NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES

COURSE: OPERATING SYSTEM

TITLE: VOICE CONTROLLED SHELL

Course Teacher: Miss Mubashra Fayyaz

GROUP MEMBERS

AFFAN ZAHOOR 20k-0142

AQMER IJAZ 20k-1086

BILAL MAMJI 20k-1702

➢ Introduction

Voice-Controlled Shell is a personal assistant to execute bash commands/scripts from voice. It is a web application connected to a Google Compute Engine VM that will listen to your commands and perform them in the Linux-based terminal. Voice Controlled Shell is implemented using python script with following libraries:

- speech recognition
- PyQt5
- os
- sys
- gtts

Voice is recognized through Google API and different mp3 files are played to ask user for further details of commands (if required). UI part of Voice Controlled Shell is designed with QT designer and converted to python using Python 2.x/3.x, PyAudio libraries.

➢ Methodology

In computing, a shell is a user interface for access to an operating system's services. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation. Our project is a new take on the traditional methods to use shells. Our voice-controlled shell uses voice commands to run bash scripts that perform at least 30 various tasks like

- Informing the user of the time, date, day, and weather.
- Listing the files, creating new files, deleting files.
- Creating new directory, changing directories, file type.
- Opening web browsers, links through url, send emails.
- Shutting the machine down, rebooting, etc.

We are using a Google API for Speech-to-text (STS) and different python libraries for conversion of Speech-to-text such as gtts, speech_recognition, etc.

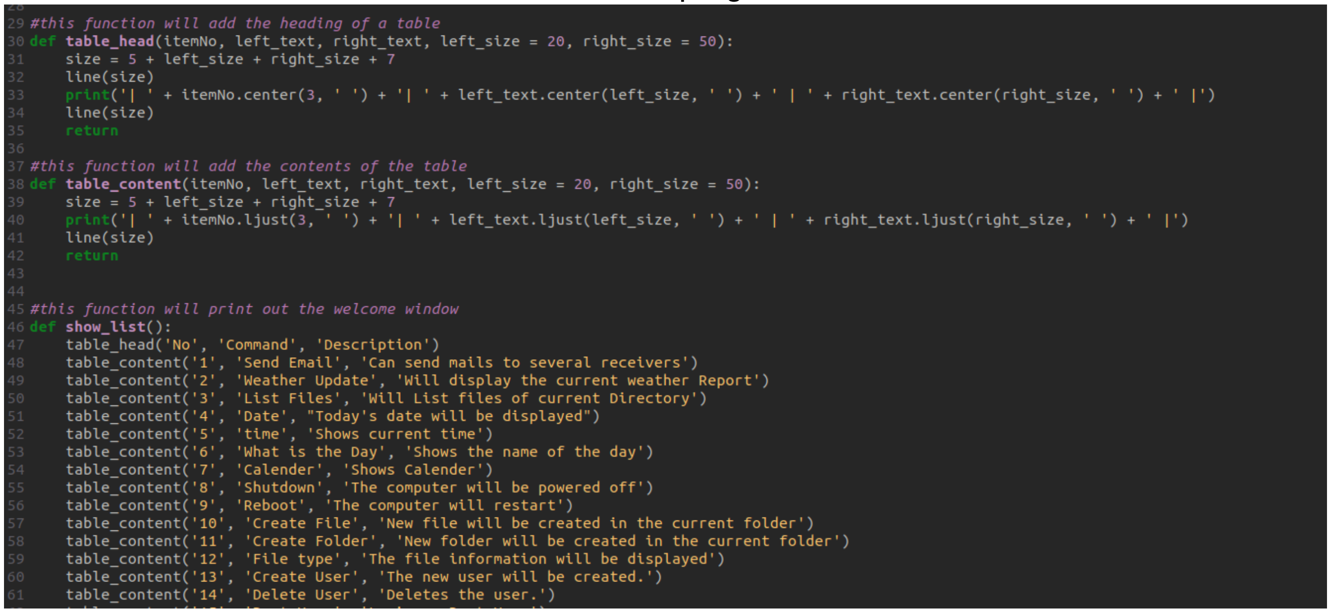
➢ Configuration steps

Our project is implemented using python. We have utilized the Google Speech-to-text (GSTT) and multiple python libraries in order to decode our voice commands to run and perform task in bash/shell script.

Moreover, we have used the py audio, webbrowser, subprocess, and system libraries that are built-in in python. The purpose of the pyaudio library is to detect the microphone and capture the sound effectively. The purpose of the subprocess and system library is to run bash commands. The purpose of webbrowser library is to open the different links in a web browser through url format.

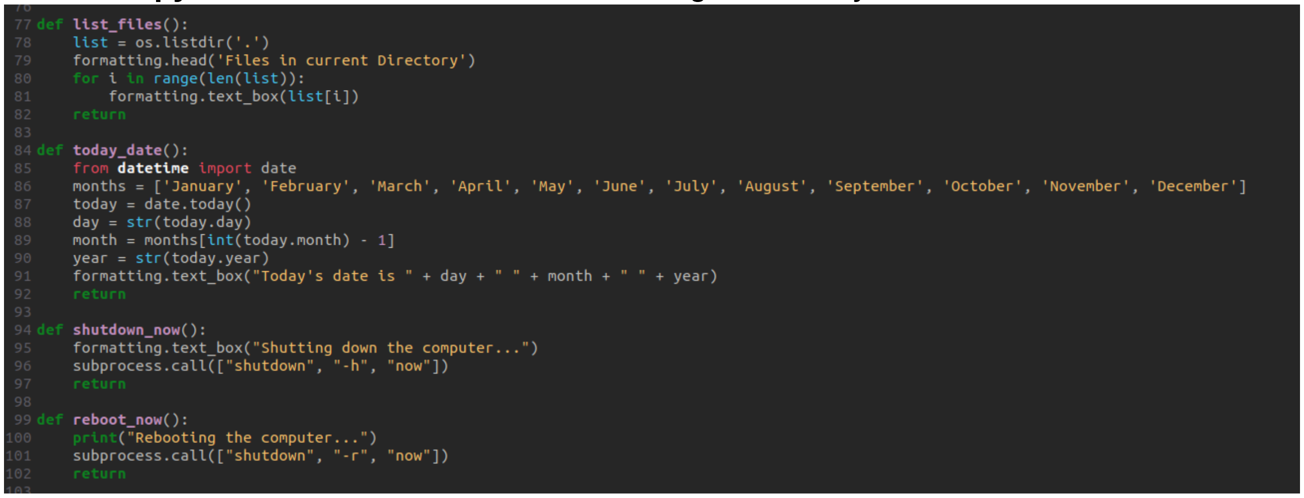
Basically, there are four different .py file in our project named formatting.py, modules.py, voice.py and main.py. The functions/coding implemented in these .py files are as follow:

- **formatting.py** contains formatting required for our code for example headings, table of content, and the main list of our source programs.



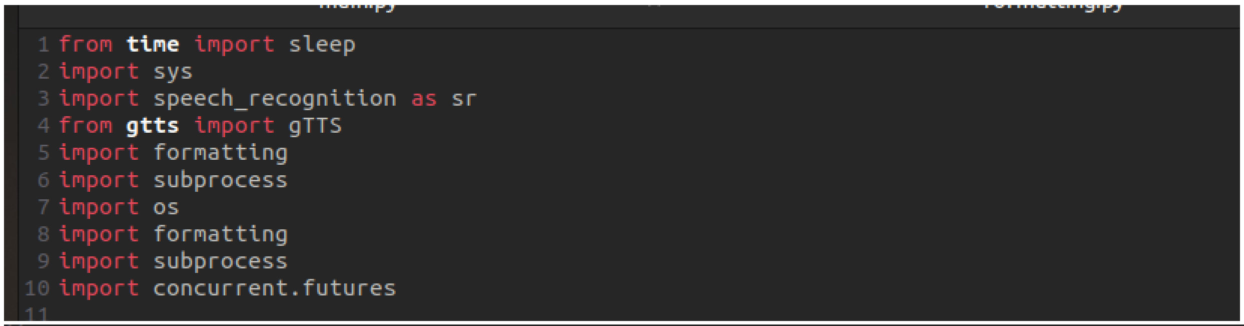
```python
#this function will add the heading of a table
def table_head(itemNo, left_text, right_text, left_size = 20, right_size = 50):
    size = 5 + left_size + right_size + 7
    line(size)
    print('| ' + itemNo.center(3, ' ') + '| ' + left_text.center(left_size, ' ') + ' | ' + right_text.center(right_size, ' ') + ' |')
    line(size)
    return

#this function will add the contents of the table
def table_content(itemNo, left_text, right_text, left_size = 20, right_size = 50):
    size = 5 + left_size + right_size + 7
    print('| ' + itemNo.ljust(3, ' ') + '| ' + left_text.ljust(left_size, ' ') + ' | ' + right_text.ljust(right_size, ' ') + ' |')
    line(size)
    return


#this function will print out the welcome window
def show_list():
    table_head('No', 'Command', 'Description')
    table_content('1', 'Send Email', 'Can send mails to several receivers')
    table_content('2', 'Weather Update', 'Will display the current weather Report')
    table_content('3', 'List Files', 'Will List files of current Directory')
    table_content('4', 'Date', "Today's date will be displayed")
    table_content('5', 'time', 'Shows current time')
    table_content('6', 'What is the Day', 'Shows the name of the day')
    table_content('7', 'Calender', 'Shows Calender')
    table_content('8', 'Shutdown', 'The computer will be powered off')
    table_content('9', 'Reboot', 'The computer will restart')
    table_content('10', 'Create File', 'New file will be created in the current folder')
    table_content('11', 'Create Folder', 'New folder will be created in the current folder')
    table_content('12', 'File type', 'The file information will be displayed')
    table_content('13', 'Create User', 'The new user will be created.')
    table_content('14', 'Delete User', 'Deletes the user.')
```

- **modules.py** contains the actual definition/coding of all 30 system commands.

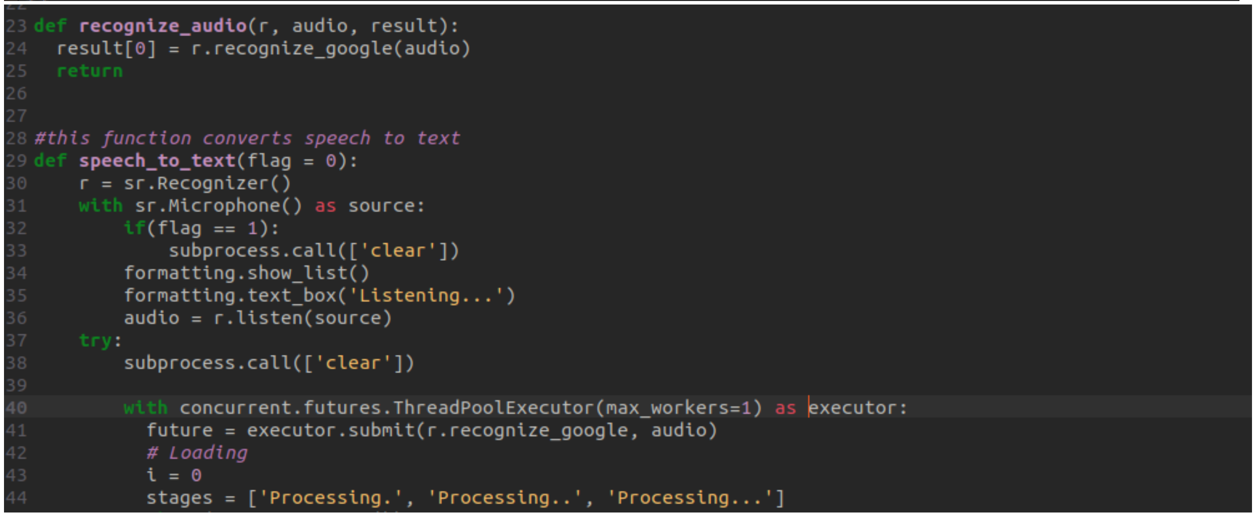


```python
def list_files():
    list = os.listdir('.')
    formatting.head('Files in current Directory')
    for i in range(len(list)):
        formatting.text_box(list[i])
    return

def today_date():
    from datetime import date
    months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
    today = date.today()
    day = str(today.day)
    month = months[int(today.month) - 1]
    year = str(today.year)
    formatting.text_box("Today's date is " + day + " " + month + " " + year)
    return

def shutdown_now():
    formatting.text_box("Shutting down the computer...")
    subprocess.call(["shutdown", "-h", "now"])
    return

def reboot_now():
    print("Rebooting the computer...")
    subprocess.call(["shutdown", "-r", "now"])
    return
```

- **voice.py** is importing all the python libraries such that speech_recognition, gtts, sys, os, that can convert Speech-to-text and can decode the voice in order to perform the tasks in bash script. It will detect clear voice and will return RequestError or UnknownValueError if some strange value is called.

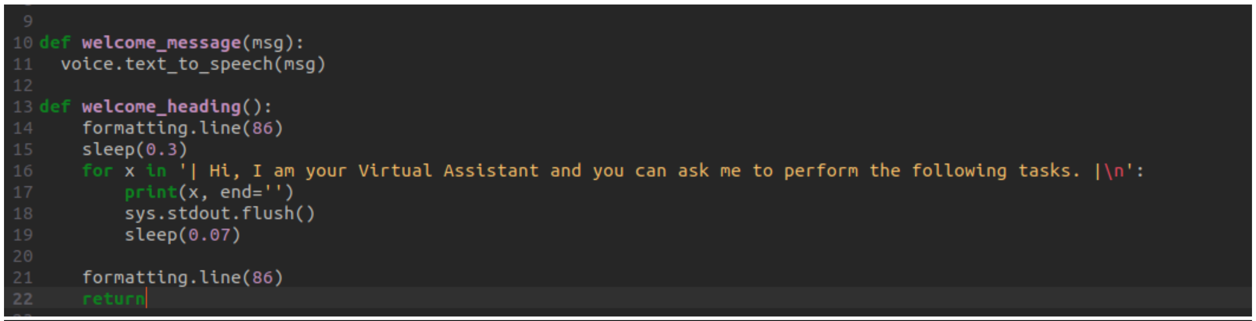


```python
from time import sleep
import sys
import speech_recognition as sr
from gtts import gTTS
import formatting
import subprocess
import os
import formatting
import subprocess
import concurrent.futures
```

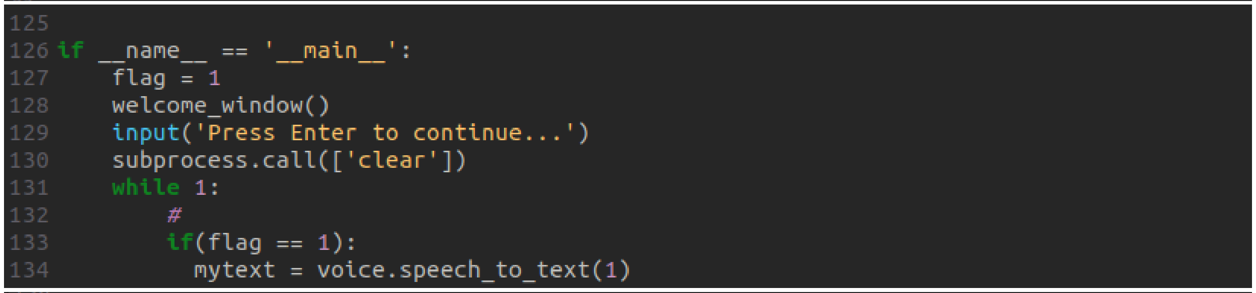


```python
def recognize_audio(r, audio, result):
  result[0] = r.recognize_google(audio)
  return


#this function converts speech to text
def speech_to_text(flag = 0):
    r = sr.Recognizer()
    with sr.Microphone() as source:
        if(flag == 1):
            subprocess.call(['clear'])
        formatting.show_list()
        formatting.text_box('Listening...')
        audio = r.listen(source)
    try:
        subprocess.call(['clear'])

        with concurrent.futures.ThreadPoolExecutor(max_workers=1) as executor:
          future = executor.submit(r.recognize_google, audio)
          # Loading
          i = 0
          stages = ['Processing.', 'Processing..', 'Processing...']
```

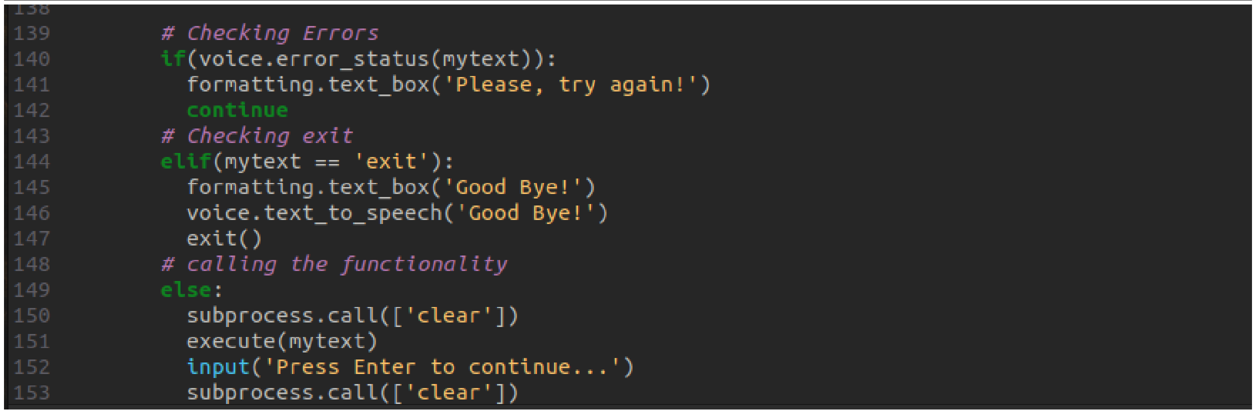
- **main.py** is importing all .py files used in this project. The output screen of the project **Voice Controlled Shell** are generated from *main.py*.



```python
def welcome_message(msg):
  voice.text_to_speech(msg)

def welcome_heading():
    formatting.line(86)
    sleep(0.3)
    for x in '| Hi, I am your Virtual Assistant and you can ask me to perform the following tasks. |\n':
        print(x, end='')
        sys.stdout.flush()
        sleep(0.07)

    formatting.line(86)
    return
```



```python

if __name__ == '__main__':
    flag = 1
    welcome_window()
    input('Press Enter to continue...')
    subprocess.call(['clear'])
    while 1:
        #
        if(flag == 1):
          mytext = voice.speech_to_text(1)
```



```python
        # Checking Errors
        if(voice.error_status(mytext)):
          formatting.text_box('Please, try again!')
          continue
        # Checking exit
        elif(mytext == 'exit'):
          formatting.text_box('Good Bye!')
          voice.text_to_speech('Good Bye!')
          exit()
        # calling the functionality
        else:
          subprocess.call(['clear'])
          execute(mytext)
          input('Press Enter to continue...')
          subprocess.call(['clear'])
```

➢ Procedure

The procedure of voice-controlled shell is as follows:

- PC has a microphone and a speaker connected to it which is always listening.
- Firstly, after compilation it will display a welcome message along with a voice command.
- Next the user will speak one of the valid commands from the list of commands that we have programmed.
- The program will verify if the command is valid and then find the action to be performed.
- In order to repeat this process, you just need to press Enter and continue.
- To stop the process, you have to call 'EXIT".

➢ Code Link:

https://drive.google.com/drive/folders/1gMRn63vaaMQTYxti_0GiBu8YoCR54q1?usp=sharing

➢ References:

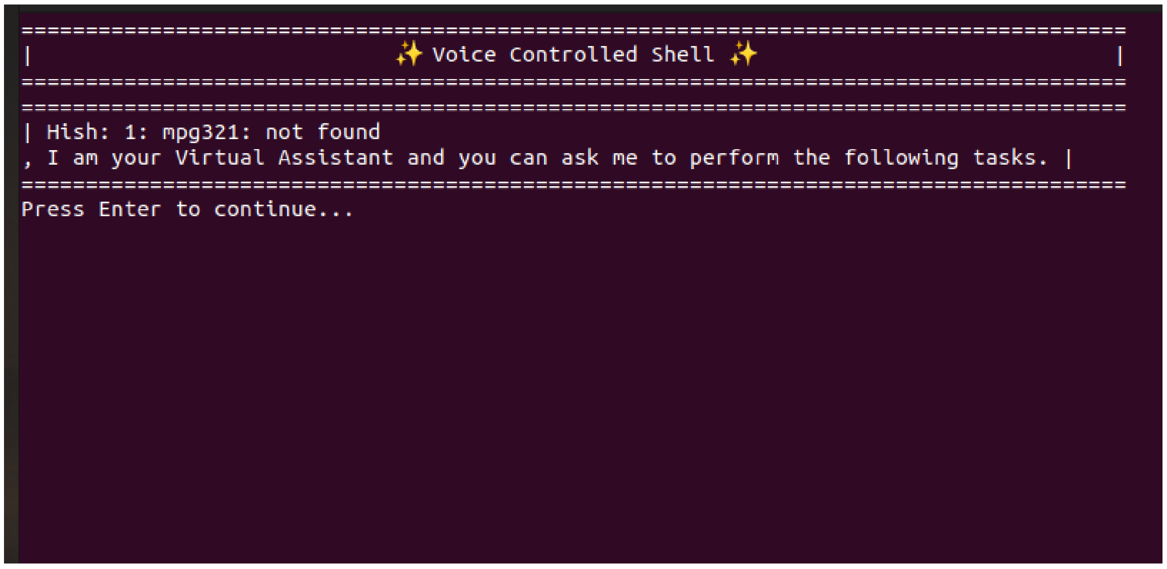
https://atwing.net/home%20automation/shell-commands-assistant/

https://www.noobslab.com/2014/06/control-your-ubuntulinux-mint-system.html

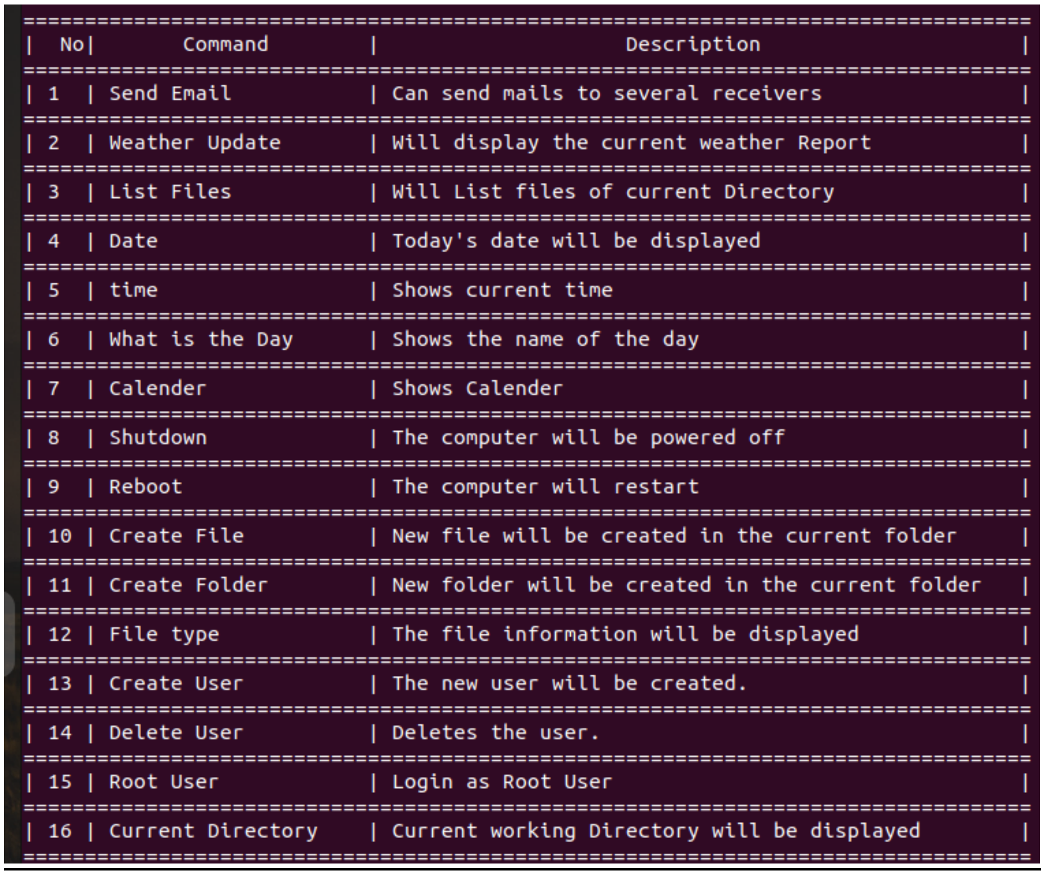
http://www.kscst.iisc.ernet.in/spp/40_series/39S_bestprojreports/39S_BE_1732.pdf

➢ Output:

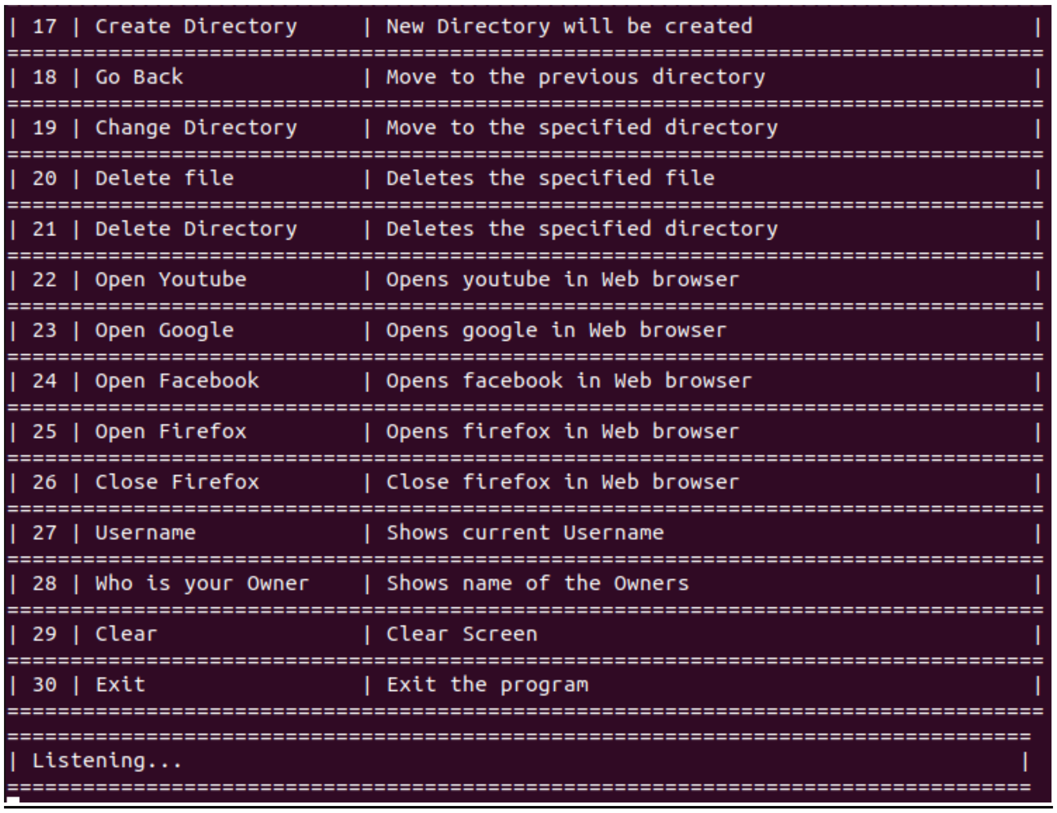
It is the output of main screen and the list of system commands. Rest of the outputs will be displayed at the time of evaluation.



```
==============================================================================
|                            ✨ Voice Controlled Shell ✨                      |
==============================================================================
==============================================================================
| Hish: 1: mpg321: not found
, I am your Virtual Assistant and you can ask me to perform the following tasks. |
==============================================================================
Press Enter to continue...
```



```
=================================================================================
|  No|       Command         |                    Description                   |
=================================================================================
| 1  | Send Email            | Can send mails to several receivers              |
=================================================================================
| 2  | Weather Update        | Will display the current weather Report          |
=================================================================================
| 3  | List Files            | Will List files of current Directory             |
=================================================================================
| 4  | Date                  | Today's date will be displayed                   |
=================================================================================
| 5  | time                  | Shows current time                               |
=================================================================================
| 6  | What is the Day       | Shows the name of the day                        |
=================================================================================
| 7  | Calender              | Shows Calender                                   |
=================================================================================
| 8  | Shutdown              | The computer will be powered off                 |
=================================================================================
| 9  | Reboot                | The computer will restart                        |
=================================================================================
| 10 | Create File           | New file will be created in the current folder   |
=================================================================================
| 11 | Create Folder         | New folder will be created in the current folder |
=================================================================================
| 12 | File type             | The file information will be displayed           |
=================================================================================
| 13 | Create User           | The new user will be created.                    |
=================================================================================
| 14 | Delete User           | Deletes the user.                                |
=================================================================================
| 15 | Root User             | Login as Root User                               |
=================================================================================
| 16 | Current Directory     | Current working Directory will be displayed      |
=================================================================================
```



```
| 17 | Create Directory      | New Directory will be created                    |
=================================================================================
| 18 | Go Back               | Move to the previous directory                   |
=================================================================================
| 19 | Change Directory      | Move to the specified directory                  |
=================================================================================
| 20 | Delete file           | Deletes the specified file                       |
=================================================================================
| 21 | Delete Directory      | Deletes the specified directory                  |
=================================================================================
| 22 | Open Youtube          | Opens youtube in Web browser                     |
=================================================================================
| 23 | Open Google           | Opens google in Web browser                      |
=================================================================================
| 24 | Open Facebook         | Opens facebook in Web browser                    |
=================================================================================
| 25 | Open Firefox          | Opens firefox in Web browser                     |
=================================================================================
| 26 | Close Firefox         | Close firefox in Web browser                     |
=================================================================================
| 27 | Username              | Shows current Username                           |
=================================================================================
| 28 | Who is your Owner     | Shows name of the Owners                         |
=================================================================================
| 29 | Clear                 | Clear Screen                                     |
=================================================================================
| 30 | Exit                  | Exit the program                                 |
=================================================================================
================================================================================
| Listening...                                                                 |
================================================================================
```