

PROJECT ON
VOICE ASSISTANT

SILIGURI INSTITUTE OF TECHNOLOGY

Project submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Electrical Engineering



By Members (GROUP-1):

Sagar Mandal - 11901621034
Rupa Sharma - 11901621026
Divya Gupta - 11901621022
Priyashree Nandi-11901621030
Amit Kumar Ram-11901621035

SILIGURI INSTITUTE OF TECHNOLOGY
(Maulana Abul Kalam Azad University of Technology (WBUT))
FACULTY OF EE DEPARTMENT

CERTIFICATE OF RACOMENDATION

*The project report titled "**VOICE ASSISTANT**" prepared by **Sagar Mandal**, Roll No:11901621034; **Divya Gupta**, Roll No: 11901621022; **Priyashree Nandi**, Roll No:11901621030; **Rupa Sharma**, Roll No: 11901621026; **Amit Kumar Ram**, Roll No:11901621035; is hereby approved and certified as a creditable study in technological subjects performed in a way sufficient for its acceptance for partial fulfillment of the degree for which it is submitted. Under the direct supervision and guidance of **RIPAM KUNDU**. It is to be understood that by this approval, the undersigned do not, necessarily endorse or approve any statement made, opinion expressed or conclusion drawn Thereon, but approve the project only for the purpose for which it is submitted.*

*Shilpi mam & Dipayan sir
kundu.*

(Trainer of the Head)

Date: 15/09/2022

Mr. Ripam

(Name of Trainer of Project)

Date:15/09/2022

*Mr. Mithun Chakraborty.
(Head of the College)
department)*

*Mr. Arup Das.
(Head of the*

ACKNOWLEDGEMENT

We are thankful to our institute Siliguri Institute Of Technology for providing necessary facilities. It is a great privilege for us to express our profound gratitude to HOD ARUP DAS and other faculties for their constant guidance, valuable suggestions, supervision and inspiration throughout the course work. We would like to express our gratitude to our supervisor for his kind cooperation and encouragement which helped us to do this project .The valuable guidance and interest taken by our respected mentors have been a motivation and source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We are also indebted to head of the department Mr. Arup Das of Electrical Engineering Technology for permitting us to pursue the project.

THANK YOU.

ABSTRACT

In this modern era, day to day life became smarter and interlinked with technology. We already know some voice assistance like google, Siri. etc. Now in our voice assistance system, it can act as a basic medical prescriber, daily schedule reminder, note writer, calculator and a search tool. This project works on voice input and give output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

INDEX

<i>Sl. no.</i>	<i>TITLE</i>	<i>Page No</i>
<i>1</i>	<i>Certificate of Recomodation</i>	<i>1</i>
<i>2</i>	<i>Acknowledgement</i>	<i>2</i>
<i>3</i>	<i>Abstract</i>	<i>3</i>
<i>4</i>	<i>Index</i>	<i>4</i>
<i>5</i>	<i>Introduction</i>	<i>5</i>
<i>6</i>	<i>Proposed Plan Of Work</i>	<i>6</i>
<i>7</i>	<i>Module</i>	<i>7-9</i>
<i>8</i>	<i>Code & Output</i>	<i>10-18</i>
<i>9</i>	<i>Conclusion</i>	<i>19</i>
<i>10</i>	<i>Reference</i>	<i>20</i>

INTRODUCTION

A voice assistant is a digital assistant that uses voice recognition, language processing algorithms, and voice synthesis to listen to specific voice commands and return relevant information or perform specific functions as requested by the user.

Based on specific commands, sometimes called intents, spoken by the user, voice assistants can return relevant information by listening for specific keywords and filtering out the ambient noise.

While voice assistants can be completely software based and able to integrate into most devices, some assistants are designed specifically for single device applications, such as the Amazon Alexa Wall Clock.

Today, voice assistants are integrated into many of the devices we use on a daily basis, such as cell phones, computers, and smart speakers. Because of their wide array of integrations, There are several voice assistants who offer a very specific feature set, while some choose to be open ended to help with almost any situation at hand.

Proposed Plan Of Work

The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer's internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature

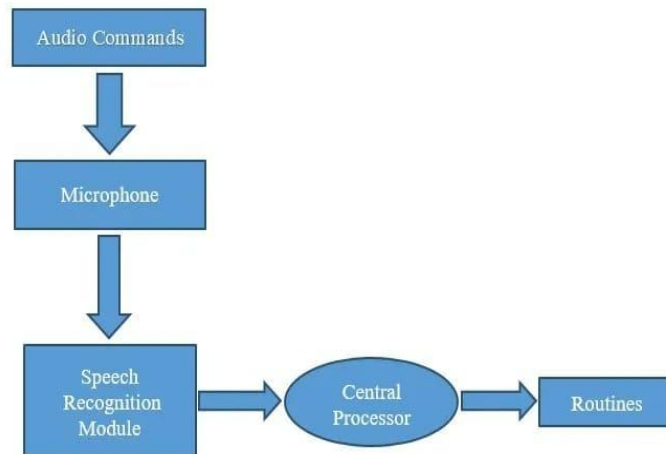


Fig-1

Fig.1 shows the 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 relevant script for execution.

But, the complexities don't stop there. Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient sounds it "hears" of a dog barking or a helicopter flying overhead, from your voice. Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and "tell" the device to filter them out. 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 change.

MODULS

Speech recognition - Speech recognition, or speech-to-text, is the ability of a machine or program to identify words spoken aloud and convert them into readable text. This package can be installed by using `pip install Speech Recognition`.

Play sound – To play audio file in a single line of code. When parameter `block=False`, there's no sound and the script executed without any error.

GTTS - GTTS (Google Text-to-Speech) is a Python library and CLI tool to interface with Google Translate text-to-speech API. We will import the GTTS library from the `gtts` module which can be used for speech translation.

Random - Python offers random module that can generate random numbers. These are pseudo-random number as the sequence of number generated depends on the seed. If the seeding value is same, the sequence will be the same.

Wikipedia - Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia.

Web Browser - The web browser module provides a high-level interface to allow displaying web-based documents to users.

Datetime - Datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction

for output formatting and manipulation. General calendar related functions.

OS(Operating System) - *The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.*

PyAutoGUI - *PyAutoGUI is essentially a Python package that works across Windows, MacOS X and Linux which provides the ability to simulate mouse cursor moves and clicks as well as keyboard button presses.*

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

Request - *The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).*

Smtplib - *The smtplib module defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP or ESMTP listener daemon.*

Urllib - *Urllib package is the URL handling module for python. It is used to fetch URLs (Uniform Resource Locators).*

re - *The Python module re provides full support for Perl-like regular expressions in Python. The re module raises the*

exception re. error if an error occurs while compiling or using a regular expression.

CODE

```
import speech_recognition as sr # recognise speech
import playsound # to play an audio file
from gtts import gTTS # google text to speech
import random # generate random numbers
import wikipedia # acces phase data form wikipedia
import webbrowser # highlevel interphase displaying web-based
import time
import datetime # combination between date and time
import os # to remove created audio files
import pyautogui # screenshot
import pyttsx3 #text to speech libery in python
import requests # http request
import smtplib # mail handeling
import urllib.request
import urllib.parse # handel url
import re

class person:
    name = "
    def setName(self, name):
        self.name = name

class sagar:
    name = "
    def setName(self, name):
        self.name = name

def there_exists(terms):
    for term in terms:
        if term in voice_data:
            return True

def engine_speak(text):
```

```

text = str(text)
engine.say(text)
engine.runAndWait()
def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        engine_speak("Hello Sir! GoodMorning!")

    elif hour>=12 and hour<18:
        engine_speak("Hello Sir! GoodAfternoon!")

    else:
        engine_speak("Hello Sir! GoodEvening!")

    engine_speak("I am sisko.Please tell me how can I help you")
r = sr.Recognizer() # initialise a recogniser
# listen for audio and convert it to text:
def record_audio(ask=""):
    with sr.Microphone() as source: # microphone as source
        if ask:
            engine_speak(ask)
        audio = r.listen(source, 5, 5) # listen for the audio via source
        print("Done Listening")
        voice_data = "
        try:
            voice_data = r.recognize_google(audio) # convert audio to text
        except sr.UnknownValueError: # error: recognizer does not
understand
            engine_speak('I did not get that')
        except sr.RequestError:
            engine_speak('Sorry, the service is down') # error: recognizer is
not connected
        print(">>", voice_data.lower()) # print what user said
        return voice_data.lower()

```

```

# get string and make a audio file to be played
def engine_speak(audio_string):
    audio_string = str(audio_string)
    tts = gTTS(text=audio_string, lang='en') # text to speech(voice)
    r = random.randint(1,800000000000)
    audio_file = 'audio' + str(r) + '.mp3'
    tts.save(audio_file) # save as mp3
    playsound.playsound(audio_file) # play the audio file
    print(sagar_obj.name + ":", audio_string) # print what app said
    os.remove(audio_file) # remove audio file

def email(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('sagarkumar.das@tnu.in', 'S7o6u8v8i4k5#') #change id
    and pass
    server.sendmail('arpan.mukherjee@tnu.in', to, content)
    server.close()

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

    elif hour>=12 and hour<18:
        engine_speak("Hello Sir! Good Afternoon!")

    else:
        engine_speak("Hello Sir! Good Evening!")

    engine_speak("I am sisko . Please tell me how can I help you")
def respond(voice_data):

```

```

# greeting
if there_exists(['hey','hi','hello']):
    greetings = ["hey, how can I help you" + person_obj.name, "hey,
what's up?" + person_obj.name, "I'm listening" + person_obj.name,
"how can I help you?" + person_obj.name, "hello" + person_obj.name]
    greet = greetings[random.randint(0,len(greetings)-1)]
    engine_speak(greet)
# name
if there_exists(["what is your name","what's your name","tell me your
name"]):

    if person_obj.name:
        engine_speak(f"My name is {sagar_obj.name},
{person_obj.name}") #gets users name from voice input
    else:
        engine_speak(f"My name is {sagar_obj.name}. what's your
name?") #incase you haven't provided your name.

if there_exists(["my name is"]):
    person_name = voice_data.split("is")[-1].strip()
    engine_speak("okay, i will remember that " + person_name)
    person_obj.setName(person_name) # remember name in person
object

if there_exists(["what is my name"]):
    engine_speak("Your name must be " + person_obj.name)

if there_exists(["your name should be"]):
    sagar_name = voice_data.split("be")[-1].strip()
    engine_speak("okay, i will remember that my name is " +
sagar_name)
    sagar_obj.setName(sagar_name) # remember name in sagar object

# greeting
if there_exists(["how are you","how are you doing"]):

```

```

    engine_speak("I'm very well, thanks for asking " +
person_obj.name)

# time
if there_exists(["what's the time", "tell me the time", "what time is
it", "what is the time"]):
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    engine_speak(f"Sir, the time is {strTime}")

# search google
if there_exists(["search for"]) and 'youtube' not in voice_data:
    search_term = voice_data.split("for")[-1]
    url = "https://google.com/search?q=" + search_term
    webbrowser.get().open(url)
    engine_speak("Here is what I found for" + search_term)

# search youtube
if there_exists(["youtube"]):
    search_term = voice_data.split("for")[-1]
    search_term = search_term.replace("on
youtube", "").replace("search", "")
    url = "https://www.youtube.com/results?search_query=" +
search_term
    webbrowser.get().open(url)
    engine_speak("Here is what I found for " + search_term)

# get stock price
if there_exists(["price of"]):
    search_term = voice_data.split("for")[-1]
    url = "https://google.com/search?q=" + search_term
    webbrowser.get().open(url)
    engine_speak("Here is what I found for " + search_term + " on
google")

```

```

# weather
if there_exists(["weather"]):
    search_term = voice_data.split("for")[-1]
    url =
"https://www.google.com/search?sxsrf=ACYBGNSQwMLDByBwdVFI
UCbQqya-ET7AAA%3A1578847393212&ei=oUwbXtbXDN-C4-EP-
5u82AE&q=weather&oq=weather&gs_l=psy-
ab.3..35i39i285i70i256j0i67l4j0i131i67j0i131j0i67l2j0.1630.4591..5475.
..1.2..2.322.1659.9j5j0j1.....0....1..gws-
wiz.....10..0i71j35i39j35i362i39._5eSPD47bv8&ved=0ahUKEwiWrJvw
wP7mAhhVfwTgGHfsNDxsQ4dUDCAs&uact=5"
    webbrowser.get().open(url)
    engine_speak("Here is what I found for on google")

# toss a coin
if there_exists(["toss", "flip", "coin"]):
    moves=["head", "tails"]
    cmove=random.choice(moves)
    engine_speak("The computer chose " + cmove)

# screenshot
if there_exists(["capture", "my screen", "screenshot"]):
    myScreenshot = pyautogui.screenshot()
    myScreenshot.save('D://pictures //screen.png')
    engine_speak("screenshot has been captured")

# to search wikipedia for definition
if there_exists(["wikipedia"]):
    search_term = voice_data.split("for")[-1]
    url = "https://en.wikipedia.org/w/index.php?search="+search_term
    webbrowser.get().open(url)
    engine_speak("Here is what I found for" + search_term)

```



```

if there_exists(["exit", "quit", "goodbye"]):
    engine_speak("bye! see you later")
    exit()

# Current location as per Google maps
if there_exists(["what is my exact location", "where am I"]):
    url = "https://www.google.com/maps/search/Where+am+I+?/"
    webbrowser.get().open(url)
    engine_speak("You must be somewhere near here, as per Google
maps")

if there_exists(["play music"]):
    search_term = voice_data.split("for")[-1]
    url = "https://music.youtube.com/search?q="+ search_term
    webbrowser.get().open(url)
    engine_speak("Here is what I found for " + search_term + " on
youtubemusic")

if there_exists(["email", "mail"]):
    try:
        engine_speak("What should I say?")
        content = record_audio()
        to = "rj9832rj@gmail.com"
        email(to, content)
        engine_speak("Email has been sent!")
    except Exception as e:
        print(e)
        engine_speak("Sorry sagar. I am not able to send this email")

if there_exists(["vs code"]):
    codePath =
"C:\\Users\\sagar\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
    engine_speak("opening Visual Studio")

```

```

os.startfile(codePath)

if there_exists(["cmd"]):
    cmdPath = "C:\\WINDOWS\\System32\\cmd.exe"
    os.startfile(cmdPath)
    engine_speak("opening command prompt")

if there_exists(["power off", "shut down"]):
    engine_speak("Do you want to shutdown your laptop")
    while True:
        command = record_audio()
        if "no" in command:
            engine_speak("Thank u sir I will not shut down the computer")
            break
        if "yes" in command:
            # Shutting down
            engine_speak("Shutting down the computer")
            os.system("shutdown /s /t 0")
            break

if there_exists(["restart"]):
    engine_speak("Do you want to restart your laptop")
    while True:
        command = record_audio()
        if "no" in command:
            engine_speak("Thank u sir I will not shut down the computer")
            break
        if "yes" in command:
            # Shutting down
            engine_speak("restarting the computer")
            os.system("shutdown /r /t 0")
            break

person_obj = person()
sagar_obj = sagar()

```

```
sagar_obj.name = 'sisko'
person_obj.name = "sagar"
engine = pyttsx3.init()

wishMe()
while(1):
    voice_data = record_audio("Listening") # get the voice input
    print("Done")
    print("Q:", voice_data)
    respond(voice_data) # respond
```

Output:



Assistent_output.mp4

CONCLUSION

*In this project “**Voice Assistant**” we discussed the design and implementation of Digital Assistance. The project is built using opensource software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easy to add additional features without disturbing current system functionalities.*

It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire system works on the verbal input rather than the next one.

REFERENCE

- [1] <https://extrudesign.com/virtual-assistant-using-python/>
- [2] <https://www.geeksforgeeks.org/voice-assistant-using-python/>
- [3] <https://alan.app/blog/voiceassistant-2/amp/#referrer=https://www.google.com&csi=0>
- [4] <https://www.geeksforgeeks.org/play-sound-in-python/>