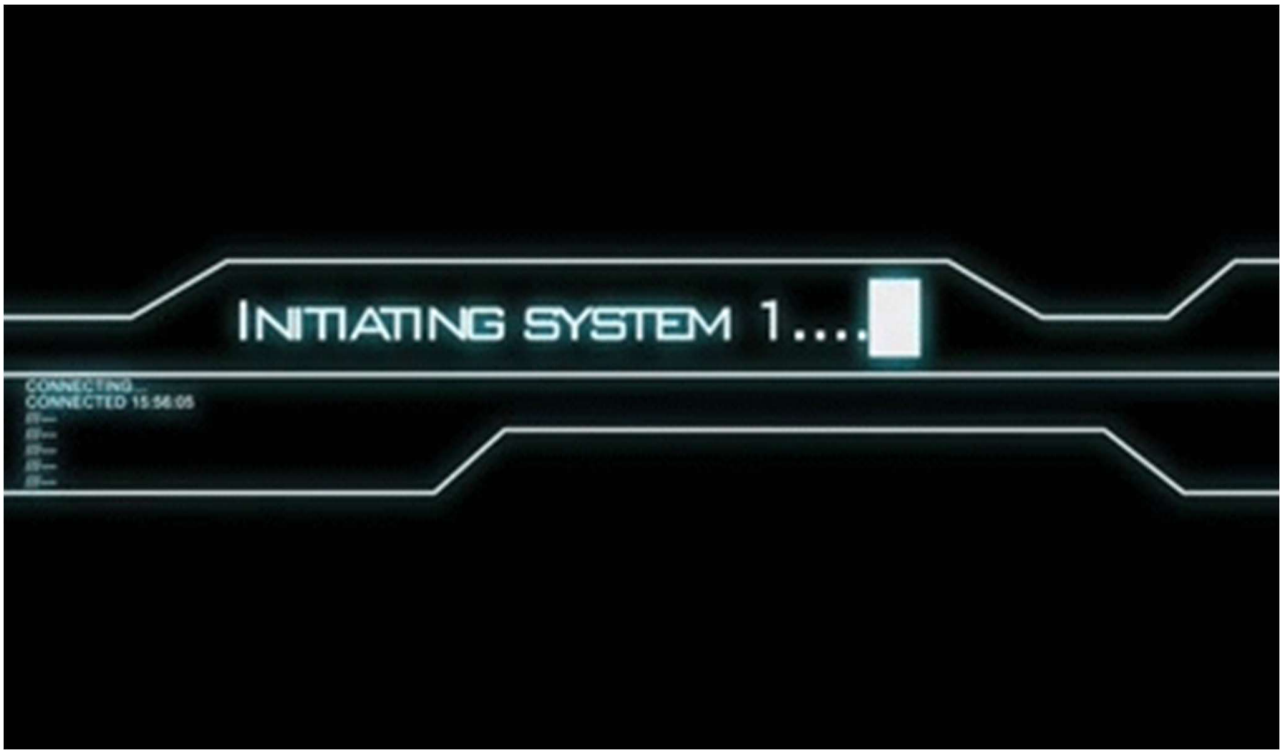
OPERATING SYSTEM PROJECT

PROJECT NXTERMINAL

-A smarter Terminal for Operating Systems-



Project By

Divyansh

INDEX

ACKNOWLEDGEMENT

Our gratitude goes to Prof Thomas Abraham for the knowledge I gained from him and for pushing me to the extreme during his classes. I actually learnt a lot from him and for guiding us to make this project better through suggestions every time we met him for review .

Objective

The main objective of this project is to create an advanced next generation easy-to-use voice activated terminal, that has conventional functionalities of a terminal and highly advanced functionalities too. We are planning to do these by integrating Core OS principles and programming with modern API Cloud tools and Cloud Computing for Speech Recognition for High Performance Fast Terminal

Project Abstract

The main object of this project is to create a much smarter, easy to use OS terminal with more functionalities. This project proposes integration of modern Artificial Intelligence concepts to Operating System concepts in order to provide a much efficient and smarter result to provide the enhancement to common functionalities with additional operations too

Functionalities

[1]Help

[2]Make Directory

[3]Delete Directory

[4]Create File

[5]Edit File Content

[6]Delete File

[7]Rename file

[8]Switch between Speech and Text

[9]List Files

[10]Change Directory

[11]Find my Phone

[12]Where am I?

Working Tools & Implementation Details

Operating System Programming using Python

The os and sys modules provide numerous tools to deal with filenames, paths, directories. The os module contains two sub-modules os.sys (same as sys) and os.path that are dedicated to the system and directories; respectively.

Whenever possible, you should use the functions provided by these modules for file, directory, and path manipulations. These modules are wrappers for platform-specific modules,

so functions like os.path.split work on UNIX, Windows, Mac OS, and any other platform supported by Python.

You can build multi-platform path using the proper separator symbol:

```
>>> import os
>>> import os.path
>>> os.path.join(os.sep, 'home', 'user', 'work')
'/home/user/work'
```

Manipulating Directories

The **getcwd()** function returns the current directory (in unicode format with **getcwdu()**).The current directory can be changed using **chdir()**:

os.chdir(path)

The **listdir()** function returns the content of a directory. Note, however, that it mixes directories and files.

The **mkdir()** function creates a directory. It returns an error if the parent directory does not exist. If you want to create the parent directory as well, you should rather use **makedirs()**:

```
>>> os.mkdir('temp') # creates temp directory inside the current directory
>>> os.makedirs(/tmp/temp/temp")
```

Once created, you can delete an **empty** directory with **rmdir()**:

```
>>> import os
>>> os.mkdir('/tmp/temp')
>>> os.rmdir('/tmp/temp')
```

Renaming files or directories

You can rename a file from an old name to a new one by using **os.rename()**. See also **os.renames()**.

PyAudio

We use this python library to create wav files using our own microphone. It makes .wav files from the input we get from the microphone. We can also playback the audio files. We use this module to make .wav files which is read by the Google's NLP speech API to recognize speech and return it as a string

Google Speech NLP API

Powerful speech recognition

Google Cloud Speech-to-Text enables developers to convert audio to text by applying powerful neural network models in an easy-to-use API. The API recognizes 120 languages and variants to support your global user base. You can enable voice command-and-control, transcribe audio from call centers, and more. It can process real-time streaming or prerecorded audio, using Google's machine learning technology.
Powered by machine learning
Apply the most advanced deep-learning neural network algorithms to audio for speech recognition with unparalleled accuracy. Cloud Speech-to-Text accuracy improves over time as Google improves the internal speech recognition technology used by Google products.

Automatically identifies spoken language

Using Cloud Speech-to-Text you can identify what language is spoken in the utterance (limit to four languages). This can be used for voice search (such as, "What is the temperature in Paris?") and command use cases (such as, "Turn the volume up.")

Contact System using Twilio API

Using the Twilio REST API, you can make outgoing calls to phones, SIP-enabled endpoints, and Twilio Client browser connections.

Initiate an outbound call with Twilio

To place an outbound call, a phone call from a Twilio phone number to an outside number, you must make an HTTP POST request to your account's

Calls initiated via the REST API are rate-limited to one per second. You can queue up as many calls as you like as fast as you want, but each call is popped off the queue at a rate of one per second.

Your POST request to the API must include the parameters From and To for Twilio to know where to direct the outbound call and what to use as the caller ID.

Specify the call's recipient

The To parameter (required) is the phone number, SIP address, or client identifier you're calling. Phone numbers should be formatted with a '+' and country code e.g., +16175551212 (E.164 format).

If you are making calls from a trial account, the To phone number must be verified with Twilio. You can verify your phone number by adding it to your Verified Caller IDs in the console.

SIP addresses must be formatted as sip:name@example.com. For example, to dial Pat's SIP address at Example Company, the Toparameter should be sip:pat@example.com.

Client identifiers must begin with the client:URI scheme. For example, to call a client named joey, the To parameter should be client:joey.

Specify the caller ID

Twilio uses the From parameter (required) to set a phone number or client identifier as the caller ID for your outbound call.

If you used a phone number for your To value in your POST request, the From value you specify must also be a phone number. Just as with the To parameter, phone numbers should be formatted with a '+' and country code, e.g., +16175551212 (E.164 format).

Any phone number you specify here must be a Twilio phone number (you can purchase a number through the console) or a verified outgoing caller id for your account.

If you use a client identifier as the value for From, your identifier must begin with the client:URI scheme. For example, to set a client named charlie as your caller ID, your From parameter should be client:charlie.

Tell Twilio what to do on the call

When you initiate an outbound call with the Twilio REST API, Twilio needs to access your instructions for how to handle the call. It does this by making a synchronous HTTP request to either:

a URL that hosts a set of TwiML instructions (this could be an XML document or web application) or the set of URLs and configuration you've created as an application in the Twilio console

Text to Speech

We are using google's text to speech to provide more interactiveness and improve accessibility. We pass the contents to be said as a parameter to google's api and the language we want it to be said it. The result comes back as .mp3 format.

We use pydub to convert the mp3 format to a .wav format which enables us to playback the audio file with good quality

Individual Contribution

Siddarth S	● Made Project Outline and Terminal Design ● Integrated Twilio API ● Integrated Google Map API ● Added 6 functionalities
Ishaan kolli	● Created module to make and read .wav files ● Integrated Speech recognition ● Implemented TTS ● Enabled complete voice navigation on edit file

Future Work

We can expand this project scope to create a complete smarter functioning Operating System, and this could be used to help amateur programmers start of with the field of Computer Science We also plan to make this project help the differently abled by helping them use their own systems in a much simpler way thereby making it better.

Future plan

[1] Improve Speech-text Accuracy
[2]Improve Speed
[3] Create a better easy-to-use Interface

CODE

```
from pydub import AudioSegment
from gtts import gTTS
import speech_recognition as sr
from twilio.rest import Client
import pyaudio
import time
import sys
import wave
import os
import webbrowser


#amount of data used at a time
CHUNK = 1024
#input and output to .wav file
FORMAT = pyaudio.paInt16
CHANNELS = 2
RATE = 44100
#amont of seconds to record at a time
RECORD_SECONDS = 4

#function to convert speech to text
def recognizespeech():
   #r has the speech recognizer class
   r =sr.Recognizer()
   #takes the audio from the output
   audiosource = sr.AudioFile('output.wav')
   with audiosource as source:
       audio = r.record(source)
   #Prints the recognized data

   try:
       data = r.recognize_google(audio)
       return data
   except sr.UnknownValueError:
       return ("Google Speech Recognition could not understand audio")
```

```
    except sr.RequestError as e:
        return ("Could not request results from Google Speech
Recognition service; {0}".format(e))

def readwavfile():
    #opens a wav file
    wf = wave.open('tts.wav','rb')
    p = pyaudio.PyAudio()
    stream = p.open(format=FORMAT,
                channels=CHANNELS,
                rate=12700,
                output=True)

    #takes data chunks at a time
    data = wf.readframes(CHUNK)

    #reads out data from wav file
    while len(data) > 0:
        stream.write(data)
        data = wf.readframes(CHUNK)

    stream.stop_stream()
    stream.close()

    p.terminate()

def makewavfile():
    p = pyaudio.PyAudio()
    #records from microphone and outputs it to output.wav file
    WAVE_OUTPUT_FILENAME = "output.wav"

    stream = p.open(format=FORMAT,
                channels=CHANNELS,
                rate=RATE,
                input=True,
                frames_per_buffer=CHUNK)

    print("* recording")

    frames = []

    #gets frame chunks one at a time and appends it

    for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
        data = stream.read(CHUNK)
```

```
        frames.append(data)

    print("* done recording")

    stream.stop_stream()
    stream.close()
    p.terminate()

    wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
    wf.setnchannels(CHANNELS)
    wf.setsampwidth(p.get_sample_size(FORMAT))
    wf.setframerate(RATE)
    wf.writeframes(b''.join(frames))
    wf.close()

def recordAudio():
    all = []
    p = pyaudio.PyAudio()
    WAVE_OUTPUT_FILENAME = "output.wav"
    stream = p.open(format=FORMAT,
                channels=CHANNELS,
                rate=RATE,
                input=True,
                frames_per_buffer=CHUNK)

    try:
        while True:
            data = stream.read(CHUNK)
            all.append(data)
    except KeyboardInterrupt:
        stream.stop_stream()
        stream.close()
        p.terminate()
        wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
        wf.setnchannels(CHANNELS)
        wf.setsampwidth(p.get_sample_size(FORMAT))
        wf.setframerate(RATE)
        wf.writeframes(b''.join(all))
        wf.close()
    print ("* done recording")

def saytts(content):
    tts = gTTS(text=content, lang='en')
    tts.save("tts.mp3")
    sound = AudioSegment.from_mp3("tts.mp3")
```

```
   sound.export("tts.wav", format="wav")
   readwavfile()

def displaymenu():
   print("-------------------------------------------------------------
-----------------------------------------")
   print("                                                  S-H-E-L-L")
   print("-------------------------------------------------------------
-----------------------------------------")
   print("Options")
   print("[0] Help")
   print("[1] Make Directory")
   print("[2] Delete Directory")
   print("[3] Create File")
   print("[4] Delete File")
   print("[5] Rename File")
   print("[6] Display File Content")
   print("[7] Edit File")
   print("[8] Switch Between Speech and Keyboard")
   print("[9] List files")
   print("[10] Change Directory")
   print("[11] Current directory")
   print("[12] Find my phone")
   print("[13] Where am I")
   print("[14] Exit")


def keyboardinput():
   displaymenu()
   n= int(input('Choose Command>>:  '))

   if n==0:
       displaymenu()
   elif(n==1):
       x=input('Enter dir name:   ')
       os.mkdir(x)
       #saytts("directory added succesfully")

   elif(n==2):
       x=input('Enter dir name to be deleted:   ')
       try:
           os.rmdir(x)
       except:
           print("Directory doesnt exist")
```

```
    elif(n==3):
        x=input('Enter file-name:   ')
        file=open(x,'w')

    elif(n==4):
        x=input('Enter file-name to be deleted:   ')
        try:
            os.remove(x)
        except:
            print("File doesnt exist")

    elif(n==5):
        x=input('Enter old file-name :   ')
        y=input('Enter new file-name       :   ')
        try:
            os.rename(x,y)
        except:
            print("File doesnt exist")

    elif(n==6):
        x=input('Enter file-name  :   ')
        file=open(x,'r')
        content=file.read()
        print(content)

    elif(n==7):
        x=input('Enter file-name    :   ')
        print('\n'*2)

print("_____________________________________________________________________
_____________________")
        print('\n'*2)
        file=open(x,'w')
        y=input('>')
        file.write(y)
        file.close()

    elif(n==8):
        speechinput()

    elif (n==10):
        x = input("Enter directory name: ")
        txt = os.getcwd()
        directory = txt + "\\" + x
        print(directory)
```

```
        os.chdir(directory)
        print(os.getcwd())

    elif(n==11):
        os.getcwd()

    elif(n==12):
        exit()

def speechinput():
    displaymenu()
    while True:
        time.sleep(2)
        makewavfile()
        cspeech = recognizespeech()
        print(cspeech)

        if(cspeech=="0" or cspeech=="zero" or cspeech=="hello"):
            displaymenu()

        elif(cspeech == "make directory" or cspeech=="one" or
cspeech=="1"):
            print("say  directory name")
            #saytts("say  directory name")
            makewavfile()
            x=recognizespeech()
            os.mkdir(x)
            #saytts("Directory added successfully")
            print("Directory Added successfully!")

        elif(cspeech=="delete directory" or cspeech=="two" or
cspeech=="2"):
            print("say directory name")
            #saytts("say directory name")
            makewavfile()
            x=recognizespeech()
            try:
                os.rmdir(x)
            except:
                #saytts("Directory does not exist")
                print("Directory doesnt exist")

        elif (cspeech=="create file" or cspeech == "3"):
            print("say file name")
            #saytts("say file name")
```

```
            makewavfile()
            x=recognizespeech() + ".txt"
            file=open(x,'w')

        elif (cspeech=="delete file" or cspeech == "4"):
            print("say file name")
            #saytts("say file name")
            makewavfile()
            x=recognizespeech() + ".txt"
            try:
                os.remove(x)
            except:
                #saytts("File does not exist")
                print("File doesnt exist")

        elif (cspeech=="rename file" or cspeech =="5"):
            print("say file name")
            #saytts("say file name")
            makewavfile()
            x=recognizespeech() + ".txt"
            try:
                os.remove(x)
            except:
                #saytts("File does not exist")
                print("File doesnt exist")

        elif(cspeech=="read file" or cspeech == "6"):
            print("say file name")
            #saytts("say file name")
            makewavfile()
            x=recognizespeech() + ".txt"
            file=open(x,'r')
            content=file.read()
            print(content)

        elif(cspeech=="edit file" or cspeech == "7"):
            print("say file name")
            #saytts("say file name")
            makewavfile()
            x=recognizespeech() + ".txt"
            print('\n'*2)

print("_____________________________________________________________________
_____________________")
            print('\n'*2)
```

```
            file=open(x,'w')
            print('Press Crtl+C to stop edititng')
            recordAudio()
            y=recognizespeech()
            file.write(y)
            file.close()

        elif (cspeech=="keyboard" or cspeech=="8"):
            keyboardinput()

        elif(cspeech=="list files" or cspeech=="9" or cspeech=="LS" or
cspeech=="list file"):
            print(os.listdir())

        elif(cspeech=="change directory" or cspeech=="10" or
cspeech=="CD"):
            print("say directory name")
            #saytts("say directory name")
            makewavfile()
            x=recognizespeech()
            txt = os.getcwd()
            directory = txt + "\\" + x
            print(directory)
            os.chdir(directory)
            print(os.getcwd())

        elif(cspeech=="current directory" or cspeech=="11" or
cspeech=="PWD"):
            print(os.getcwd)

        elif(cspeech=="stop" or cspeech=="exit" or cspeech=="14"):
            exit()

        elif(cspeech=="find my phone" or cspeech=="12"):
            account_sid = "AC6b0ba6d168fe1a1430a722175bff1276"
            auth_token = "6d4ebb736d0841eb2cc96c6bde68cb01"
            client = Client(account_sid, auth_token)
            call= client.calls.create(
                    to="+917639993250",
                    from_="+1 469 314 9196",
                    url="http://demo.twilio.com/docs/voice.xml"
            )
            print(call.sid)
            time.sleep(3)
```

```
        elif(cspeech=="location" or cspeech=="where am I" or
cspeech=="13"):
            url="map.html"
            webbrowser.open(url,new=2)
        else:
            #saytts("command not found")
            print("command not found")

def main():
    speechinput()

if __name__ == "__main__":
    main()
```

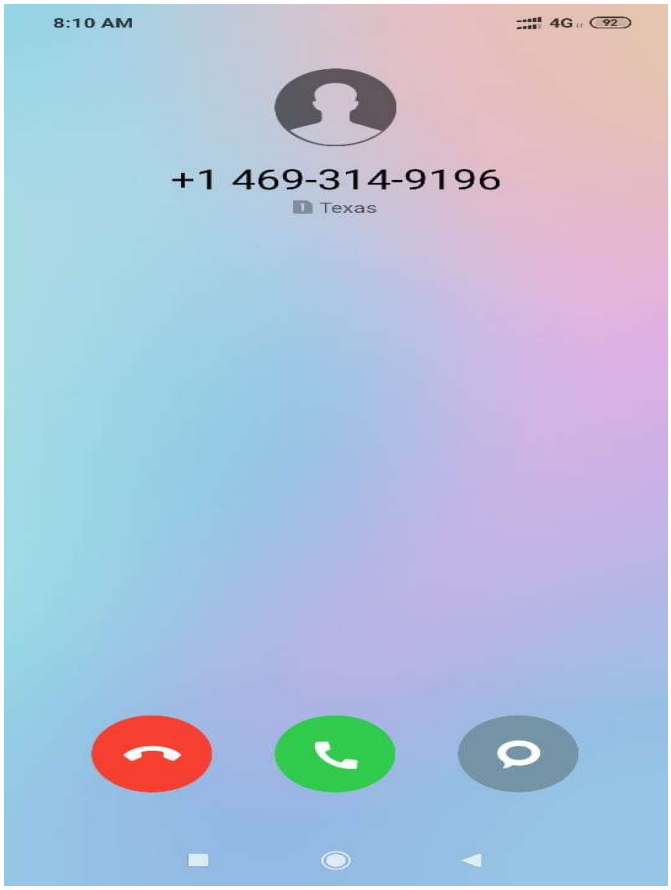
Map.html

```
<!DOCTYPE
html>
            <html>
            <head>
             <meta charset="utf-8">
             <meta http-equiv="X-UA-Compatible" content="IE=edge">
             <title>Location</title>
            </head>
            <body>
```

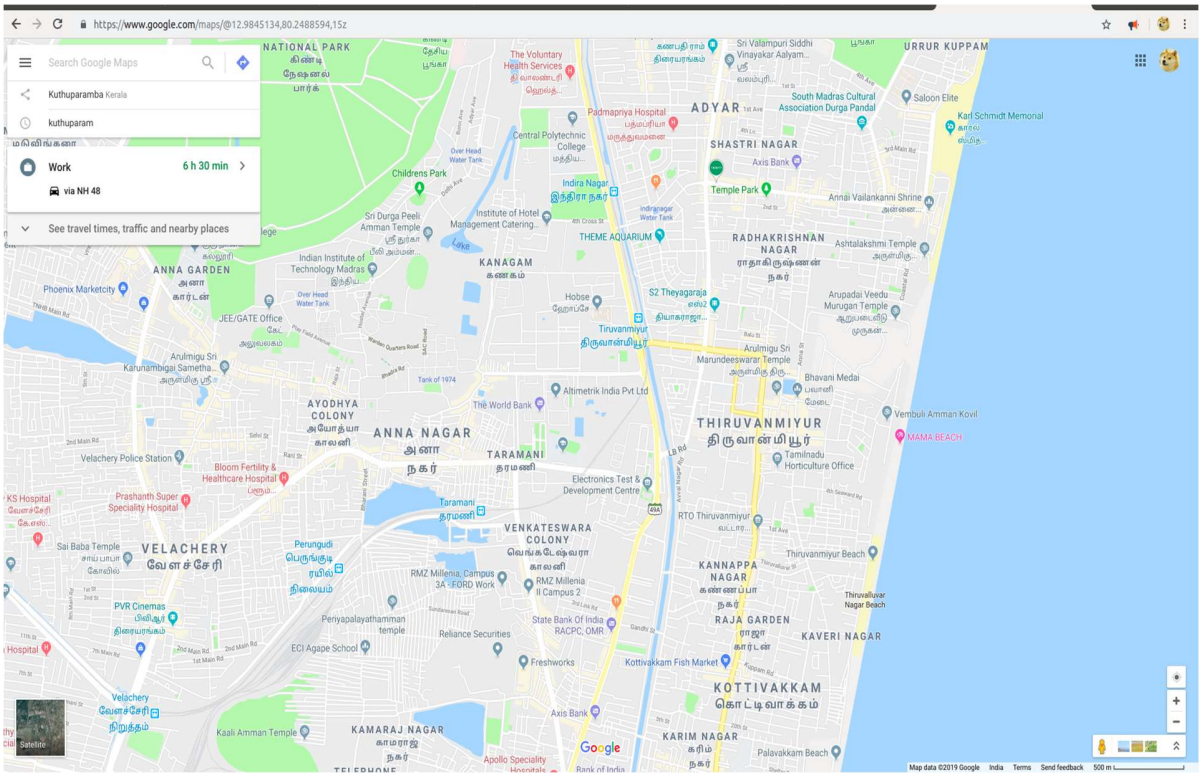
```
<p id="demo"></p>

<script>

   var x = document.getElementById("demo");

   getLocation()

   function getLocation() {

     if (navigator.geolocation) {

       navigator.geolocation.getCurrentPosition(showPosition);

     } else {

       x.innerHTML = "Geolocation is not supported by this browser.";

     }

  function showPosition(position) {

     x.innerHTML =
"https://www.google.com/maps/@"+position.coords.latitude+","+position.c
oords.longitude;

     window.open(x.innerHTML);

   }

</script>

</body>

</html>
```

Result

Contact Protocol using Twilio API



Location tracker using Google Map API


https://www.google.com/maps/@12.9845134,80.2488594,15z
Search Google Maps
Kuthuparamba Kerala
kuthuparam
Work
6 h 30 min
via NH 48
See travel times, traffic and nearby places
NATIONAL PARK
URRUR KUPPAM
ADYAR
SHASTRI NAGAR
RADHAKRISHNAN NAGAR
ANNA GARDEN
KANAGAM
AYODHYA COLONY
ANNA NAGAR
TARAMANI
THIRUVANMIYUR
VELACHERY
VENKATESWARA COLONY
KANNAPPA NAGAR
RAJA GARDEN
KAVERI NAGAR
KOTTIVAKKAM
KARIM NAGAR
KAMARAJ NAGAR
Childrens Park
Temple Park
THEME AQUARIUM
Phoenix Marketcity
JEE/GATE Office
Velachery Police Station
Prashanth Super Speciality Hospital
Bloom Fertility & Healthcare Hospital
Sai Baba Temple
PVR Cinemas
ECI Agape School
Periyapalayathamman temple
Reliance Securities
Freshworks
Kottivakkam Fish Market
Palavakkam Beach
Thiruvanmiyur Beach
Vembuli Amman Kovil
MAMA BEACH
Google
Satellite
Map data ©2019 Google
India
Terms
Send feedback
500 m

Terminal SHELL Menu

```
-----------------------------------------------------------------------------------------------
                                        S-H-E-L-L
-----------------------------------------------------------------------------------------------
Options
[0] Help
[1] Make Directory
[2] Delete Directory
[3] Create File
[4] Delete File
[5] Rename File
[6] Display File Content
[7] Edit File
[8] Switch Between Speech and Keyboard
[9] List files
[10] Change Directory
[11] Current directory
[12] Find my phone
[13] Where am I
[14] Exit
```

Edit File Functionality

References

1) StackOverflow
2) https://www.youtube.com/
3) www.Github.com
4) Geeksforgeeks
5) https://thomas-cokelaer.info/tutorials/python/module_os.html
6) www.Twilio.com
7) https://cloud.google.com/