COMPUTER SCIENCE

PRACTICAL PROJECT (2020-2021)

R. HARSHARAJ XII-A2 06

K. JISHNURAJ XII-A2 08

R. MUHMMAD ABRAR XII-A2 15

D. SREEJITH XII-A2 29

PROJECT

J.A.S.H

JUST A SIMPLE HELPER

JASH- a PYTHON based VOICE ASSISTANT

PROBLEM DEFINITION:

Using python programming language an AI (Artifical intelligence) is made to provide various purpose for the user.

Artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience. Since the development of the digital computer in the 1940s, it has been demonstrated that computers can be programmed to carry out very complex tasks as, for example, discovering proofs for mathematical theorems or playing chess with great proficiency. Still, despite continuing advances in computer processing speed and memory capacity, there are as yet no programs that can match human flexibility over wider domains or in tasks requiring much everyday knowledge. On the other hand, some programs have attained the performance levels of human experts and professionals in performing certain specific tasks, so that artificial intelligence in this limited sense is found in applications as diverse as medical diagnosis, computer search engines, and voice or handwriting recognition.

Voice assistant can be used in cases of visual impairment and text is also displayed which can be useful in situations of hearing impairment. All our end users need to summon a query and expect the answers they require with the help of a voice engine by Google which we have utilized in our project.

PROJECT ANALYSIS :

MODULES INCLUDED :

As we built this project in python we used a number of modules that ensure the working of this project . The modules we included are as follows :

- import gTTs
- import time
- import playsound
- import pyaudio
- import speech_recognition as sr
- import os
- import random
- import webbrowser
- import subprocess
- import wikipedia
- import time
- import translate

FUNCTIONS INCLUDED:

1. SQL BASED:

- displayres(are):- To display the details of Restaurants in an area.
- displayhos(are):- To display the details of Hospitals in an area.
- displayfuel(are):- To display the details of Fuels Stations in an area.

2. speak(audio):- To convert a given text to mp3 and playing the audio.

3. translate():- To translate given phrases from English to other languages.

4. wiki(s):- To retrieve information from wikipedia.com .

CODE:

The code is usually built on the basis of speech recognition and interpretaion which is achieved by using gTTs module(Google text to speech) . The project is aimed at providing personal asssistance for users. This AI also provides a lot of services from saying the time to cracking jokes (if the user feels bored). It can also surf the internet with the help of the user's voice command. We are using **Google speech recognition API** and google text to speech for voice input and output respectively. The system's microphone and speaker are used as source of input and output respectively. Key features of our project are as follows:-

1. User Greetings:

- Commands- Hello, Jash, How are you, Thanks, Your name?

2. Exception handling:

- Many users tend to play with their voice assistants by using some inappropriate words which are recognized and replied by our assistant.
- If the internet connection is not proper "Could not request results" will be returned.
- If the audio is not clear "Oops! I could not understand audio" will be returned.

3. Wikipedia:

   - This function can be invoked by prefixing “tell me about” before the phrase the user actually wants to know about. Eg: ‘Tell me about Abdul Kalam ’ will return 2 sentences about Abdul Kalam from Wikipedia.com

4. Translate:

   - This function can be invoked by saying “translator”. Used to translate English to 106 other languages.

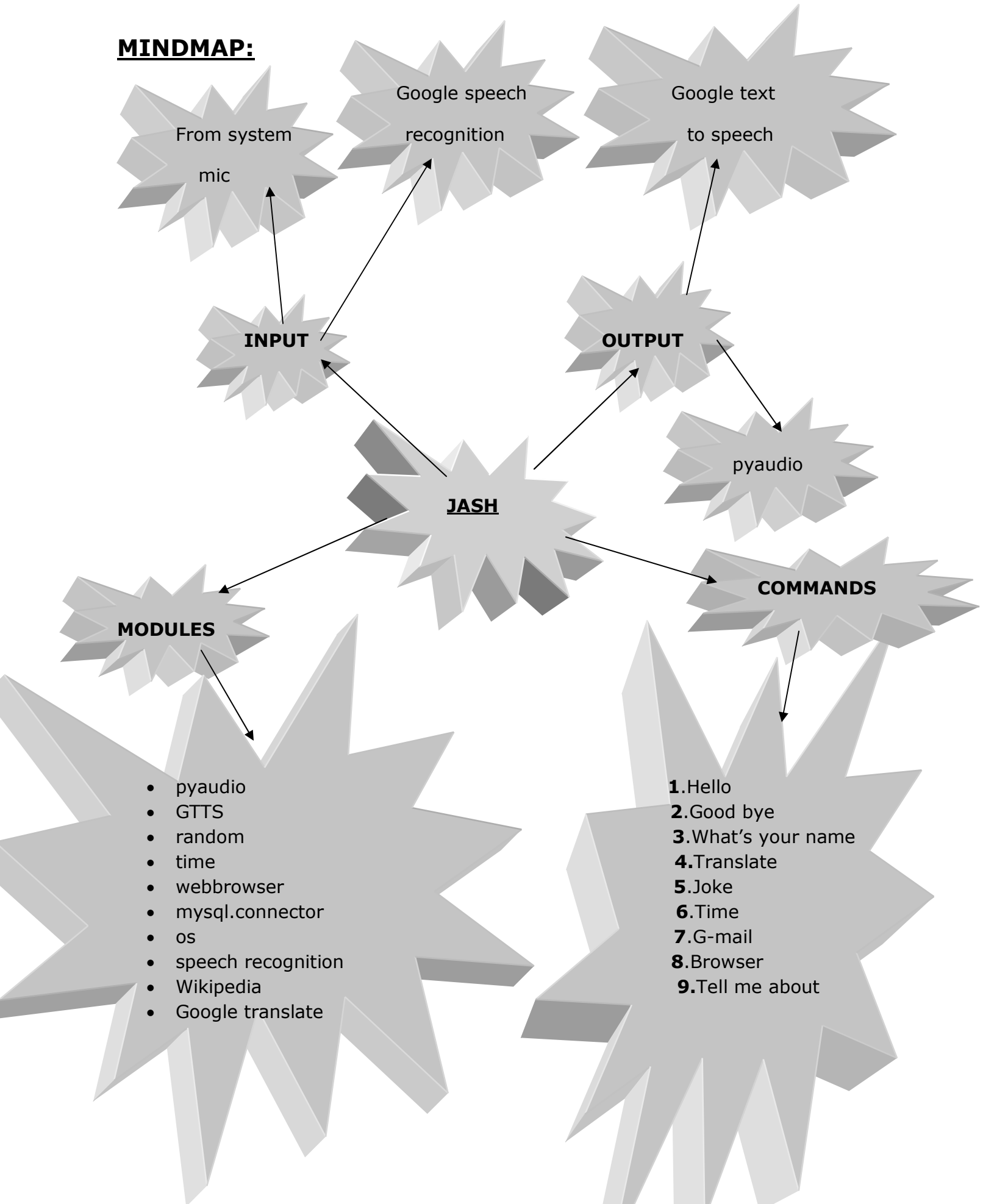
5. Connection to SQL Database:

   - This function can be invoked by saying “search for nearby locations.”
   - The database consists of records of Hospitals, Restaurants and Petrol Stations from 3 areas. [ARUMBAKKAM, ANNA NAGAR, VADAPALANI]

6. Other Functionalities:

   - Shutting down the system: Invoked by “Go to sleep” command.
   - Games: Invoked by “Play some games” command.
   - Time: Invoked by “What is the time now” command.
   - Joke: Invoked by “Tell me a joke” command.
   - Open websites: Invoked by suffixing ‘.com’ behind the website name the user wishes to open.
   - Navigation: Invoked by “open maps” command.
   - System Information: Invoked by “Show me system information” command.

MINDMAP:


From system mic
Google speech recognition
Google text to speech
INPUT
OUTPUT
pyaudio
JASH
MODULES
COMMANDS
pyaudio
GTTS
random
time
webbrowser
mysql.connector
os
speech recognition
Wikipedia
Google translate
1.Hello
2.Good bye
3.What's your name
4.Translate
5.Joke
6.Time
7.G-mail
8.Browser
9.Tell me about

PROJECT CODE:

```
from gtts import gTTS

import time

import playsound

import pyaudio

import speech_recognition as sr

import os

import random

import webbrowser

import subprocess

from time import localtime, strftime

import time

import translate

from googletrans import Translator

import mysql.connector

import wikipedia

doss = os.getcwd()

i=0

n=0

mydb=mysql.connector.connect(host='localhost',database='nearbylo
c',user='root',passwd='sample')

mycursor=mydb.cursor()

mycursor.execute("SELECT * FROM hos where Area='ARUMBAKKAM'")

hosar=mycursor.fetchall()

mycursor.execute("SELECT * FROM hos where Area='ANNA NAGAR'")
```

```
hosan=mycursor.fetchall()
mycursor.execute("SELECT * FROM hos where Area='VADAPALANI'")
hosvd=mycursor.fetchall()
mycursor.execute("SELECT * FROM res where Area='ARUMBAKKAM'")
resar=mycursor.fetchall()
mycursor.execute("SELECT * FROM res where Area='ANNA NAGAR'")
resan=mycursor.fetchall()
mycursor.execute("SELECT * FROM res where Area='VADAPALANI'")
resvd=mycursor.fetchall()
mycursor.execute("SELECT * FROM fuel where Area='ARUMBAKKAM'")
flsar=mycursor.fetchall()
mycursor.execute("SELECT * FROM fuel where Area='ANNA NAGAR'")
flsan=mycursor.fetchall()
mycursor.execute("SELECT * FROM fuel where Area='VADAPALANI'")
flvd=mycursor.fetchall()
def displayhos(are):
        if are=='ARUMBAKKAM':
            for i in hosar:
                print(i)
        elif are=='ANNA NAGAR' or are=='ANNANAGAR':
            for i in hosan:
                print(i)
        elif are=='VADAPALANI':
            for i in hosvd:
                print(i)
```

```
def displayres(are):
        if are=='ARUMBAKKAM':
            for i in resar:
                print(i)
        elif are=='ANNA NAGAR':
            for i in resan:
                print(i)
        elif are=='VADAPALANI':
            for i in resvd:
                print(i)
def displayfuel(are):
        if are=='ARUMBAKKAM':
            for i in flsar:
                print(i)
        elif are=='ANNA NAGAR' or are=='ANNANAGAR':
            for i in flsan:
                print(i)
        elif are=='VADAPALANI':
            for i in flvd:
                print(i)
def speak(audio):
    tts = gTTS(text=rand,lang='en')
    filename ='speech'+str(random.randint(1,1000))+'.mp3'
    tts.save(filename)
    print('Jash  : ',rand)
```

```
    playsound.playsound(filename)


def translate():

    ip=input('Enter the statement you want to translate : ')

    languages={'af': 'afrikaans', 'sq': 'albanian', 'am':
'amharic', 'ar': 'arabic', 'hy': 'armenian', 'az':
'azerbaijani', 'eu': 'basque', 'be': 'belarusian', 'bn':
'bengali', 'bs': 'bosnian', 'bg': 'bulgarian', 'ca': 'catalan',
'ceb': 'cebuano', 'ny': 'chichewa', 'zh-cn': 'chinese
(simplified)', 'zh-tw': 'chinese (traditional)', 'co':
'corsican', 'hr': 'croatian', 'cs': 'czech', 'da': 'danish',
'nl': 'dutch', 'en': 'english', 'eo': 'esperanto', 'et':
'estonian', 'tl': 'filipino', 'fi': 'finnish', 'fr': 'french',
'fy': 'frisian', 'gl': 'galician', 'ka': 'georgian', 'de':
'german', 'el': 'greek', 'gu': 'gujarati', 'ht': 'haitian
creole', 'ha': 'hausa', 'haw': 'hawaiian', 'iw': 'hebrew', 'hi':
'hindi', 'hmn': 'hmong', 'hu': 'hungarian', 'is': 'icelandic',
'ig': 'igbo', 'id': 'indonesian', 'ga': 'irish', 'it':
'italian', 'ja': 'japanese', 'jw': 'javanese', 'kn': 'kannada',
'kk': 'kazakh', 'km': 'khmer', 'ko': 'korean', 'ku': 'kurdish
(kurmanji)', 'ky': 'kyrgyz', 'lo': 'lao', 'la': 'latin', 'lv':
'latvian', 'lt': 'lithuanian', 'lb': 'luxembourgish', 'mk':
'macedonian', 'mg': 'malagasy', 'ms': 'malay', 'ml':
'malayalam', 'mt': 'maltese', 'mi': 'maori', 'mr': 'marathi',
'mn': 'mongolian', 'my': 'myanmar (burmese)', 'ne': 'nepali',
'no': 'norwegian', 'ps': 'pashto', 'fa': 'persian', 'pl':
'polish', 'pt': 'portuguese', 'pa': 'punjabi', 'ro': 'romanian',
'ru': 'russian', 'sm': 'samoan', 'gd': 'scots gaelic', 'sr':
'serbian', 'st': 'sesotho', 'sn': 'shona', 'sd': 'sindhi', 'si':
'sinhala', 'sk': 'slovak', 'sl': 'slovenian', 'so': 'somali',
'es': 'spanish', 'su': 'sundanese', 'sw': 'swahili', 'sv':
'swedish', 'tg': 'tajik', 'ta': 'tamil', 'te': 'telugu', 'th':
'thai', 'tr': 'turkish', 'uk': 'ukrainian', 'ur': 'urdu', 'uz':
'uzbek', 'vi': 'vietnamese', 'cy': 'welsh', 'xh': 'xhosa', 'yi':
'yiddish', 'yo': 'yoruba', 'zu': 'zulu', 'fil': 'Filipino',
'he': 'Hebrew'}

    print(languages)
```

```
    print('You can use the above code for your own language!')
    lang=input('Enter the language-code you want this to be
translated : ')
    translator=Translator()
    k=translator.translate(ip,dest=lang)
    print(k)
def wiki(s):
    print(wikipedia.summary(s,sentences=2))
while (i<1):
    r = sr.Recognizer()
    with sr.Microphone() as source:
        audio = r.adjust_for_ambient_noise(source)
        n=(n+1)
        print("LISTENING....")
        audio = r.listen(source)
                                                    # interprete
audio (Google Speech Recognition)
    try:
        s = r.recognize_google(audio)
        message = (s.lower())
        print ('User       : ',message)
        if 'goodbye' in message or ('power off') in message or
('stop') in message:
            rand ='Goodbye Sir. Jash powering off in 3.. 2.. 1..
0'
            speak(audio)
            break
```

```
        if 'translator' in message:
            translate()
        if 'tell me about' in message:
            se=message[14::]
            wiki(se)
        elif ('hello') in message or ('hi') in message:
            rand ="Hello Sir. I'm Jash - Your Assistant . At
Your Service Sir. "
            speak(audio)
        elif ('thanks') in message or ('tanks') in message or
('thank you') in message:
            l1=['You are welcome','You are Extremely Welcome
!','Always a pleasure ! ','You are welcome,I was literally made
for this thing..!']
            rand =random.choice(l1)
            speak(audio)
        elif ('jash') in message:
            rand ='Yes Sir . Waiting for your command!..'
            speak(audio)
        elif  ('how are you') in message or ('and you') in
message or ('are you okay') in message:
            rand ="I'm fine! What can I do for you??"
            speak(audio)
        elif  ('f***') in message or ('a******') in message:
            rand ='Be polite please'
            speak(audio)
        elif ('your name') in message:
```

```
            rand ="I'm Jash. At your service dude!"
            speak(audio)
        elif ('wi-fi') in message:
            REMOTE_SERVER = "www.google.com"
            rand ='We are connected'
            speak(audio)
        elif ('.com') in message :
            rand = 'Opening' + message[5:]
            Chrome = "C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe %s"
            speak(audio)
            mess=message[5:-1]+'m'
            webbrowser.get(Chrome).open('http://www.'+mess)
            print ('')
        elif ('google maps') in message:
            query = message
            stopwords = ['google', 'maps']
            querywords = query.split()
            resultwords  = [word for word in querywords if
word.lower() not in stopwords]
            result = ' '.join(resultwords)
            Chrome = ("C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe %s")

webbrowser.get(Chrome).open("https://www.google.com/maps/place/"
+result+"/")
            rand = 'Results shown!'
```

```
            tts = gTTS(text=rand,lang='en')
            filename ='speech13.mp3'
            tts.save(filename)
            print('Jash : ',rand)
            playsound.playsound(filename)

        if ('nearby') in message:
            choice=int(input('Enter [1] for HOSPITALS and [2]
for RESTAURANTS AND [3] FOR FUEL PUMPS: '))
            area=input('Enter your area : ')
            are=area.upper()
            if choice==1:
                displayhos(are)
            elif choice==2:
                displayres(are)
            elif choice==3:
                displayfuel(are)
            else:
                print('INVALID CHOICE!')

        if ('sleep mode') in message:
            rand = ['good night']
            speakmodule.speak(rand,n,mixer)
            os.system('rundll32.exe powrprof.dll,SetSuspendState
0,1,0')
```

```
        if ('time') in message:
            tim = strftime("%X", localtime())
            rand = tim
            speak(audio)


        if ('joke') in message:
            ques=['What's the best thing about
Switzerland?','Hear about the new restaurant called Karma?']
            ans=['I don't know, but the flag is a big
plus.','There's no menu: You get what you deserve.' ]
            k=random.randint(0,len(ques)-1)
            rand=ques[k]
            speak(audio)
            time.sleep(1)
            rand=ans[k]
            speak(audio)
        if ('system information') in message :
            rand='Your system information will be shown in a
Command Window!'
            speak(audio)
            os.system('cmd /k "systeminfo"')
        if ('games')in message or ('game') in message:
            rand="I can help you to play the following games.."
            speak(audio)
            print('Ant, Bagels, Bounce, Cannon, Connect, Crypto,
Fidget, Flappy, Guess, Life, Maze, Memory, Minesweeper, Pacman,
Paint, Pong, Simonsays, Snake, Tictactoetiles, Tron')
```

```
            rand='Please enter your choice..'
            speak(audio)
            choice=input(' ')
            chf=choice.lower()
            fin='python -m freegames.'+chf
            os.system(fin)
     if ('what can you do') in message:
            rand="Here's what I can do: "
            speak(audio)
            print('1. Open URL. ')
            print('2. Navigation - Google maps.')
            print('3. Jokes.')
            print('4. Translate Phrases.')
            print('5. Tell you about a person, place and many
such things.')
            print('4. Check connection status.')
            print('5. Show system Information.')
            print('6. Play games.')
            print('7. Tell you the time.')
            print('8. Search for Nearby Locations')
    except sr.UnknownValueError:
        print("Oops! I could not understand audio")
        rand='Try Saying It Again...'
        speak(audio)
    except sr.RequestError as e:
        print("Could not request results$; {0}".format(e))
```

OUTPUT:

```
LISTENING....
User       :  hello
Jash  :  Hello Sir. I'm Jash - Your Assistant . At Your Service Sir.
LISTENING....
User       :  what can you do
Jash  :  Here's what I can do:
1. Open URL.
2. Navigation - Google maps.
3. Jokes.
4. Translate Phrases.
5. Tell you about a person, place and many such things.
4. Check connection status.
5. Show system Information.
6. Play games.
7. Tell you the time.
8. Search for Nearby Locations
LISTENING....
User       :  tell me a joke
Jash  :  Hear about the new restaurant called Karma?
Jash  :  There's no menu: You get what you deserve.
```

	S.NO	NAME	AREA	ADDRESS	PHONE NO.
▶	1	INDIAN OIL	ARUMBAKKAM	JAWAHARLAL NEHRU MAIN ROAD	729410983
	2	TOTAL PETROL PUMP	ARUMBAKKAM	POONAMALLEE HIGH ROAD	38240322
	3	BHARAT PETROLEUM	ANNA NAGAR	1ST AVENUE	729410234
	4	HP PETROL BUNK	ANNA NAGAR	2ND AVENUE	38240242
	5	SHELL PETROL PUMP	VADAPALANI	ARCOT ROAD	729410542
	6	HP BETROL BUNK	VADAPALANI	ARCOT ROAD	38240232

	S.NO	NAME	AREA	ADDRESS	PHONE NO.
▶	1	PRIME INDIAN HOSPITALS	ARUMBAKKAM	POONAMALLEE HIGH ROAD	44892340
	2	VASANTHI ORTHOPAEDIC HOSPITAL	ARUMBAKKAM	RAZACK GARDEN MAIN ROAD	44893340
	3	GOVT. SIDHA HOSPITAL	ARUMBAKKAM	NEAR ANNA ARCH	44892320
	4	UMA EYE CLINIC	ANNA NAGAR	2ND AVENUE	443792340
	5	SUNDARAM MEDICAL FOUNDATION	ANNA NAGAR	4TH AVENUE	443792540
	6	VIHAA MULTISPECIALITY HOSPITAL	ANNA NAGAR	3RD AVENUE	443792240
	7	VIJYA HOSPITALS	VADAPALANI	ARCOT ROAD	44292340
	8	SIMS	VADAPALANI	JAWAHARLAL MAIN ROAD (100FT ROAD)	44193340
	9	PALLAVA HOSPITAL	VADAPALANI	JAWAHARLAL NEHRU MAIN ROAD (100FT ROAD)	44832320

	S.NO	NAME	AREA	ADDRESS	PHONE NO.
▶	1	A2B VEG RESTAURANT	ARUMBAKKAM	JAWAHARLAL NEHRU MAIN ROAD (100FT ROAD)	44892340
	2	VELU MILITARY HOTEL	ARUMBAKKAM	POONAMALLEE HIGH ROAD	44893340
	3	HOTEL SHAN ROYAL	ARUMBAKKAM	NEAR KOYAMBEDU FLYOVER	44892320
	4	MC DONALDS	ANNA NAGAR	2ND AVENUE	443792340
	5	LITTLE HUT	ANNA NAGAR	4TH AVENUE	443792540
	6	HAUNTED	ANNA NAGAR	3RD AVENUE	443792240
	7	NAMMA VEEDU VASANTHA BHAVAN	VADAPALANI	ARCOT ROAD	44292340
	8	KFC	VADAPALANI	FORUM VIJAYA MALL	44193340
	9	MC DONALDS	VADAPALANI	FORUM VIJAYA MALL	44832320

SYSTEM REQUIREMENTS :

For this project the system requirements are as follows:

- IDLE ver.3.7 64bit
- A good Internet connection
- Pre-Installed Python modules (AS MENTIONED ABOVE)
- Chrome web-browser.
- System Mic
- System Speaker
- MySQL Workbench 8.0 CE

FUTURE ENHANCEMENTS:

This project is utilized as a key-framework for many existing projects like

- Google assistant
- Amazon's Alexa
- Apple's Siri
- Hike's Natasha
- And many more…

Some of the examples of future enhancements of this project are:

- Blue Brain Project, an attempt to create a synthetic brain by reverse-engineering the mammalian brain down to the molecular level.
- Google Brain A deep learning project part of Google X attempting to have intelligence similar or equal to human-level.
- Human Brain Project
- NuPIC, an open source implementation by Numenta of its cortical learning algorithm.

BIBLIOGRAPHY:

1. https://www.geeksforgeeks.org/performing-google-search-using-python-code/
2. https://www.geeksforgeeks.org/convert-text-speech-python/
3. https://pythonbasics.org/python-play-sound/
4. https://people.csail.mit.edu/hubert/pyaudio/docs/
5. http://net-informations.com/python/net/browser.htm
6. Computer Science by Sumita Arora Class(XII) 2020 Edition