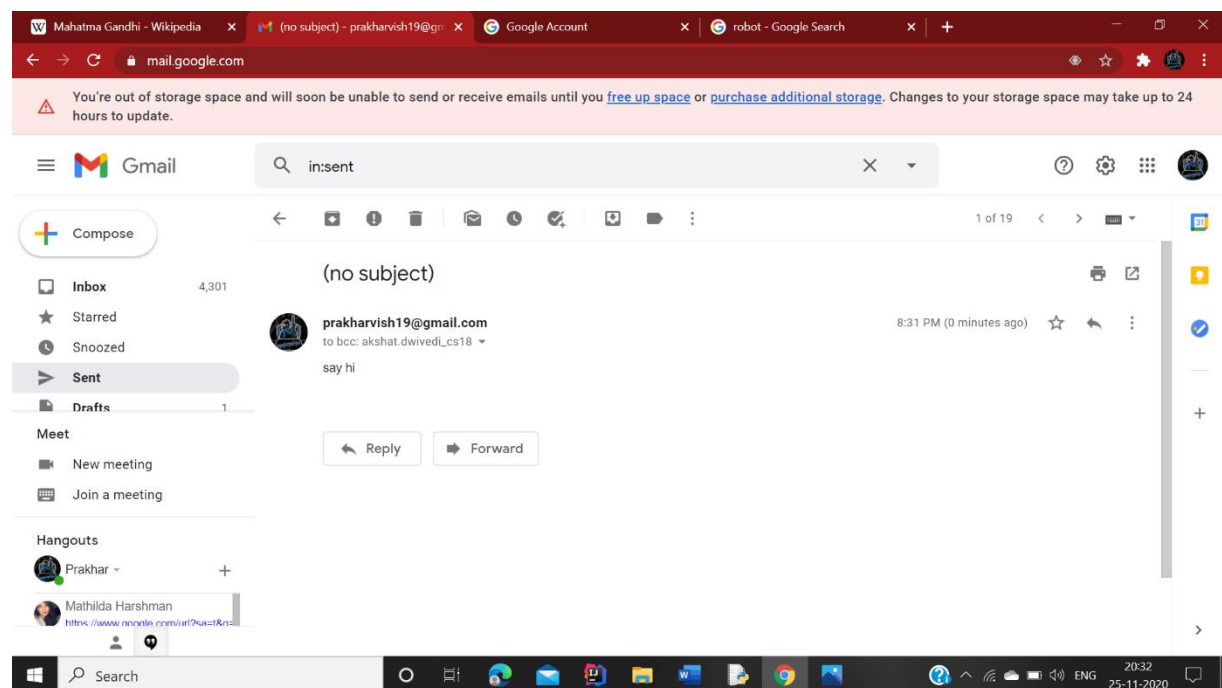
Command-

```

VoiceAssistant (1) (1).py - C:\Users\HP\Desktop\VoiceAssistant (1) (1).py (3.9.0)
File *Python 3.9.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\HP\Desktop\VoiceAssistant (1) (1).py =====
Listening...
Recognizing...
User said: email to Akshat

Listening...
Recognizing...
User said: say hi
  
```

Output -



And Many More.....

References/Bibliography

- a. Google (<https://www.google.com>)
- b. GeeksforGeeks (<https://www.geeksforgeeks.org>)
- c. JavaTpoint (<https://www.javatpoint.com>)