VOICE COMMAND EXECUTION USING PYTHON

A project report submitted in partial fulfillment of the requirement for degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

C.LOKESH(R141197),
K.PRASAD(R141437),
P.SAITEJA(R141198)

Under the guidance of

Mr. P. Santosh Kumar

Asst. Professor in Department of Computer Science & Engineering

RGUKT RKVALLEY

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES

AP IIIT RGUKT RKVALLEY

KADAPA(Dist), Andhra Pradesh-516330, India

JUL 2019 – NOV 2019

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES
(A.P Government Act 18 0f 2008)
IIIT RKVALLEY, RGUKT – AP
Department of Computer Science & Engineering

CERTIFICATE OF PROJECT COMPLETION

This is to certify that the project entitled **"Voice Command Execution Using Python"** submitted by **C.LOKESH(R141197), K.PRASAD(R141437), P.SAITEJA(R141198)** under our guidance and supervision for the partial fulfillment for the degree of Bachelor of Technology in Computer Science & Engineering during the academic year of July 2019 – November 2019 at AP IIIT RGUKT RKVALLEY.

To the best of our knowledge, the result embodied in this dissertation work have not been submitted to any university or institute for the award of any degree of diploma.

Project Guide	**Project Co-ordinator**
Mr. P. Santosh Kumar, Asst. Professor in Dept. of CSE, AP IIIT RGUKT RKVALLEY	Mr. Ravi Kumar, Asst. Professor in Dept. of CSE, AP IIIT RGUKT RKVALLEY

Head of the Dept.

Mr. N.Chandrasekhar,
HOD & Asst.Prof in Dept. Of CSE,
AP IIIT RGUKT RKVALLEY

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES
(A.P Government Act 18 0f 2008)
IIIT RKVALLEY, RGUKT – AP
Department of Computer Science & Engineering

DECLARATION

We **C.LOKESH(R141197), K.PRASAD(R141437), P.SAITEJA(R141198)** hereby declare that the work which is being presented in this project entitled, **“Voice Command Execution Using Python”** submitted to **RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES, RKVALLEY** in the partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING,** is an authentic record of our own work carried out under the supervision of **Mr. P. Santosh Kumar** in Department of Computer Science and Engineering, RGUKT, RKVALLEY. The matter embodied in this project report has not been submitted by us for the award of any other degree.

C.LOKESH(R141197),
K.PRASAD(R141437),
P.SAITEJA(R141198)

Abstract

To develop a python program that allows the user to perform basic operations like searching files, browsing the web, sending emails, opening files, creating a memo, tweeting tweets, playing entertaining files, etc. To access the available system resources by installing some of the required packages to perform their defined tasks. All the above-mentioned things need to be performed with the use of a program rather than software.

Voice Command Execution Using Python program runs in a terminal or command prompt just like any other program. It uses the terminal as an interactive interface and doesn't need any external interface. The input to our program must be given in the form of speech only. Using the speech recognition module, the given speech can be converted into text and based on its content the command will be executed. The output will be given in both audio and text. For the conversion of audio to text, we need a proper internet connection and the rest of the things based on the command delivered uses the internet connection.

CONTENTS

1. Introduction

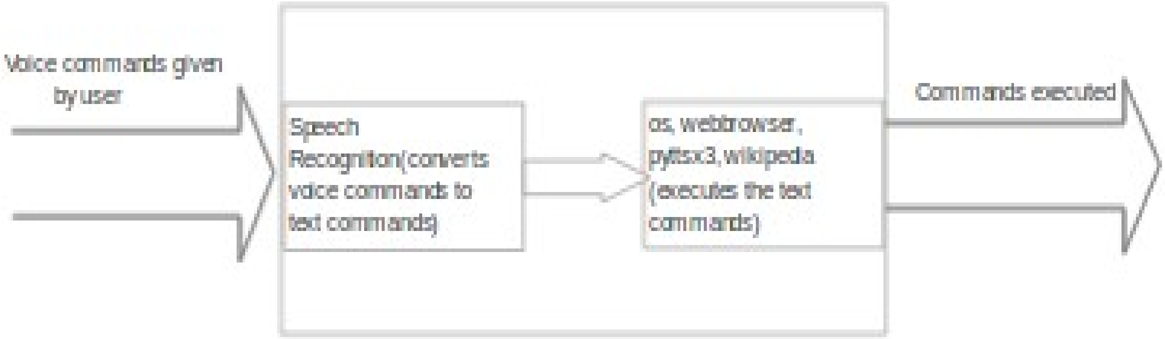
This Voice Command Execution Using program is designed to provide some services with our voice commands instead of accessing the interactive interface physically. Anyone can create the programs for their use based on their necessities. If they have basic programming knowledge of python. Using our program, we can access Wikipedia content with python from the terminal itself. We can also perform major system operations like power off, reboot, etc. We can open entertainment files like audio and video files, etc. We can search for files and directories with the walk method using os package.we can search in any web browser with mentioned sites along with the required query.

We can also post tweets in our twitter accounts by creating a developer account on twitter and using our secured credentials like consumer API key, consumer secret key, OAuth token, etc. We can also make a to-do-list or memo int txt format .we can also send mails to the intended users only with a message. But for this, we need to enable access for fewer security applications from our email account. We can only open files with predefined extensions like mp3, mp4, txt, pdf, jpg, etc. We can also open directories. We can create directories and sub-directories in the path given. For posting our status on facebook we just need our login credentials. For all the above-mentioned things we need a proper internet connection.

Objectives of the project:

- Through this program, anyone can learn how to automate the system with voice commands.
- Operating machine resources through voice.

- It can be executed any operating system which supports python.
- We can access any file in our system without traversing physically.
- We can add any non-existing service/operation we wish to use our system effective.


Voice commands given by user
Speech Recognition(converts voice commands to text commands)
os, webbrowser, pyttsx3, wikipedia (executes the text commands)
Commands executed

BLOCK DIAGRAM

1.1 Existing System

Already existing softwares like Simon, Julius, etc. need training data and they can execute only the commands which were coded in them during their inception. But all of these softwares need the installation of some other packages for their full-pledged use. Thus, makes their installation procedure hefty.

1.2 Proposed System

Our proposed system is not software like Simon, Julius, etc. It is just a python program that needs some basic packages for speech conversion and utilizing system resources. But all these installations are pretty easy when compared to the installation of those softwares. As it is just a python program, it can be executed in any environment which supports python execution. This program can be made and run by anyone who have some basic knowledge of python programming. In python, we have a lot of built-in functions for several modules.

This program can read the speech given by the user for a stipulated amount of time relying on user choice, then the given speech is initially converted into signals and then to text. All these things can be done with the usage of the module speech recognition. For converting this speech to text, speech recognition has several engines and APIs, available in both online and offline. As the offline mode is not accurate and precise, we prefer online mode. We are using Operating System(OS) package, to perform the operations which can be of any system-related type like processes, threads, event handlers, etc., We are using Wikipedia package to directly browse the stuff from the terminal itself.

The ultimate goal of this project is to provide an interactive interface to the user in the terminal prompt itself to communicate with the system and to access all the available system information in one place. Previous versions of existing operating systems like mint, ubuntu, etc. once had the voice command executing softwares. Later they were removed due to their poor portability, lack of latest technology support, compatibility, etc. But for our program, there were no such constraints because python is a high-level language that is developing by a lot of developers continuously.

2. System Environment

2.1 Hardware Configuration

Processor	:	Intel i5
RAM	:	4GB
Hard Disk	:	10GB

2.2 Software Configuration

Operating System	:	Ubuntu 18.10
Language	:	Python
Packages	:	speech recognition, pyttsx3, os, wikipedia, webbrowser, smtplib, ssl, twitter

3. Importance and working mechanism of various packages

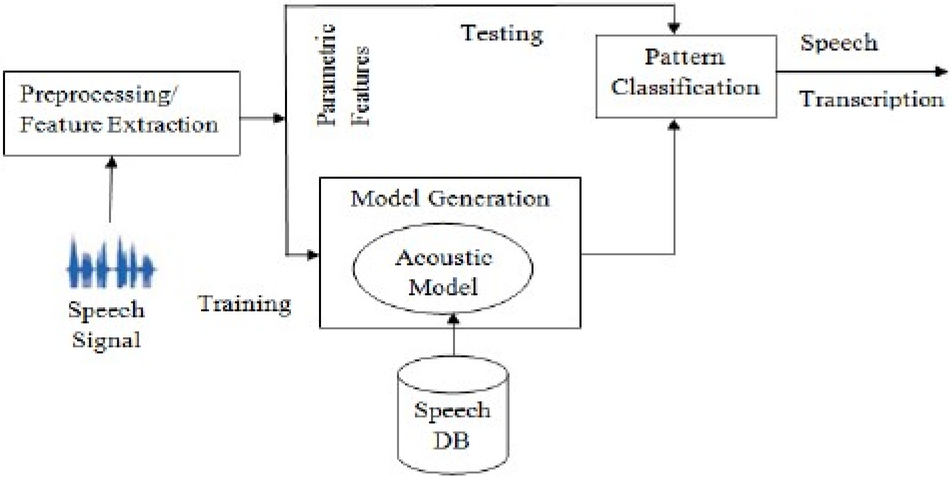
There are a number of packages available for doing a single task. Based on our convenience and ease of use we are choosing one among the existing packages. Out of all, here we are discussing some of the package details along with their working mechanisms.

3.1 Speech Recognition:

In general, speech can be termed as the expression of thoughts and feelings by articulate sounds. Recognizing the articulated sound is called speech recognition. It allows the elderly, physically and visually impaired to interact with several products and services quickly and naturally without GUI. In initial stages, we were accessing the computers with the use of commands, after that with the use of a graphical user interface. Now, in this modern world mostly no one is interested accessing anything using hands. This is because of the effects of voice assistants like Alexa, Siri, Ok/Hey Google, etc., But these things have not entered the market with that much ease. There is a lot of effort from numerous scientists who worked hard for this. In 1952 three Bell Labs Researchers namely Stephen Balashek, R.Biddulph, and K.H.Davis for single-speaker digit recognition built a system called “Audrey”. Thereafter several scientists worked on its extensions. But these systems require users to pause after each word. Later Professor Dabbala Rajagopal Reddy (Turing Award Winner and RGUKT former Chancellor) was the first person who succeeded in continuous speech recognition.

Working Mechanism:

Communication using phonetic combinations of vowel and consonant sounds that form the sound of its words is called speech. The package speech recognition is used for recognizing the words spoken by the user. In the speech recognition package, we use an instance of Recognizer to deal with a variety of settings and functionality for recognizing speech from an audio source to recognize speech. We can create a Recognizer instance only with the help of speech recognition. It has seven methods for recognizing speech from an audio source using various APIs. We create another instance for Microphone class similar to Recognizer instance. A microphone instance is used to take input from the user using Mic. After the instantiation of Recognizer and Microphone classes, we need to use the method listen( ) in the Microphone class for capturing input inside of the block with. It records input from the source until silence is detected by taking an audio source as its argument. To handle ambient noise, you will need to use the adjust_for_ambient_noise( ) method of the Recognizer class. As the input is from a microphone, it is far less predictable than input from an audio file. So it is a good idea to do this anytime you listen for microphone input. At the time of execution, the method adjust_for_ambient_noise( ) waits for a second to do its work and analyzes the audio source by one-second default. We can adjust it with the use of keyword argument duration. We have seven APIs for each Recognizer instance for recognizing speech from an audio source. Out of all, Google Web Speech API can be used with ease because of SpeechRecognition ships with a default API key for this API. We must pass the audio_data as an argument to it and the remaining arguments like language, key and show_all are optional. Since there is a possibility of getting UnknownValueError or RequestError as exceptions, we do this in the try block. Thus, if everything goes good we will get the text output using the speech input.



Working mechanism of speech recognition

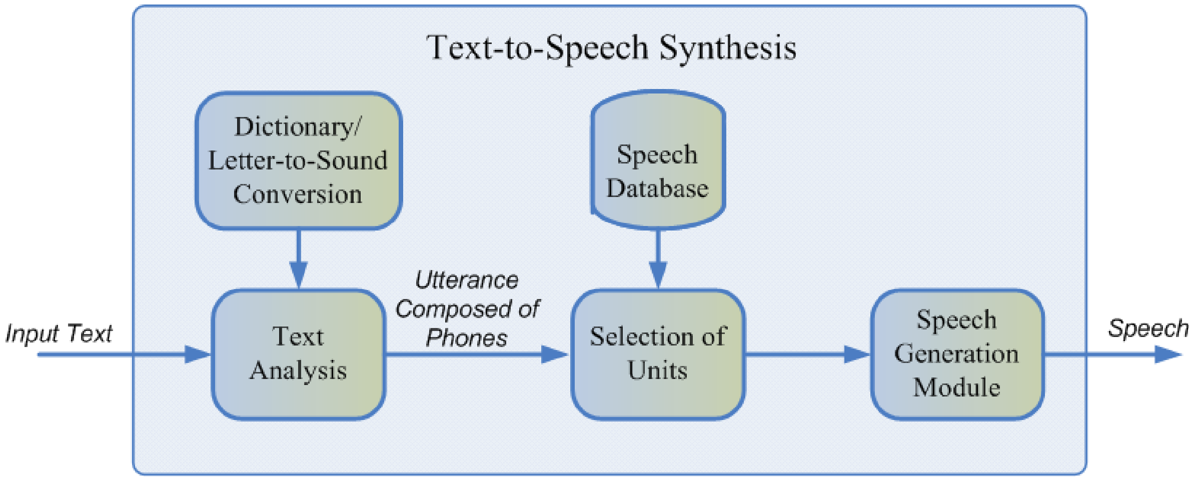
3.2 pyttsx3:

We have several APIs for converting text to speech. The top APIs are Text-to-Speech, Google Cloud Speech, IBM Watson, Microsoft Text Translation, etc., But most of these APIs need internet for their full-pledged use. For this reason, we are using the pyttsx3 package. It stands for python text to speech version 3. This package has several methods, out of which several have options for changing their parameters for our convenient use.

Working mechanism:

For using this package, we need to install it using pip. After that, we need to initialize it using the method init( ). Thereafter for changing any of its parameters like rate, volume, voice, etc., we need to get them first using getProperty( ). The method runAndWait( ) blocks the

execution when processing all the currently queued commands. Callbacks can be invoked for engine notifications appropriately. It returns none and continues the execution when all the queued commands before this call are emptied from the queue. The method say( ) is used for saying the text passed in it as parameter and type of text is an optional parameter. The stop( ) method clears the command queue and stops the current utterance.



Working mechanism of pyttsx3

3.3 OS:

The package OS in python has a lot of functions to communicate and interact with the underlying Operating System in several different ways. It delivers a wide range of effective methods to manipulate files and directories. It performs tasks like checking if certain files or directories exist at a location, doing operations on every file in the directory that satisfies some user-defined criteria, creating or deleting files or directories and much more.

Working mechanism:

The method listdir( ) returns a list containing the names of all the files and directories if the path is provided as an argument. Else present working directory can be taken as the path. The method system( ) executes the command passed in it as an argument in a subshell. Its implementation can be done by calling the Standard C function system( ). The limitations in both python and C are the same. It returns the exit status of the process on Unix. The method walk( ) is used to generate a 3-tuple for each directory in the tree, including the root by walking either top-down or bottom-up. Simply it returns all the filenames and directory names inside a root tree.

3.4 Wikipedia and webbrowser:

The package wikipedia in python library makes it easy to access and parse data from the most popular general reference Wikipedia. It is one of the largest and most popular sources for information. The method search( ) takes a query as an argument. It returns a list of article names containing the query. The method summary( ) takes a query as an argument and extracts the summary related to the query. We can customize the number of sentences in the summary to be displayed. The method page( ) is used to get the plain text of the data.

The package webbrowser is used to shift from the terminal to the browser. The browser can be firefox, chrome, Operamini, etc., The method get( ) is used to get the browser we wish and returns a controller object for the browser type. The method open_new( ) takes URL as an argument and opens the URL in a new window of the browser. The method open_new_tab( ) takes the URL as an argument and opens the URL in a new tab of the browser.

3.5 SMTP and SSL:

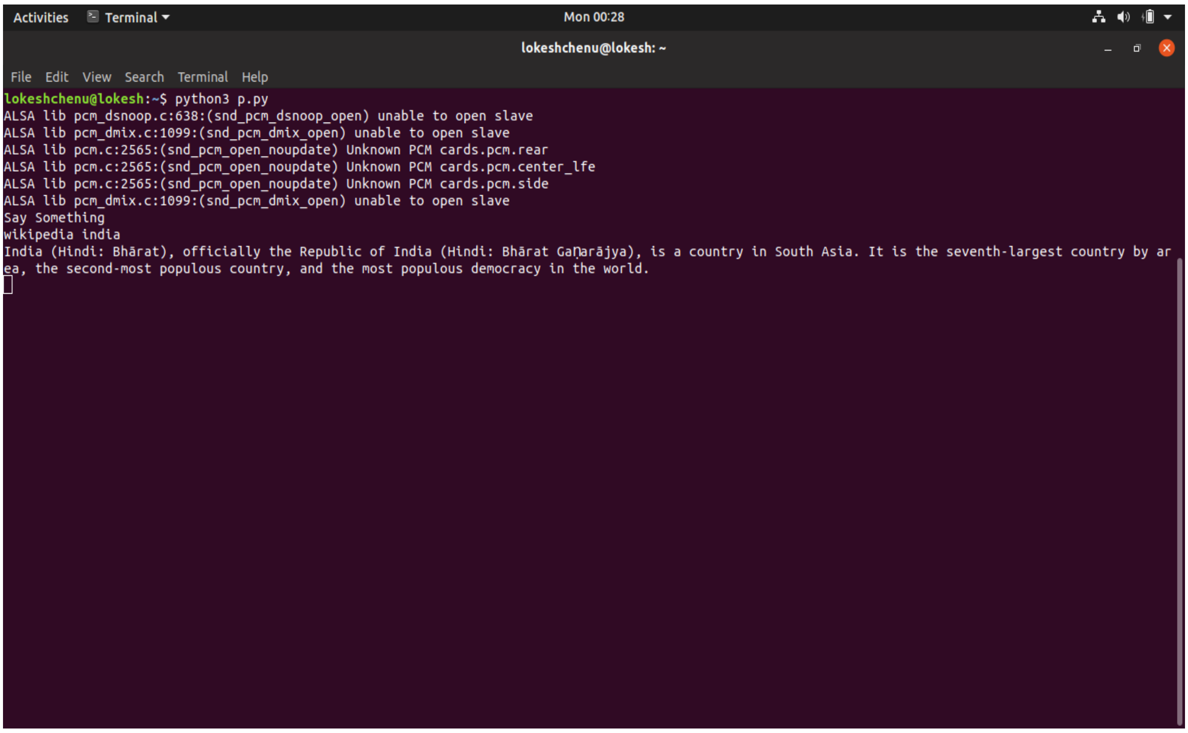
Both SMTP and SSL packages are combinedly used for dealing with e-mails. SMTP stands for Simple Mail Transfer Protocol. It deals with sending and routing e-mail between mail servers. Using smtblib module from python it explicates SMTP client session object using which we can send mail to any Internet machine with SMTP listener daemon. When dealing with SSL it uses the port 465 whereas when using TSL it uses 587. We are using SSL as it is secured from the beginning of an SMTP connection. SSL stands for Secure Socket Layer. It provides peer
authentication facilities for network sockets, both client-side and server-side. SSL module default settings are not appropriate for our application. By using create_default_context() from the SSL module, we will load the system trusted CA certificates which enables hostname
checking and certificate validation. Accessing the mail accounts from less secure applications is not trustworthy, by default it is disabled. So only after enabling access from less secure apps, it is possible to access the mail account from less secure apps.

4. Impementation

```
import pyttsx3
import wikipedia
import datetime
import webbrowser
import os
import smtplib
import ssl
import speech_recognition as sr
from twitter import *
def command():
	r=sr.Recognizer()
	m=sr.Microphone()
	with sr.Microphone() as source:
		print("Say Something")
		r.adjust_for_ambient_noise(source)
		audio=r.listen(source)
	try:
		text=r.recognize_google(audio)
		engine.say(text)
		return text.lower()
	except Exception as e:
		print("Loki, I can't understood what you said")

def reduce(cmd,key):
	cmd=cmd.replace(key,"",1)
	cmd=cmd.strip()
	return cmd
engine = pyttsx3.init()
rate = engine.getProperty('rate')
engine.setProperty('rate',rate-100)
engine.say("Hai loki")

while(1):
	cmd=command()
```

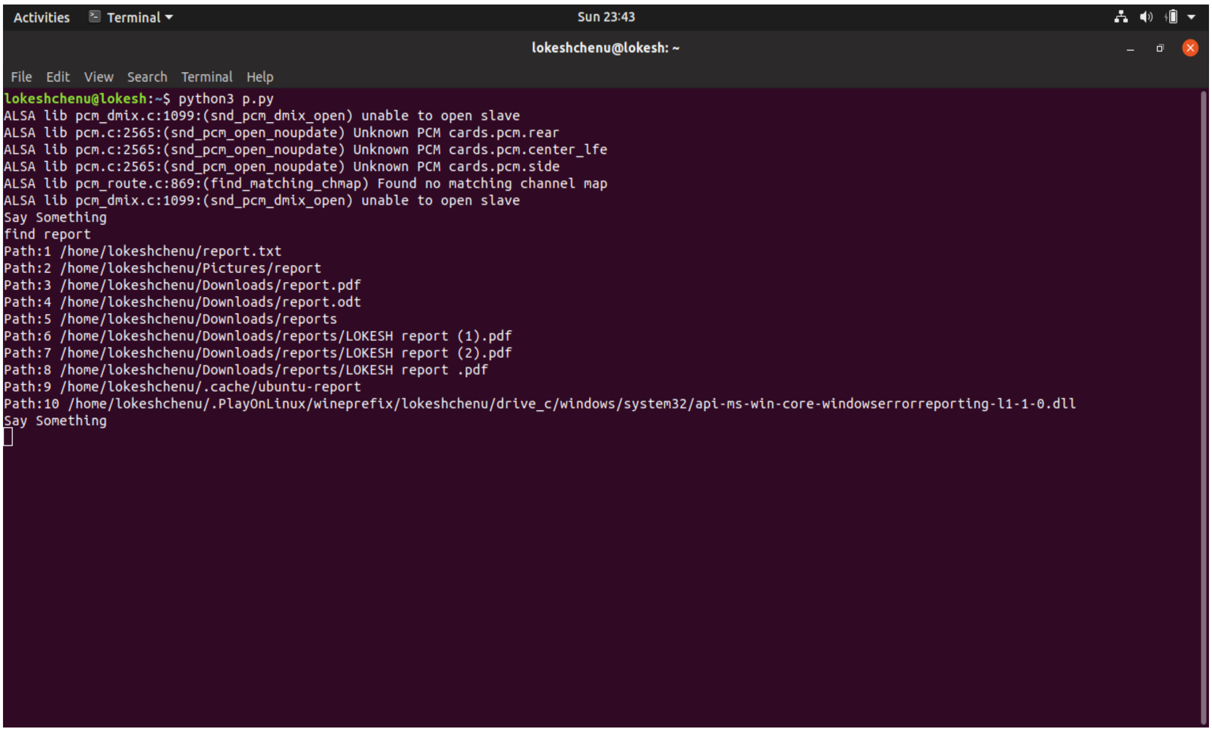
```
if(cmd==None):
        continue
    print(cmd)
    if cmd.startswith('stop'):
        exit()
    elif cmd.startswith('shutdown'):
        os.system('systemctl poweroff')
    elif cmd.startswith('restart'):
        os.system('systemctl reboot')

    copy=cmd.split()
    if(len(copy)>1):
        if cmd.startswith('wikipedia'):
            cmd=reduce(cmd,'wikipedia')
            res=wikipedia.summary(cmd,sentences=2)
            print(res)
            engine.say(res)

        elif cmd.startswith('play music'):
            path ="/home/lokeshchenu/Music/"
            files = os.listdir(path)
            for i in range(len(files)):
                f=path+files[i]
                os.system ('mplayer' + " "+f)

        elif 'find' in cmd:
            cmd=reduce(cmd,'find')
            i=1
            for root, dirs, files in os.walk('/home'):
                for file in files:
                    if cmd in file:
                        print("Path:"+str(i)+"
"+root+'/'+str(file))
                        i=i+1
                if i>10:
                    break
                for file in dirs:
                    if cmd in file:
                        print("Path:"+str(i)+"
"+root+'/'+str(file))
                        i=i+1
                if i>10:
                    break

        elif cmd.startswith('open'):
            cmd=reduce(cmd,'open')
            cmd=cmd.split()
            site=cmd[0]
            try:
               webbrowser.get('firefox')

webbrowser.open_new_tab('http://www.'+site+'.com')
            except:

    webbrowser.open_new('http://www.'+site+'.com')

        elif cmd.startswith('hummingbird'):
            cmd=reduce(cmd,'hummingbird')
            t = Twitter(auth=OAuth('34314692112-
yaD0nk8OwhfV5d4GpeBNaIpZfK8TaaooLdarhoap','ghqZ5ssDV7ZC
jCP6xb8C0BAmEWAaTOVszuyLteDsdWBQfMy' ,
'0aIyApxfRjUgAB9irK2HFqbfJc',
'gEOzqfLQ1QAqzeRzsgnOUV6sj2g7MpSbAhYQDk8gcHSvllW82iB
'))
            t.statuses.update(status=cmd)

    if(len(copy)>2):
        if cmd.startswith('note'):
            cmd=reduce(cmd,'note')
            cmd1=cmd
            cmd=cmd.split()
            cmd1=reduce(cmd1,cmd[0])
            fname=cmd[0]+".txt"
            f=open(fname,"w+")
            f.write(cmd1)
            f.close()
            f=open(fname,"r")
            file_contents=f.read()
            print(file_contents)
            f.close()

        elif cmd.startswith('search'):
            cmd=reduce(cmd,'search')
            cmd1=cmd
            cmd=cmd.split()
            cmd1=reduce(cmd1,cmd[0])
            site=cmd[0]
            try:
                webbrowser.get('firefox')

webbrowser.open_new_tab("https://"+site+".com/search?q=%s" %
cmd1)
            except:

    webbrowser.open_new("https://"+site+".com/search?q=%s" %
cmd1)

        elif cmd.startswith('email'):
            cmd=reduce(cmd,'email')
            cmd1=cmd
            cmd=cmd.split()
            cmd1=reduce(cmd1,cmd[0])
            user=cmd[0]


m_l=["r141437@rguktrkv.ac.in","r141198@rguktrkv.ac.in","r141291
@rguktrkv.ac.in"]
            receiver_mail=""
            if(user=="prasad"):
                receiver_mail=m_l[0]
            elif(user=="hari"):
                receiver_mail=m_l[1]
            elif(user=="sai"):
                receiver_mail=m_l[2]

            if(receiver_mail != "" ):
                context = ssl.create_default_context()
                with smtplib.SMTP_SSL("smtp.gmail.com",
465, context=context) as server:

    server.login("lokeshchenu7198@gmail.com", "xxxxxxxx")

    server.sendmail("lokeshchenu7198@gmail.com", receiver_mail,
cmd1)
    engine.runAndWait()
engine.stop()
```

5. Output (Screenshots)



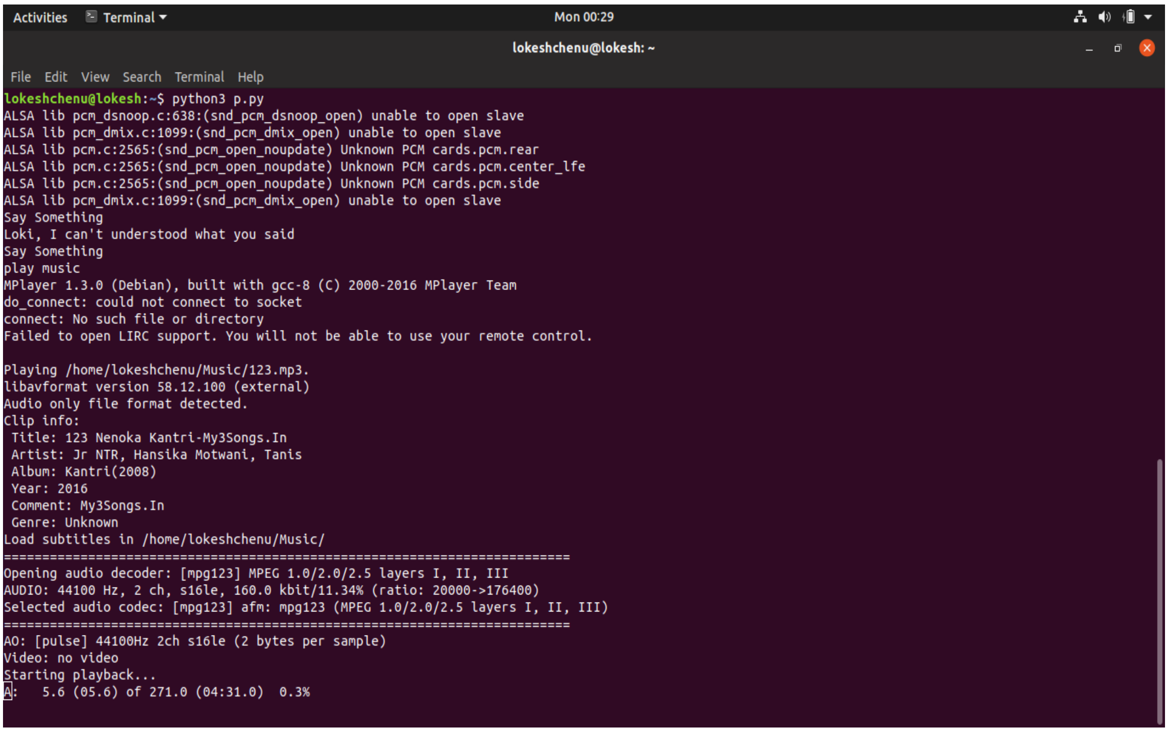
```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
wikipedia india
India (Hindi: Bhārat), officially the Republic of India (Hindi: Bhārat Gaṇarājya), is a country in South Asia. It is the seventh-largest country by ar
ea, the second-most populous country, and the most populous democracy in the world.
```

Figure 5.1 Wikipedia search



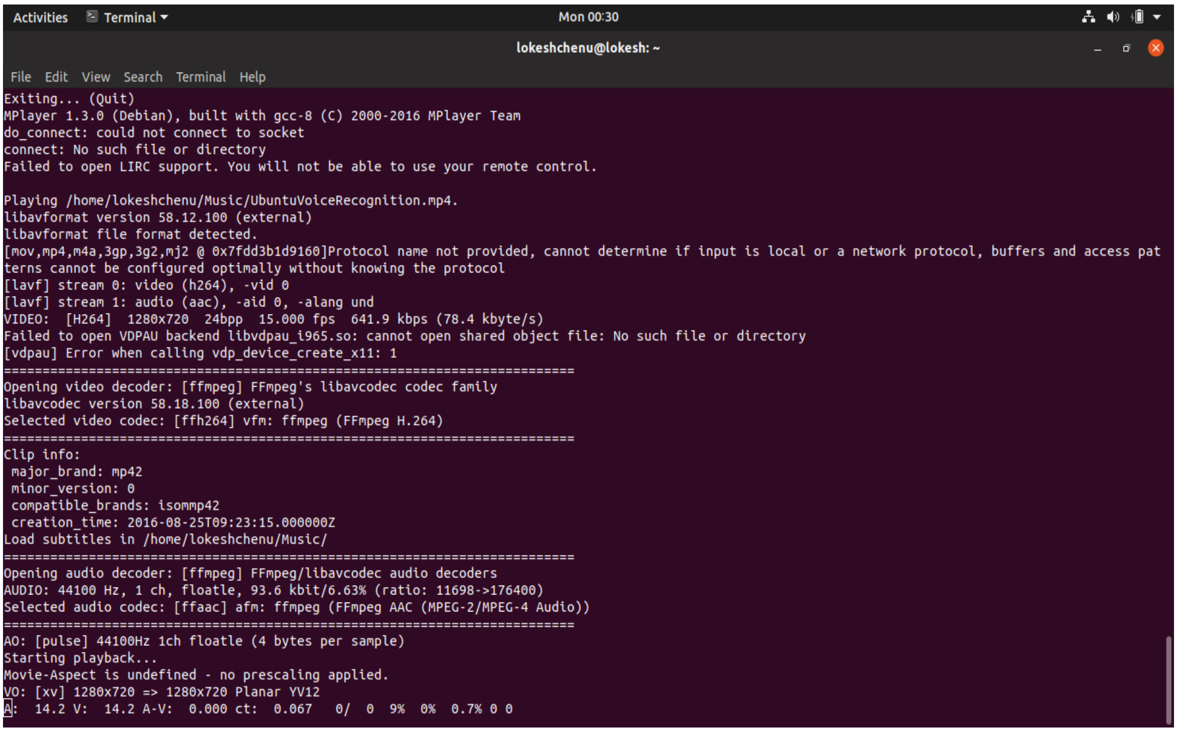
```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:869:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
find report
Path:1 /home/lokeshchenu/report.txt
Path:2 /home/lokeshchenu/Pictures/report
Path:3 /home/lokeshchenu/Downloads/report.pdf
Path:4 /home/lokeshchenu/Downloads/report.odt
Path:5 /home/lokeshchenu/Downloads/reports
Path:6 /home/lokeshchenu/Downloads/reports/LOKESH report (1).pdf
Path:7 /home/lokeshchenu/Downloads/reports/LOKESH report (2).pdf
Path:8 /home/lokeshchenu/Downloads/reports/LOKESH report .pdf
Path:9 /home/lokeshchenu/.cache/ubuntu-report
Path:10 /home/lokeshchenu/.PlayOnLinux/wineprefix/lokeshchenu/drive_c/windows/system32/api-ms-win-core-windowserrorreporting-l1-1-0.dll
Say Something
```

Figure 5.2 File Search



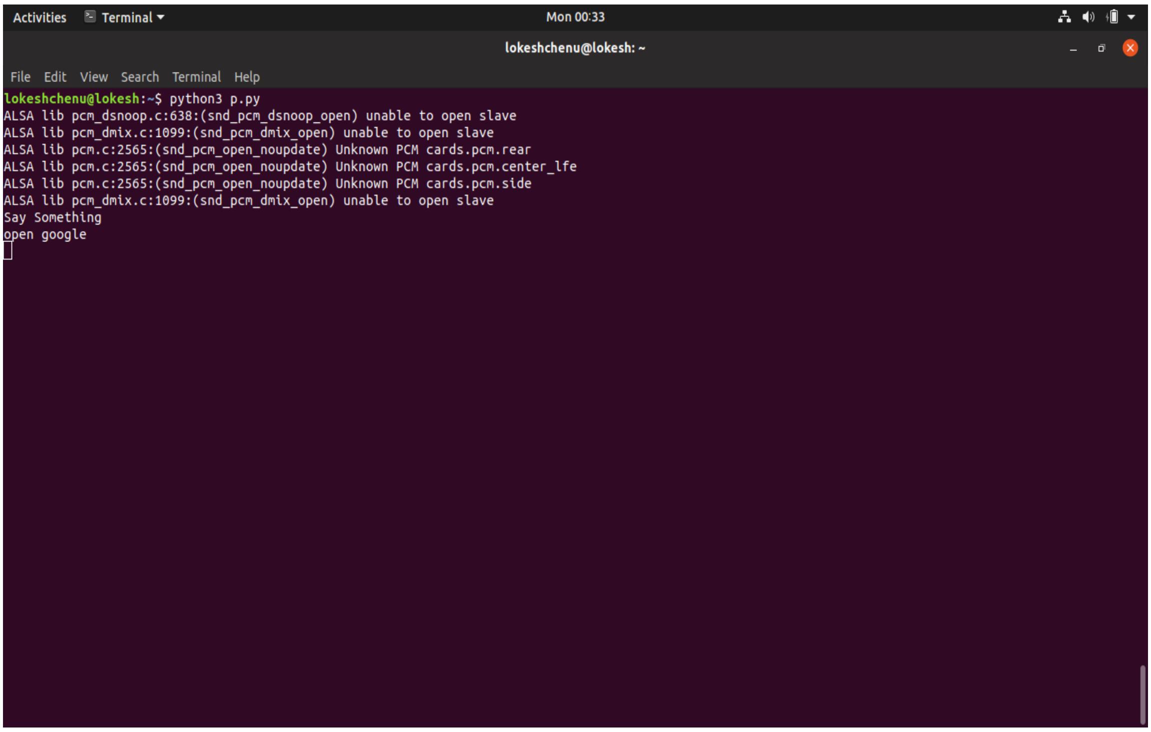
```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
Loki, I can't understood what you said
Say Something
play music
MPlayer 1.3.0 (Debian), built with gcc-8 (C) 2000-2016 MPlayer Team
do_connect: could not connect to socket
connect: No such file or directory
Failed to open LIRC support. You will not be able to use your remote control.

Playing /home/lokeshchenu/Music/123.mp3.
libavformat version 58.12.100 (external)
Audio only file format detected.
Clip info:
 Title: 123 Nenoka Kantri-My3Songs.In
 Artist: Jr NTR, Hansika Motwani, Tanis
 Album: Kantri(2008)
 Year: 2016
 Comment: My3Songs.In
 Genre: Unknown
Load subtitles in /home/lokeshchenu/Music/
==========================================================================
Opening audio decoder: [mpg123] MPEG 1.0/2.0/2.5 layers I, II, III
AUDIO: 44100 Hz, 2 ch, s16le, 160.0 kbit/11.34% (ratio: 20000->176400)
Selected audio codec: [mpg123] afm: mpg123 (MPEG 1.0/2.0/2.5 layers I, II, III)
==========================================================================
AO: [pulse] 44100Hz 2ch s16le (2 bytes per sample)
Video: no video
Starting playback...
A:   5.6 (05.6) of 271.0 (04:31.0)  0.3%
```

Figure 5.3.1 Playing musis files



```
Exiting... (Quit)
MPlayer 1.3.0 (Debian), built with gcc-8 (C) 2000-2016 MPlayer Team
do_connect: could not connect to socket
connect: No such file or directory
Failed to open LIRC support. You will not be able to use your remote control.

Playing /home/lokeshchenu/Music/UbuntuVoiceRecognition.mp4.
libavformat version 58.12.100 (external)
libavformat file format detected.
[mov,mp4,m4a,3gp,3g2,mj2 @ 0x7fdd3b1d9160]Protocol name not provided, cannot determine if input is local or a network protocol, buffers and access pat
terns cannot be configured optimally without knowing the protocol
[lavf] stream 0: video (h264), -vid 0
[lavf] stream 1: audio (aac), -aid 0, -alang und
VIDEO:  [H264]  1280x720  24bpp  15.000 fps  641.9 kbps (78.4 kbyte/s)
Failed to open VDPAU backend libvdpau_i965.so: cannot open shared object file: No such file or directory
[vdpau] Error when calling vdp_device_create_x11: 1
==========================================================================
Opening video decoder: [ffmpeg] FFmpeg's libavcodec codec family
libavcodec version 58.18.100 (external)
Selected video codec: [ffh264] vfm: ffmpeg (FFmpeg H.264)
==========================================================================
Clip info:
 major_brand: mp42
 minor_version: 0
 compatible_brands: isommp42
 creation_time: 2016-08-25T09:23:15.000000Z
Load subtitles in /home/lokeshchenu/Music/
==========================================================================
Opening audio decoder: [ffmpeg] FFmpeg/libavcodec audio decoders
AUDIO: 44100 Hz, 1 ch, floatle, 93.6 kbit/6.63% (ratio: 11698->176400)
Selected audio codec: [ffaac] afm: ffmpeg (FFmpeg AAC (MPEG-2/MPEG-4 Audio))
==========================================================================
AO: [pulse] 44100Hz 1ch floatle (4 bytes per sample)
Starting playback...
Movie-Aspect is undefined - no prescaling applied.
VO: [xv] 1280x720 => 1280x720 Planar YV12
A:  14.2 V:  14.2 A-V:  0.000 ct:  0.067   0/  0  9%  0%  0.7% 0 0
```

Figure 5.3.2 Playing video files



```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
open google
```

Figure 5.4.1 Opening websites

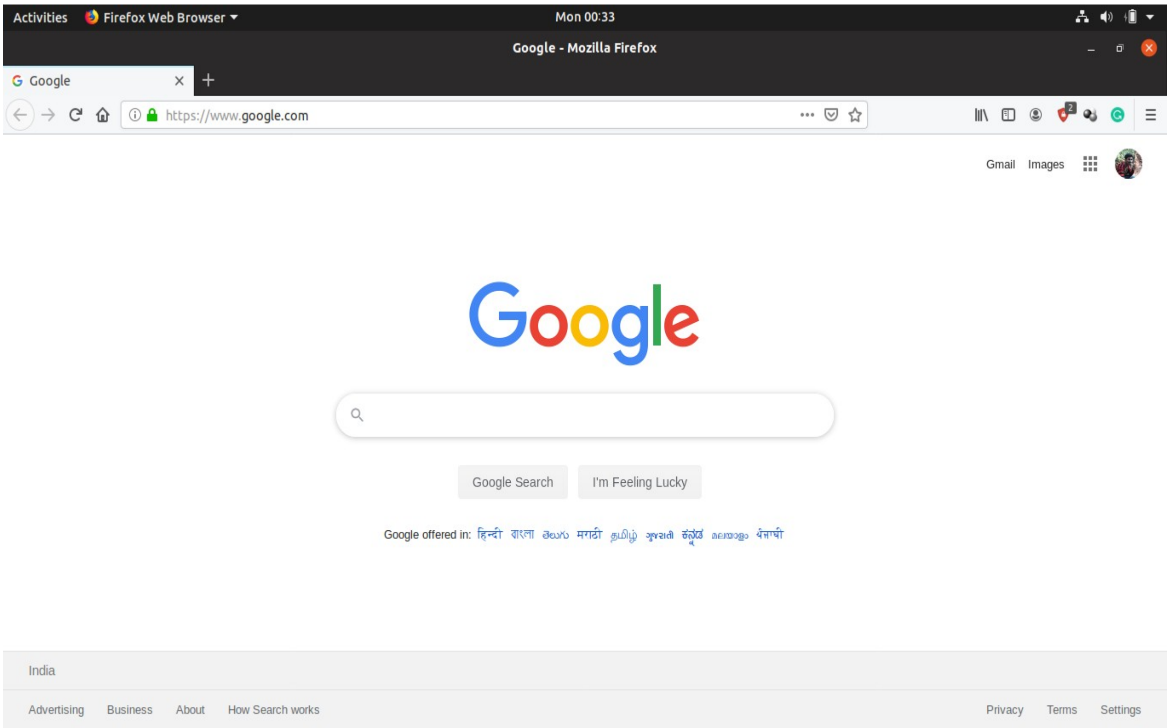
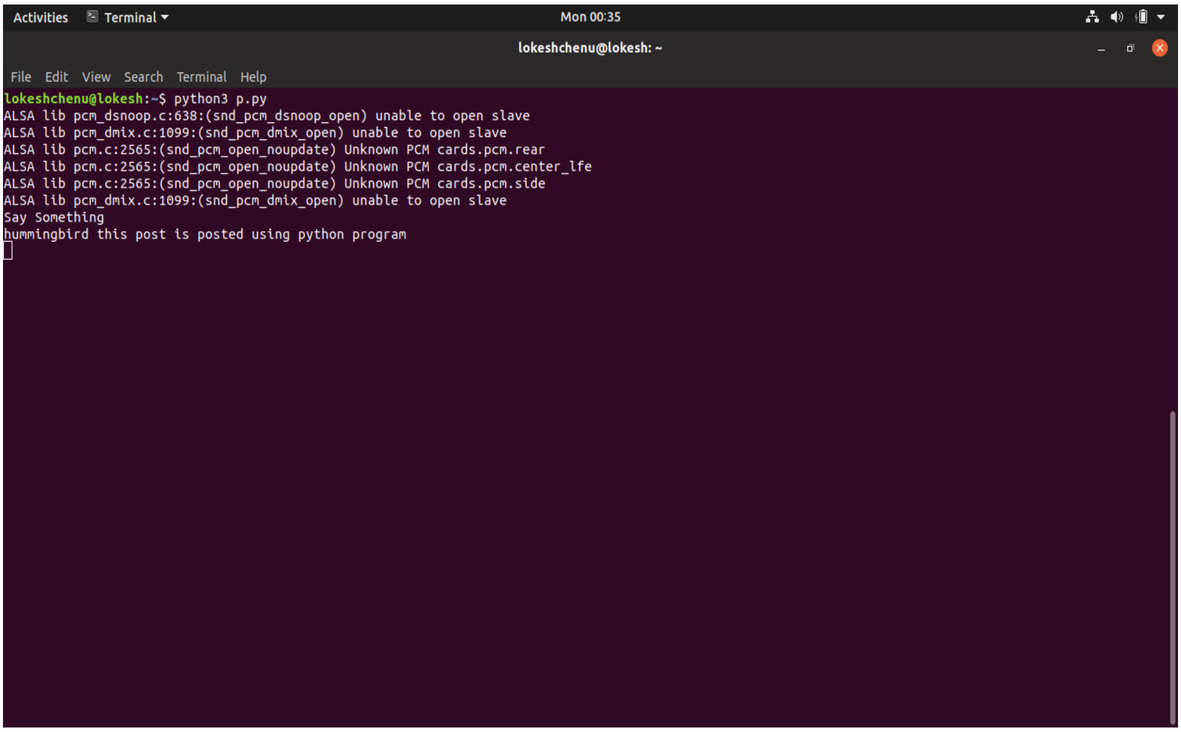
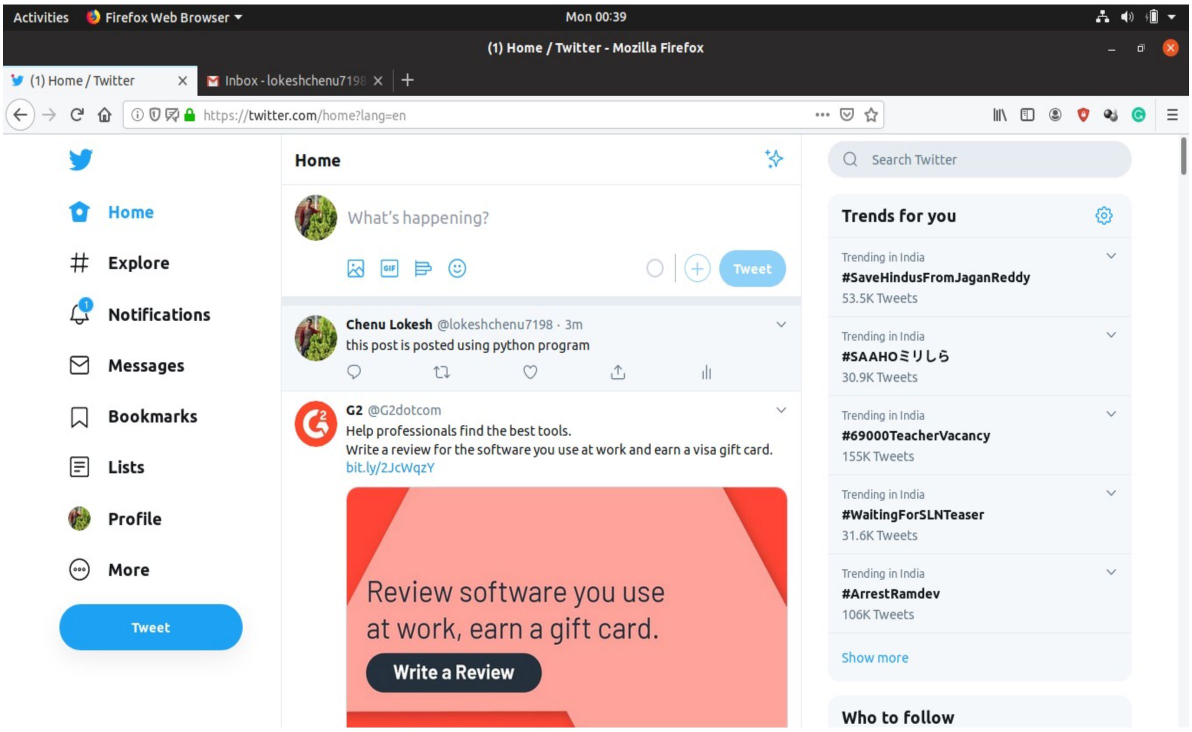


Figure 5.4.2 Result page google site



```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
hummingbird this post is posted using python program
```

Figure 5.5.1 Twitter Post



Chenu Lokesh @lokeshchenu7198 · 3m
this post is posted using python program

G2 @G2dotcom
Help professionals find the best tools.
Write a review for the software you use at work and earn a visa gift card.
bit.ly/2JcWqzY

Review software you use at work, earn a gift card.
Write a Review

Trending in India
#SaveHindusFromJaganReddy
53.5K Tweets

Trending in India
#SAAHOミリしら
30.9K Tweets

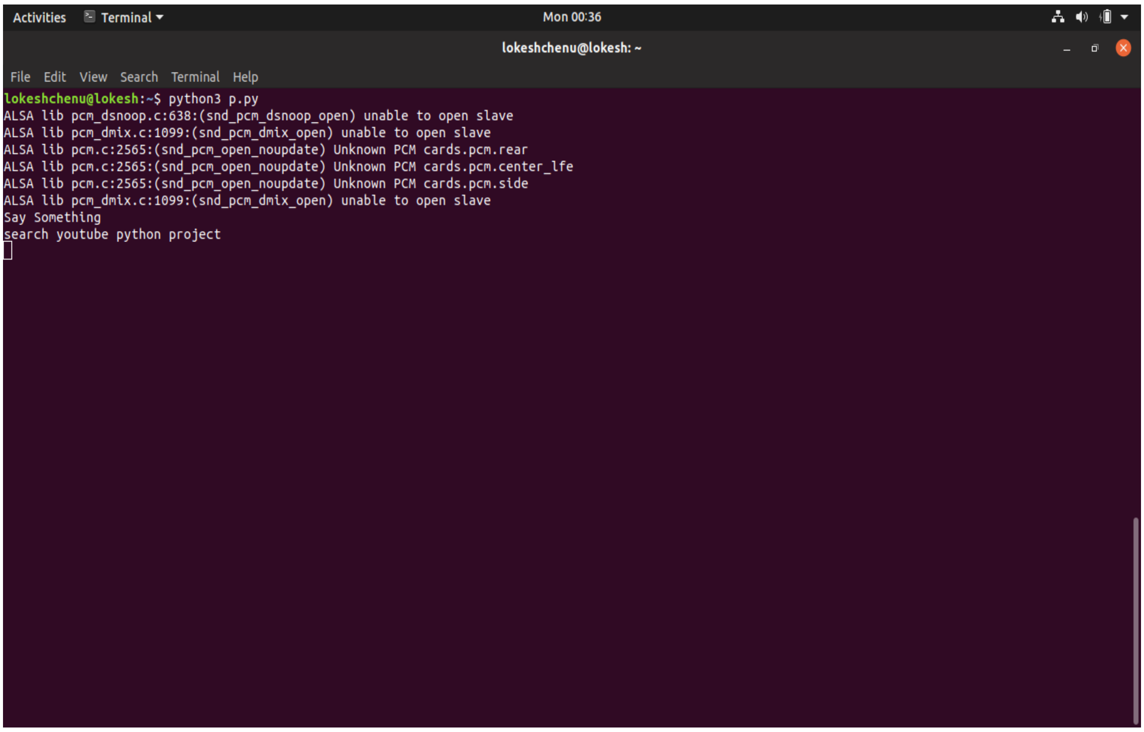
Trending in India
#69000TeacherVacancy
155K Tweets

Trending in India
#WaitingForSLNTeaser
31.6K Tweets

Trending in India
#ArrestRamdev
106K Tweets

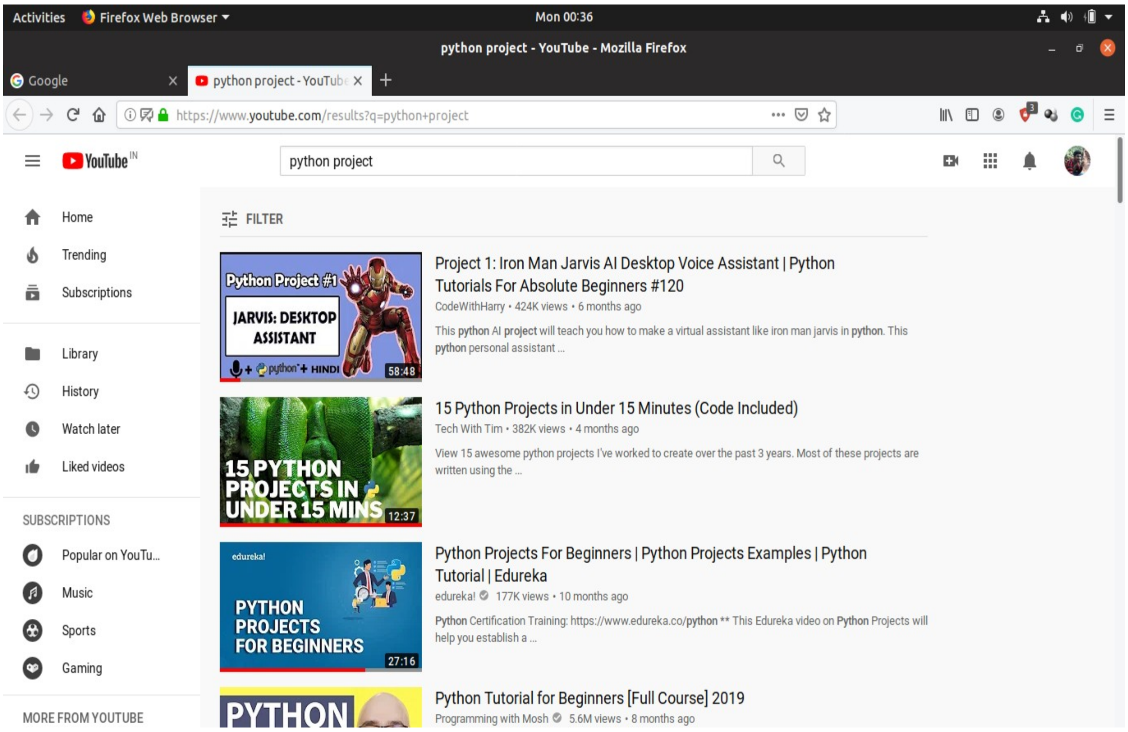
Show more

Figure 5.5.2 Result page Twitter



```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
search youtube python project
```

Figure 5.6.1 Searching website with query



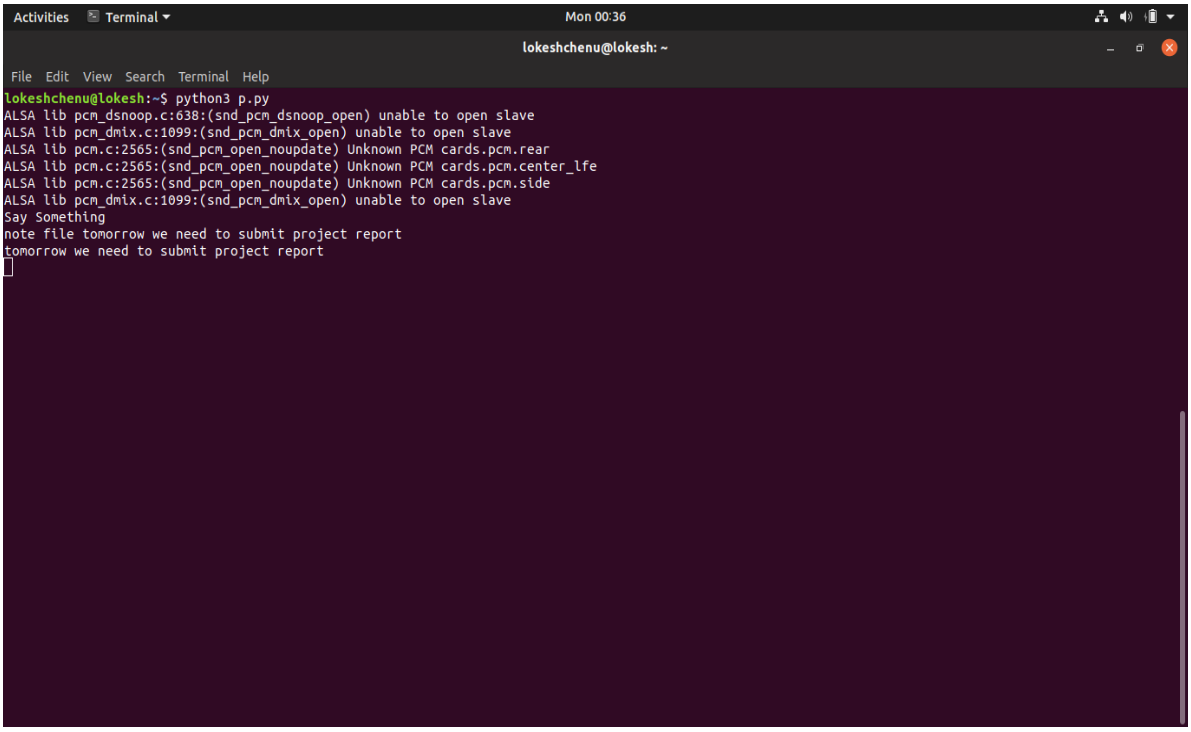
Project 1: Iron Man Jarvis AI Desktop Voice Assistant | Python Tutorials For Absolute Beginners #120
CodeWithHarry • 424K views • 6 months ago
This python AI project will teach you how to make a virtual assistant like iron man jarvis in python. This python personal assistant ...

15 Python Projects in Under 15 Minutes (Code Included)
Tech With Tim • 382K views • 4 months ago
View 15 awesome python projects I've worked to create over the past 3 years. Most of these projects are written using the ...

Python Projects For Beginners | Python Projects Examples | Python Tutorial | Edureka
edureka! • 177K views • 10 months ago
Python Certification Training: https://www.edureka.co/python ** This Edureka video on Python Projects will help you establish a ...

Python Tutorial for Beginners [Full Course] 2019
Programming with Mosh • 5.6M views • 8 months ago

Figure 5.6.2 Result page youtube with query



```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
note file tomorrow we need to submit project report
tomorrow we need to submit project report
```

Figure 5.7.1 Creating a to-do-list or memo

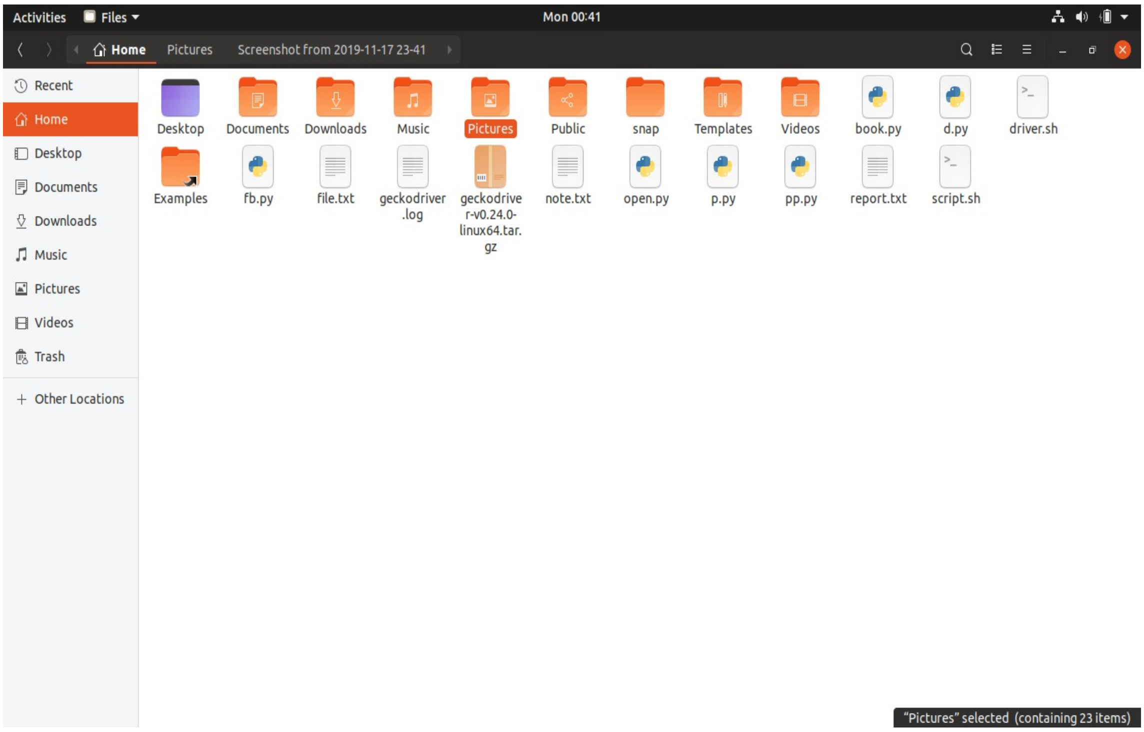
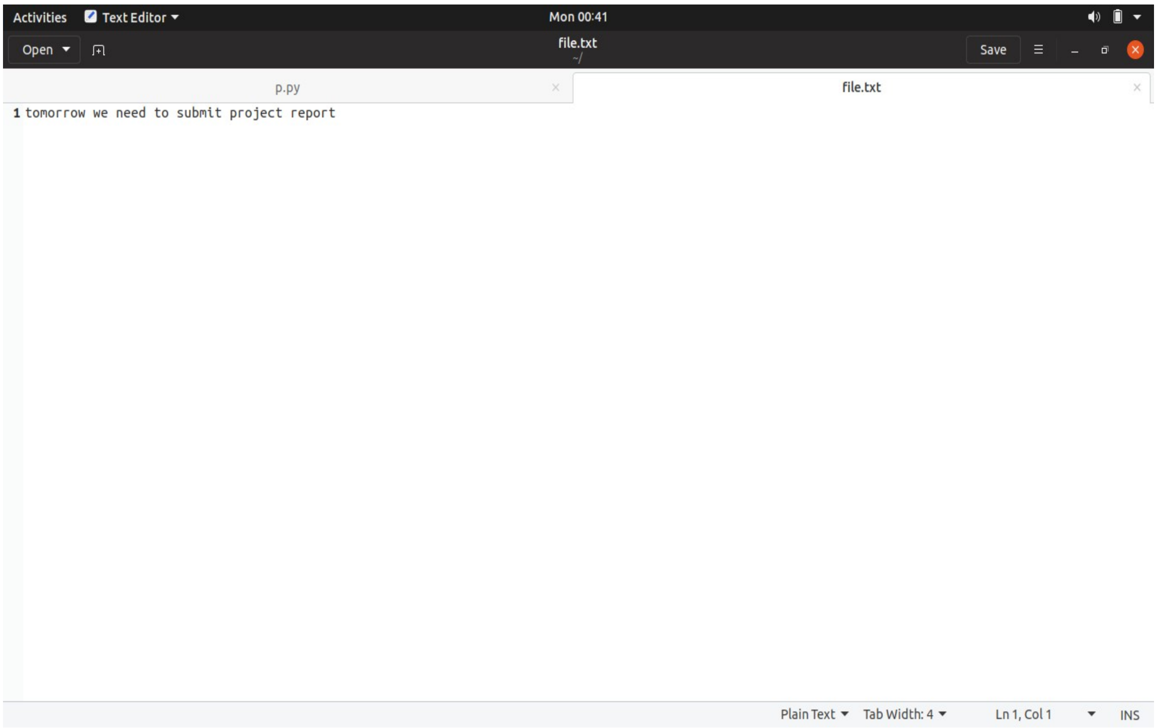
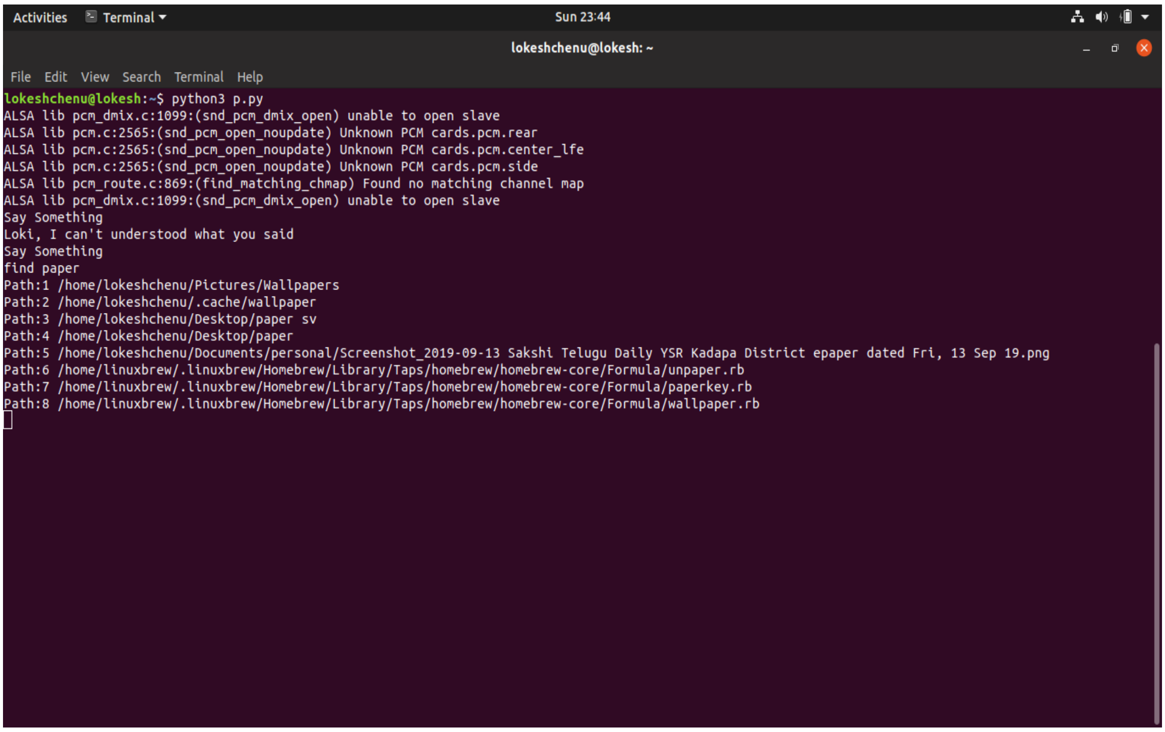


Figure 5.7.2 Directory showing file with name file.txt



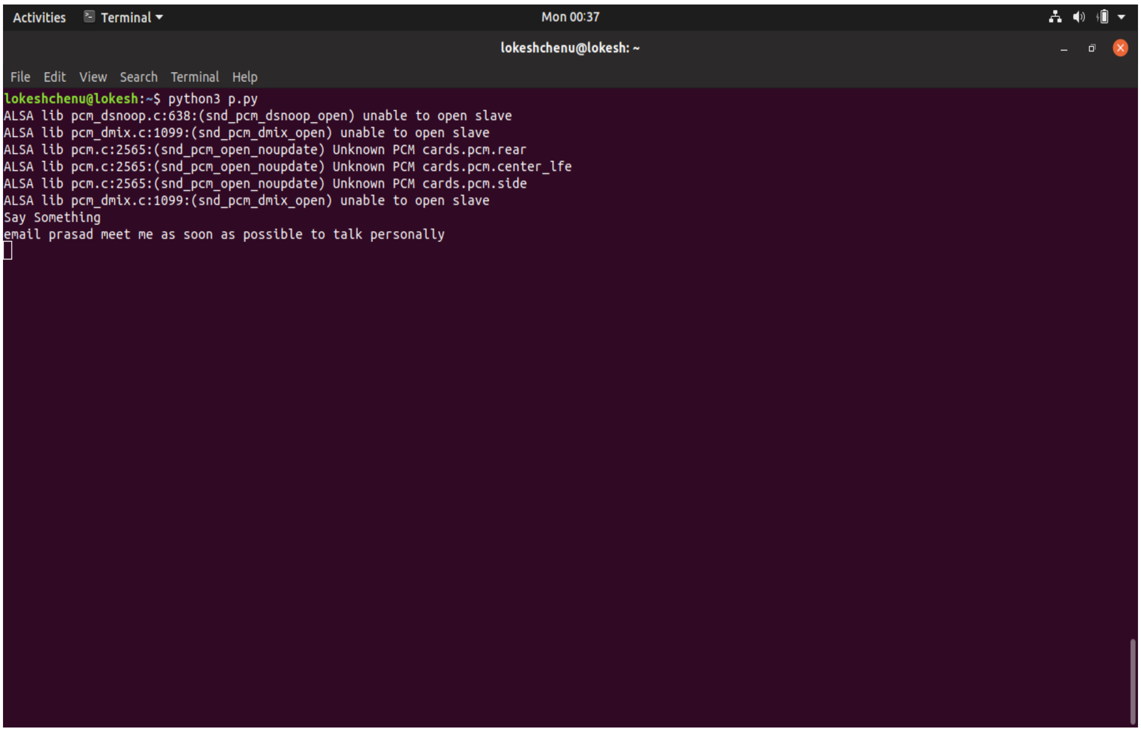
```
tomorrow we need to submit project report
```

Figure 5.7.2 File showing contents of file.txt



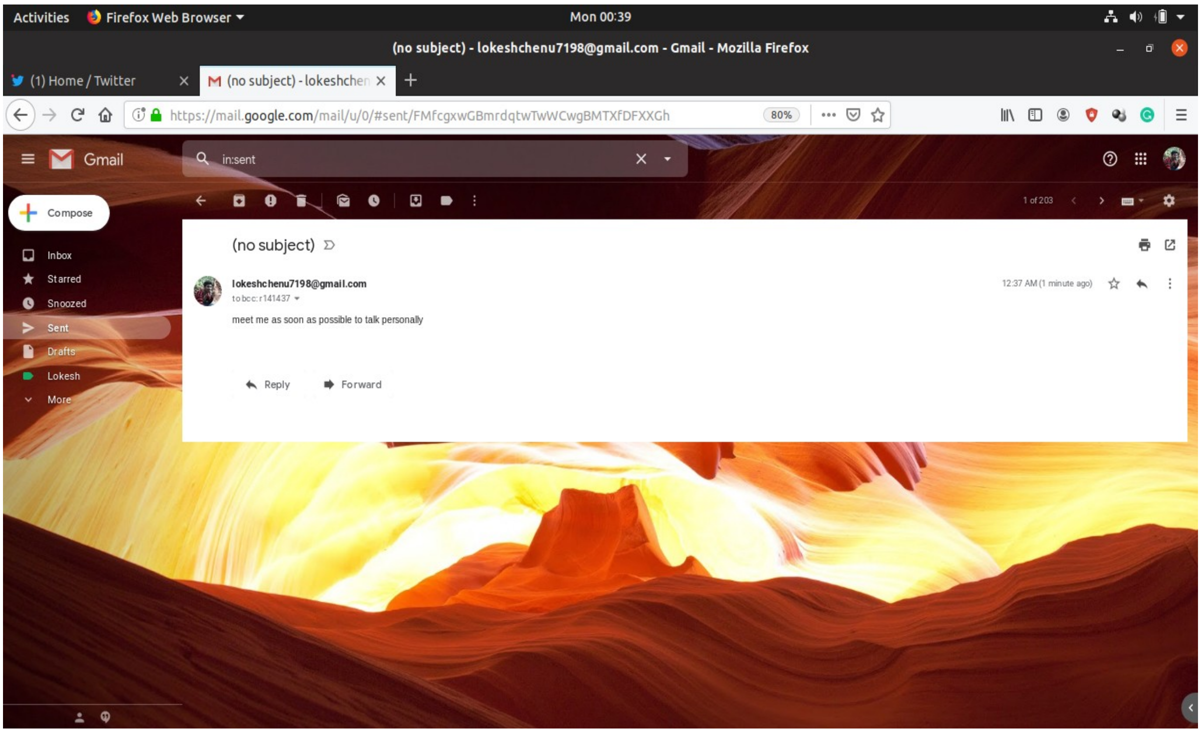
```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:869:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
Loki, I can't understood what you said
Say Something
find paper
Path:1 /home/lokeshchenu/Pictures/Wallpapers
Path:2 /home/lokeshchenu/.cache/wallpaper
Path:3 /home/lokeshchenu/Desktop/paper sv
Path:4 /home/lokeshchenu/Desktop/paper
Path:5 /home/lokeshchenu/Documents/personal/Screenshot_2019-09-13 Sakshi Telugu Daily YSR Kadapa District epaper dated Fri, 13 Sep 19.png
Path:6 /home/linuxbrew/.linuxbrew/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/unpaper.rb
Path:7 /home/linuxbrew/.linuxbrew/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/paperkey.rb
Path:8 /home/linuxbrew/.linuxbrew/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/wallpaper.rb
```

Figure 5.8 Directory search



```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
email prasad meet me as soon as possible to talk personally
```

Figure 5.9.1 Sending email


Activities
Firefox Web Browser
Mon 00:39
(no subject) - lokeshchenu7198@gmail.com - Gmail - Mozilla Firefox
(1) Home / Twitter
(no subject) - lokeshchen
https://mail.google.com/mail/u/0/#sent/FMfcgxwGBmrdqtwTwWCwgBMTXfDFXXGh
80%
Gmail
in:sent
Compose
Inbox
Starred
Snoozed
Sent
Drafts
Lokesh
More
1 of 203
Reply
Forward

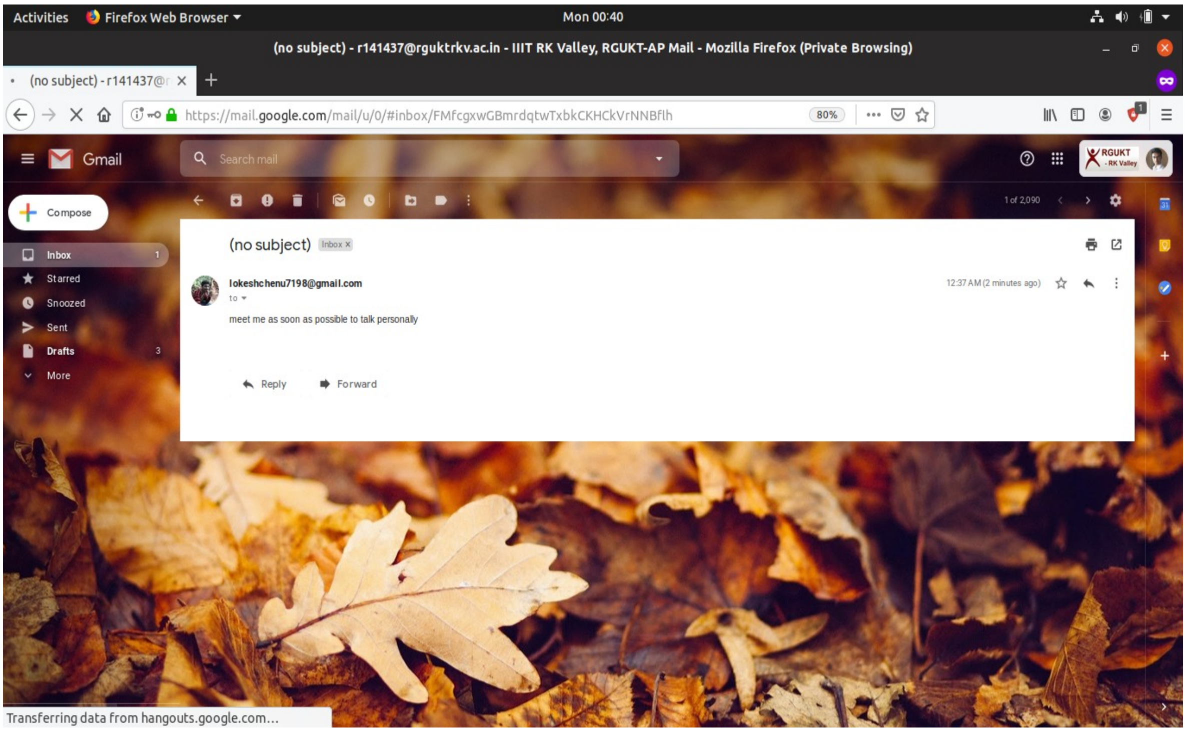
(no subject)

lokeshchenu7198@gmail.com
to bcc: r141437

12:37 AM (1 minute ago)

meet me as soon as possible to talk personally

Figure 5.9.2 Email sent successfully



(no subject) - r141437@rguktrkv.ac.in - IIIT RK Valley, RGUKT-AP Mail - Mozilla Firefox (Private Browsing)

https://mail.google.com/mail/u/0/#inbox/FMfcgxwGBmrdqtwTxbkCKHCkVrNNBflh

(no subject) Inbox ×

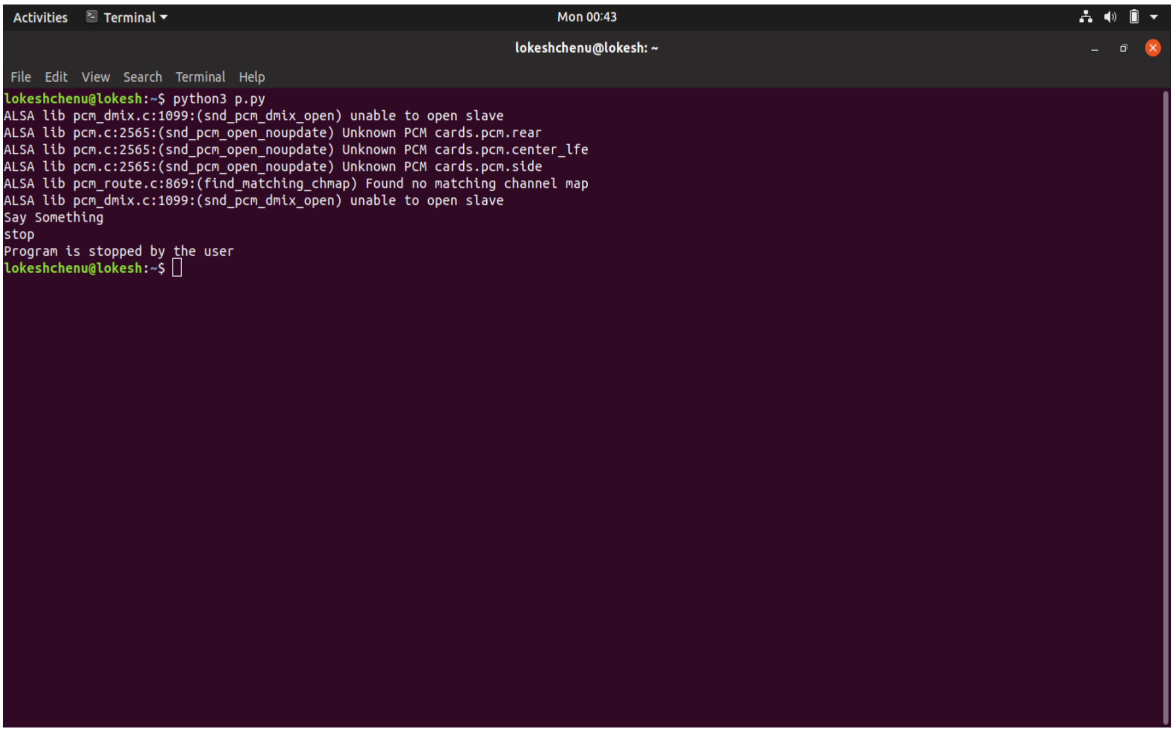
lokeshchenu7198@gmail.com
to

12:37 AM (2 minutes ago)

meet me as soon as possible to talk personally

Transferring data from hangouts.google.com...

Figure 5.9.3 Email received successfully



```
lokeshchenu@lokesh:~$ python3 p.py
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:869:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Say Something
stop
Program is stopped by the user
lokeshchenu@lokesh:~$ 
```

Figure 5.10 Terminating program manually

6.Result

This program lets the user access the system services and many more things from the internet. To most of the people, it seems like part of Alexa, Siri, etc., But in reality, it is somewhat varied from all of them. It is just a program, not software. Using the various packages like os, webbrowser, etc., we are utilizing the system resources and executing the tasks to be performed for the user based on their choice. These are the things which let the program varied from the softwares. This program resembles a small set of unsupervised learning techniques which is an applied concept of Machine Learning. In the future, if we train them using some training data then it will become supervised learning.

7.References

[1] pypi.org/project

[2] docs.python.org/2/library/webbrowser.html

[3] pyttsx3.readthedocs.io

[4] www.realpython.com

[5] www.geeksforgeeks.org

[6] www.github.com

[7] www.stackoverflow.com

[8] www.quora.com

[9] T. M. Mitchell, “Machine Learning”, McGraw-Hill International,1997.

[10] Pedregosa et al.,Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.