1



# National University of Computer & Emerging Sciences (FAST-NU)

**Voice Controlled Operating System**

**Supervisor: Co Supervisor:**

**Miss Anaum Hamid Miss Rabia Ansari**

**By:**

**Project Team**

**Hiba Zubair : 18k-1361  
Ramsha Nadeem : 18k-0263**

**Submission Date**

## 18th December,2020

​

2

1. **Introduction:**

Voice commands allows you to perform multiple actions with voice. This project focuses on accessing shell commands to run bash scripts by simply speaking commands.

The project is implemented using Python, with the aid of Google API for speech-to-text services.

The purpose of this project is to simplify the process of executing commands.

1. **How PROJECT STARTS:**

* The project asks the user to speak a command from the list of commands, provided in the README.txt file.
* The user then speaks the command, which is recognized, and the respective task is performed.
* If the audio is unrecognizable, a message is printed and the user has to repeat the command.
* If the command is invalid, the user has to speak the command again.
* The program is exited when the user says “exit”.

1. **PROBLEMS FACED:**

The major problem that occurred was to identify the libraries that were needed in order to detect the audio, and convert it to text. And how to link the operating system with the python program, for it to manipulate shell commands.   
Even after the installation of the libraries, there were multiple errors and warnings that occurred while trying to detect audio.

1. **SOLUTION TO EACH PROBLEM:**

Multiple libraries were installed, the errors were resolved by trial-and-error methodology, it was easier once it was identified how to work with the libraries.

Furthermore:

OS , webbrowser , timedate, subprocess libraries provided by python provide the functionality of linking the code to operating system shell and manipulating the commands.

1. **THE ACTUAL WORKING OF THE PROJECT (Methodology):**

**a. EXPLANATION OF SOURCE CODE:**

There are two python files:

* main.py
* sys\_commands.py

**Functions in main.py and their explanation:**

* sound(): receives a text command, converts it into audio using gTTS.
* detectAudio(): receives two objects, microphone and recognizer, used to detect audio from the mic.
* main(): gets the microphone and recognizer objects from speech\_recognition library, detects audio, and send the specific command to sys\_commands.py in order to perform the specific task.

**Functions in sys\_commands.py and their explanation:**

There is just one function that recieves text commands and performs the said task.

1. **PROJECT ASPECTS in CURRENT/FUTURE TECHNOLOGY:**

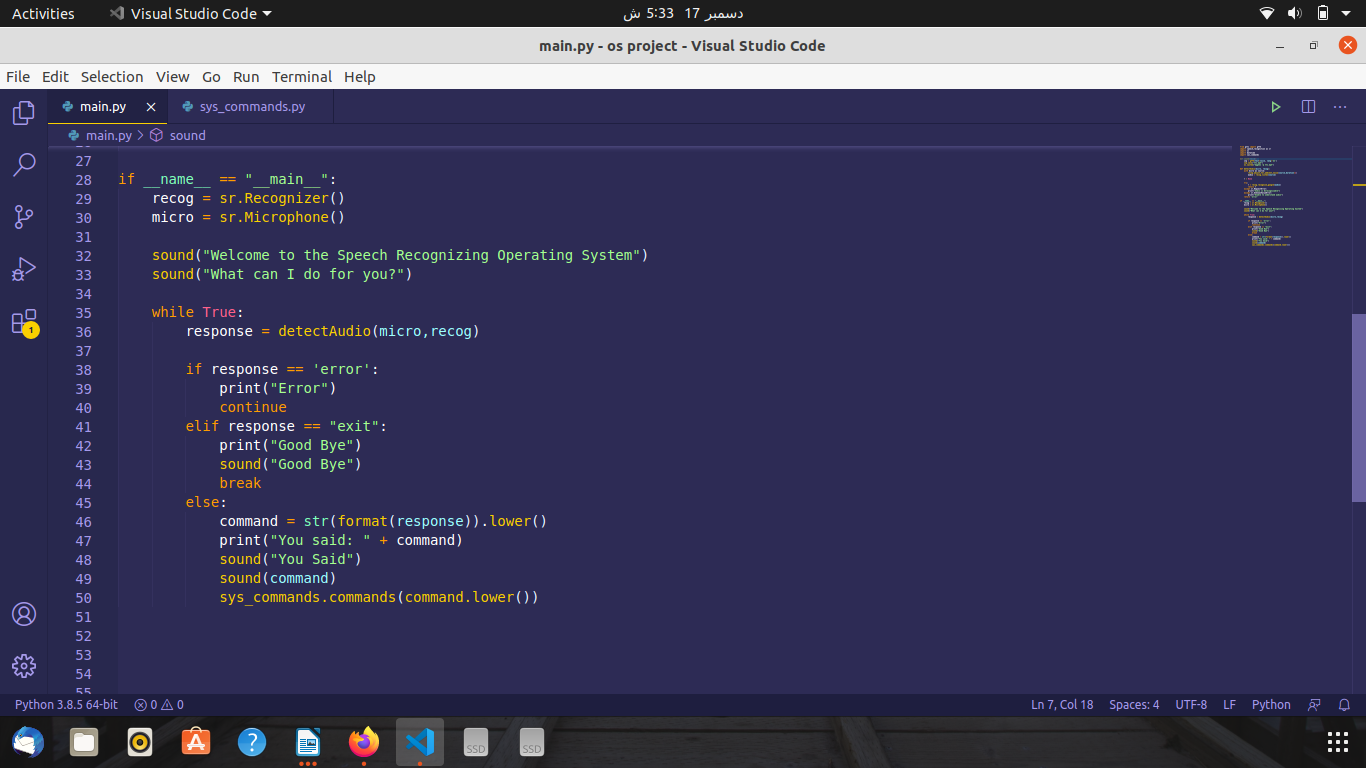
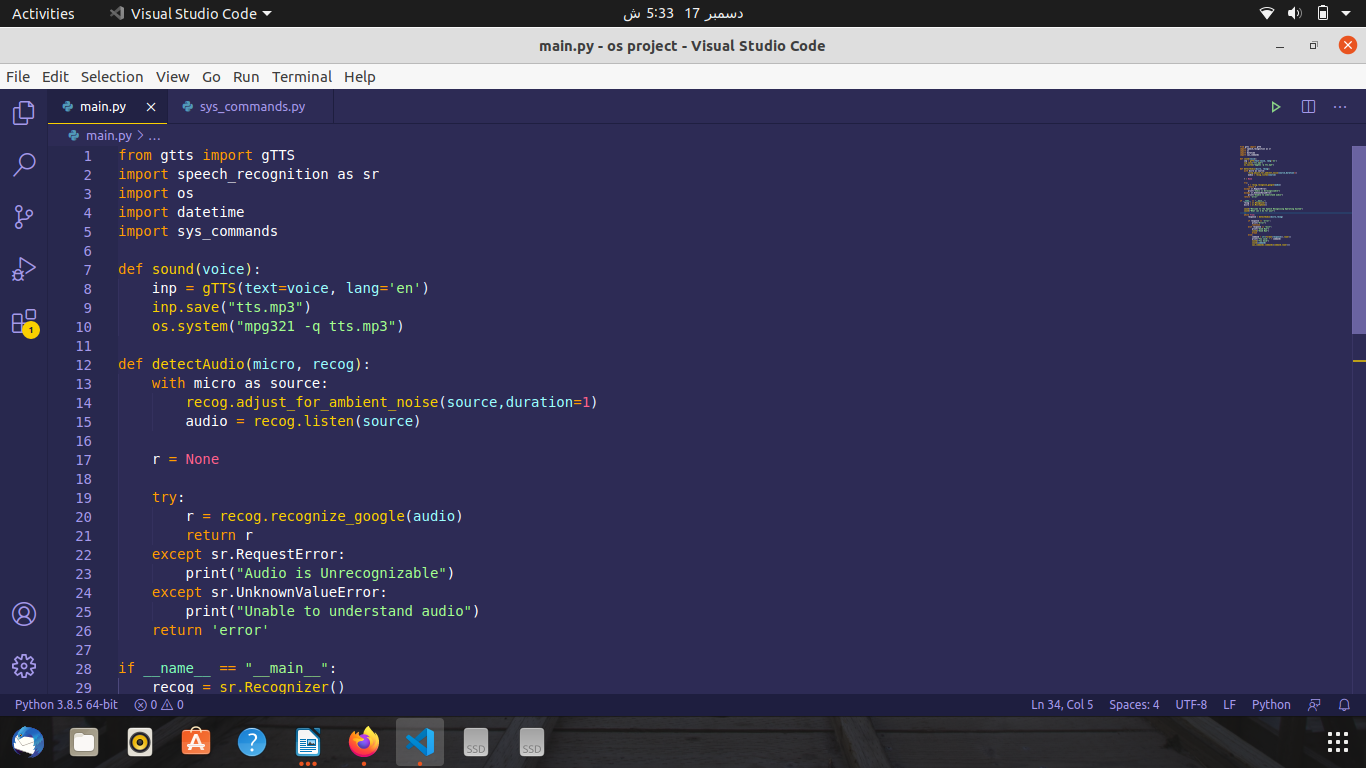
Voice controlled shell that responds to voice commands and executes shell scripts/commands in order to simplify the process of executing processes and to automate the use of the computer itself.

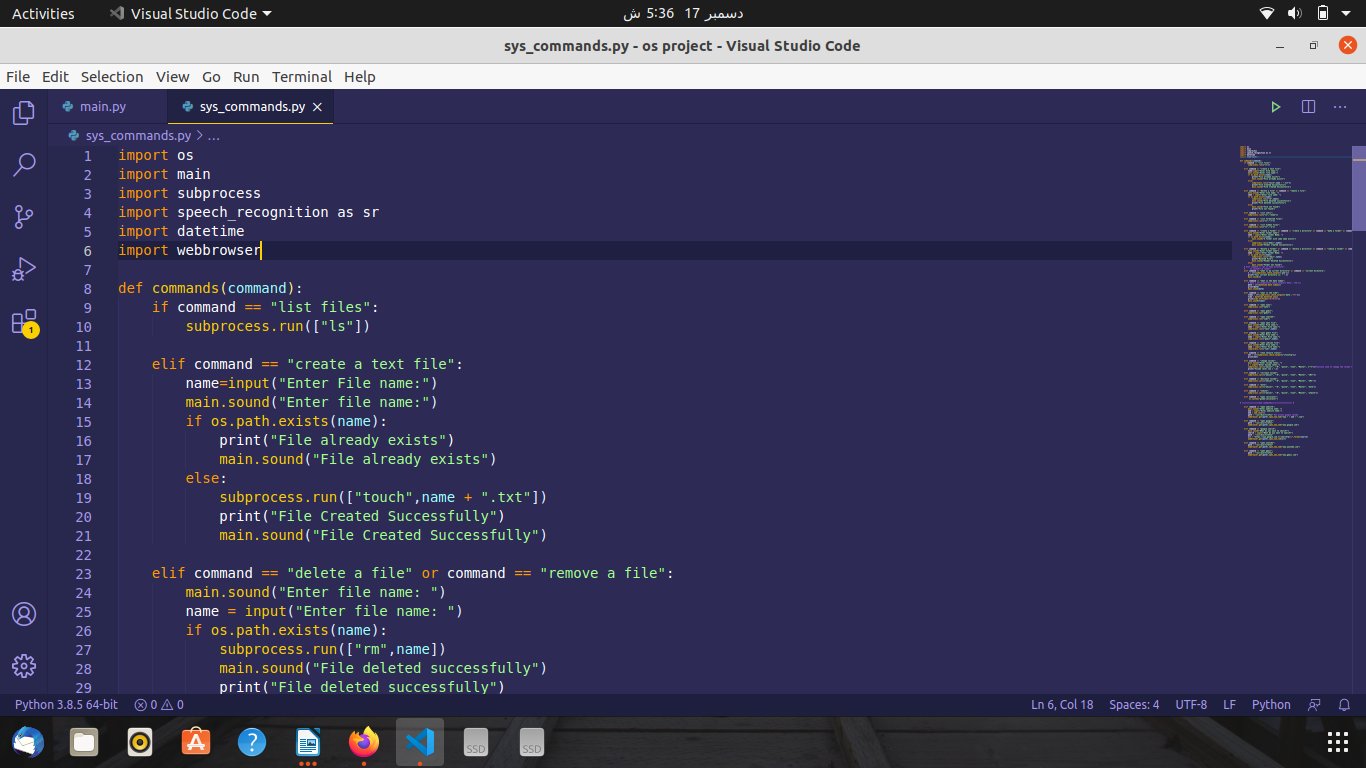
The aim is to ease the use of the system, even for those people who may not be familiar with programming at all.

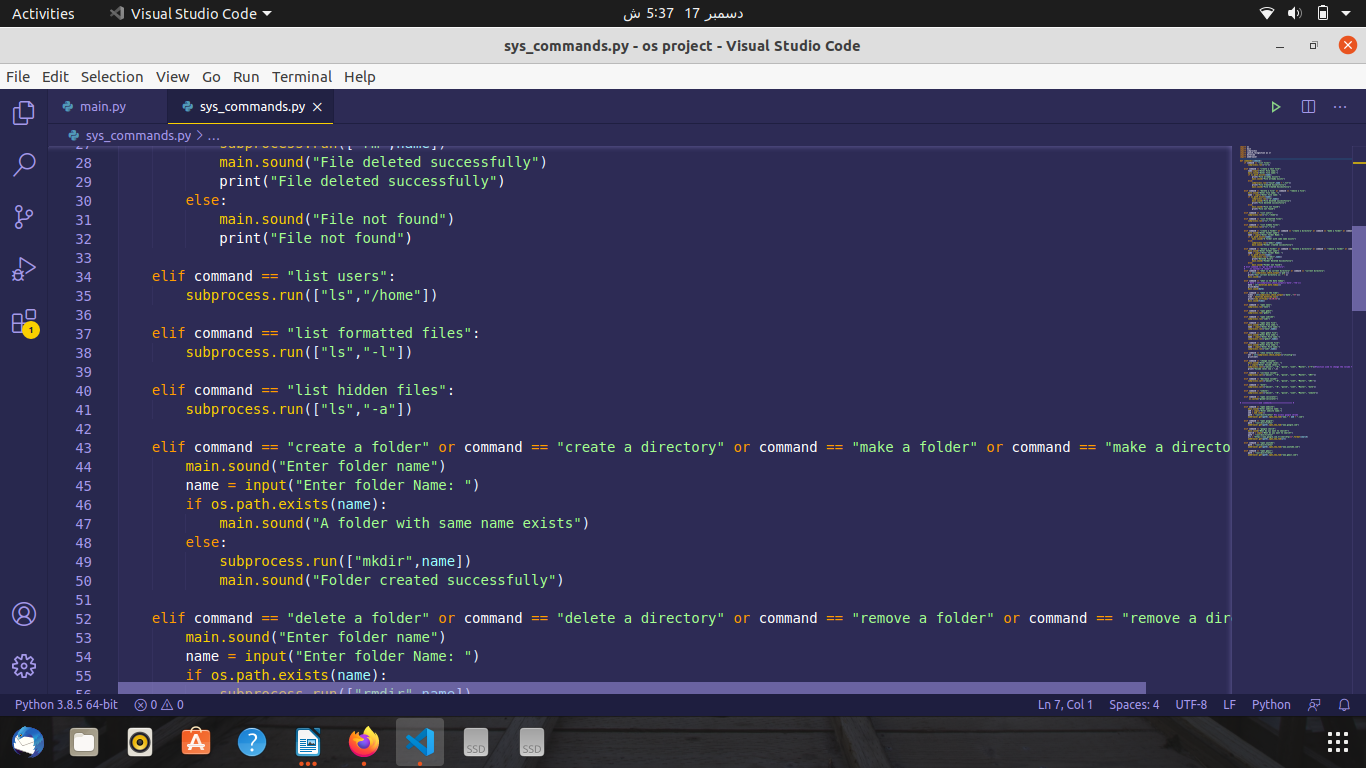
1. **PROJECT CONFIGURATION AND CODE:**

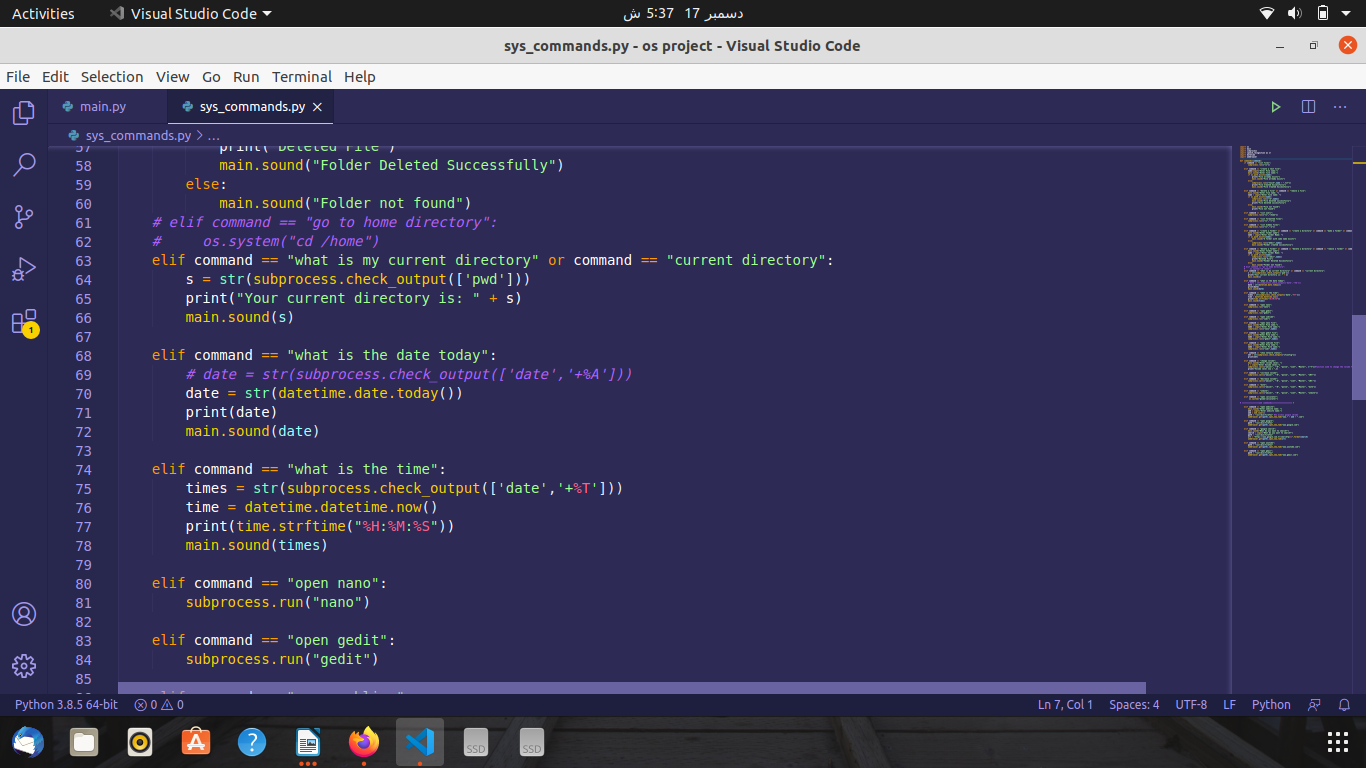
* First of all the system packages have to be updated.
* Then python, its setup tools and pip was required.
* The required libraries are:
  + - libasound-dev
    - portaudio19-dev
    - libportaudio2
    - libportaudiocpp0
    - ffmpeg
    - libav-tools
    - pyaudio
    - SpeechRecognition
    - gTTS (google text-to-speech)

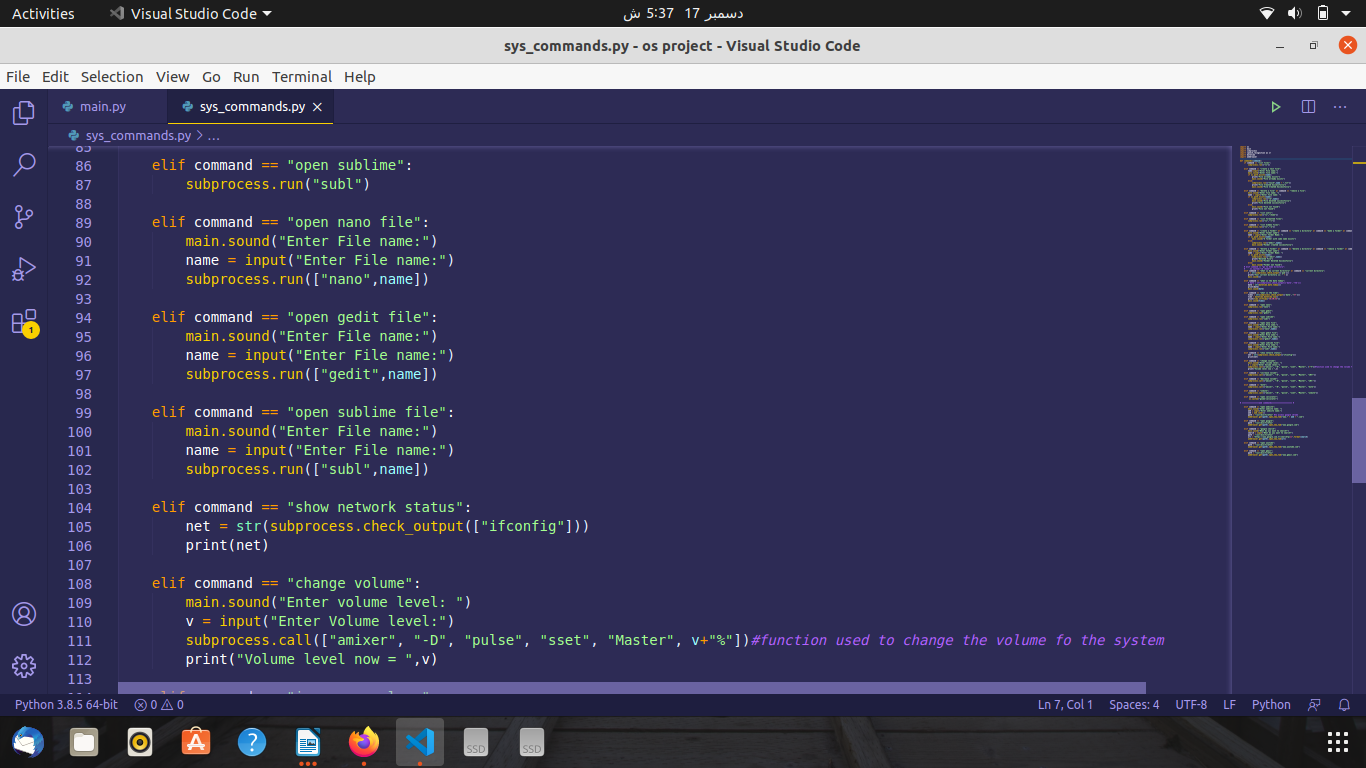
Code Snippets:

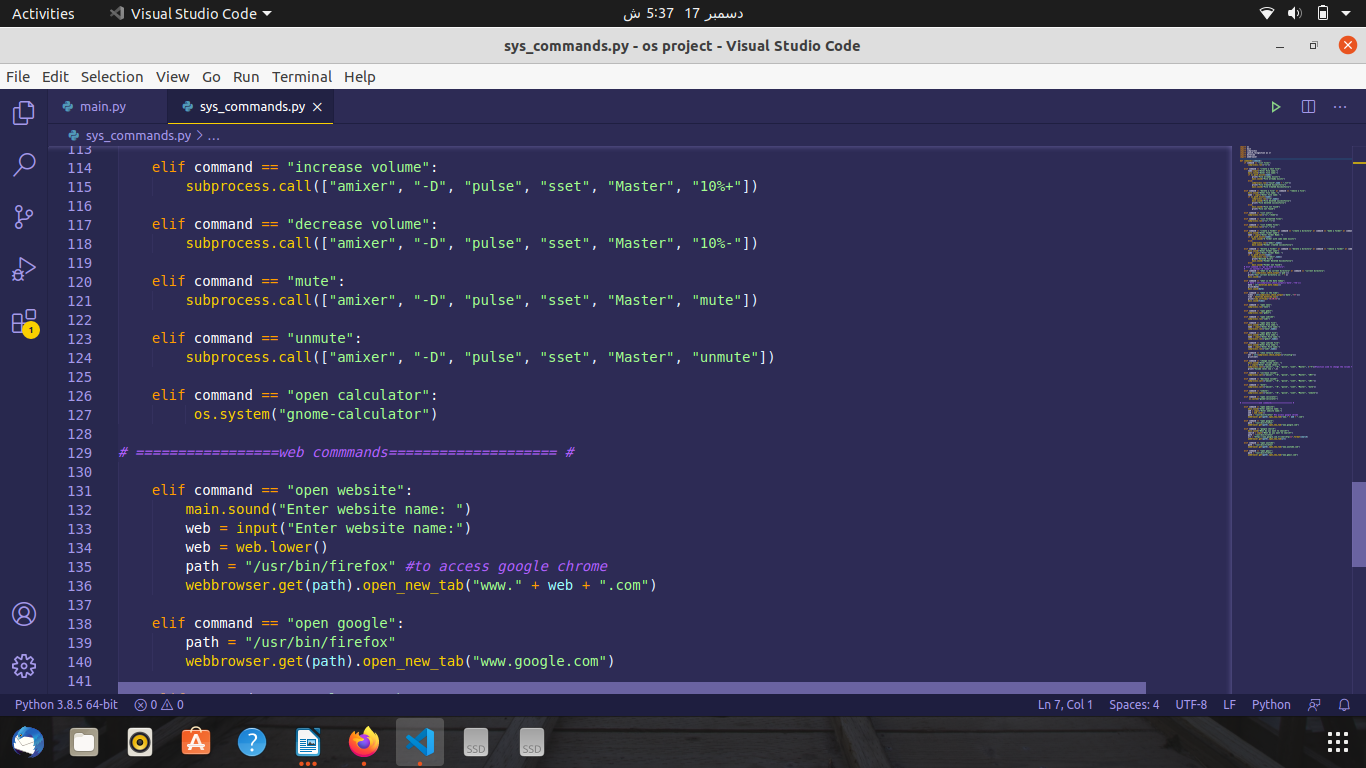


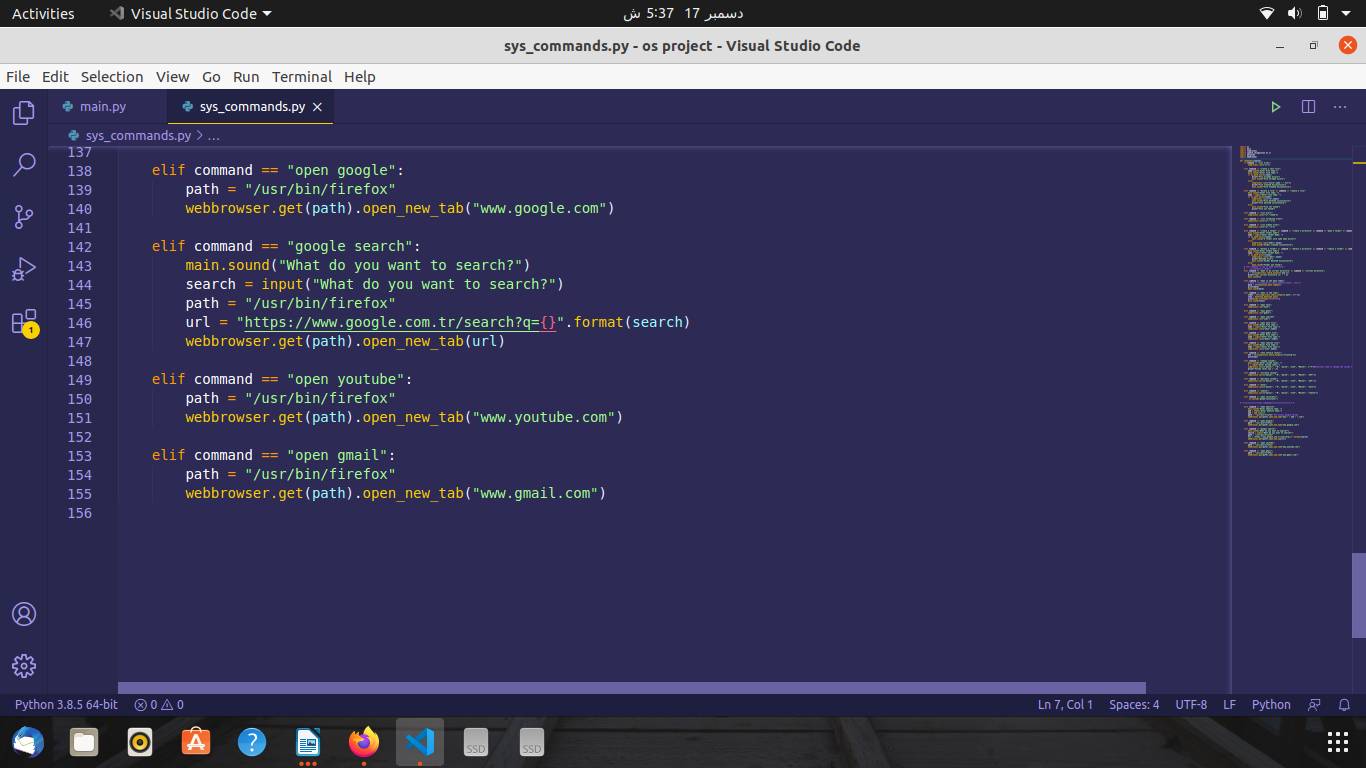


****

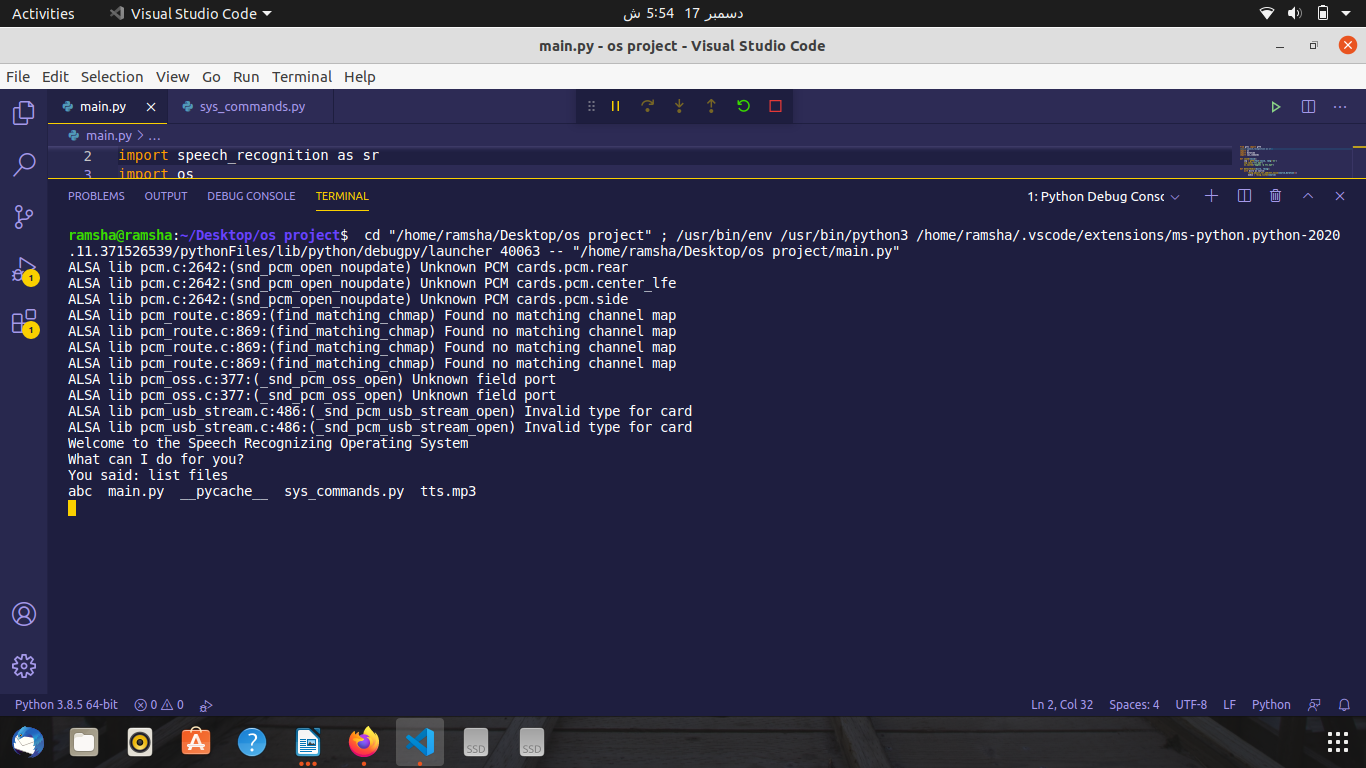
****

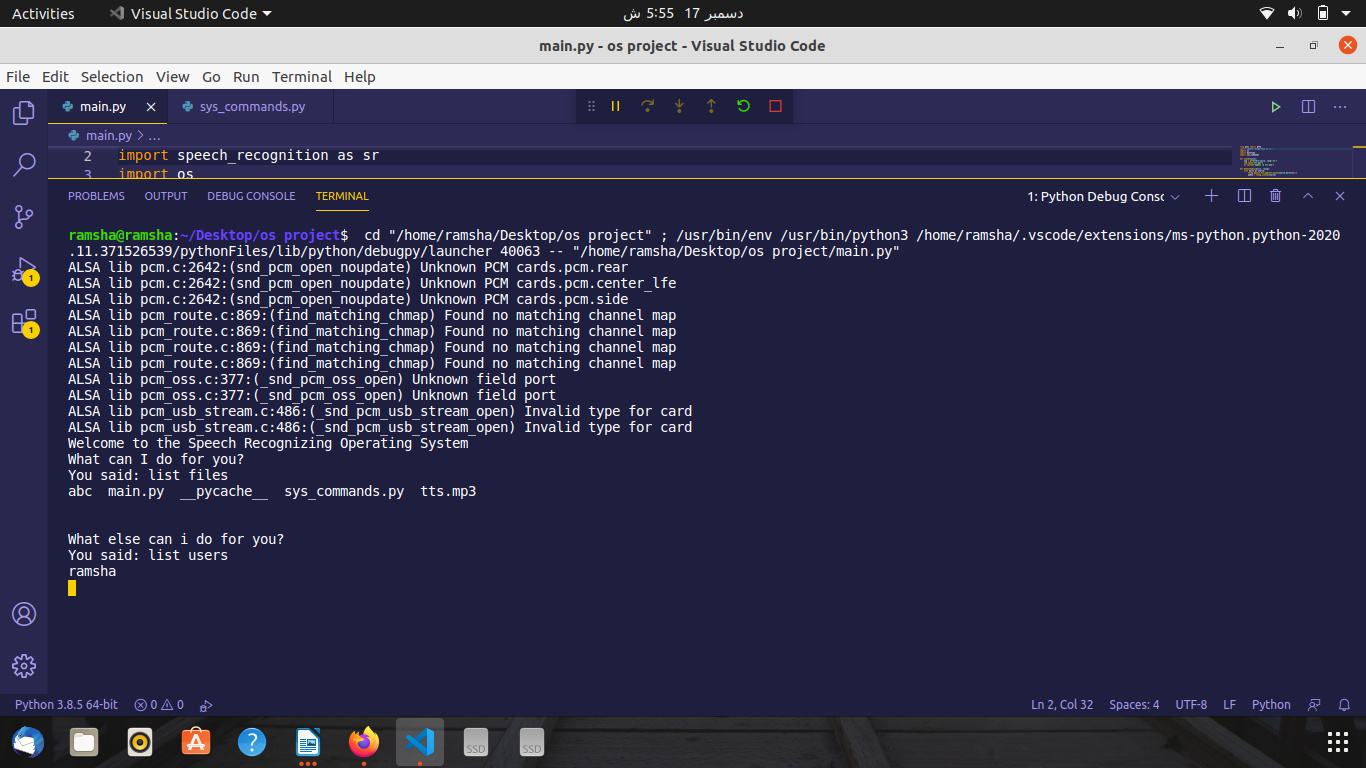
****

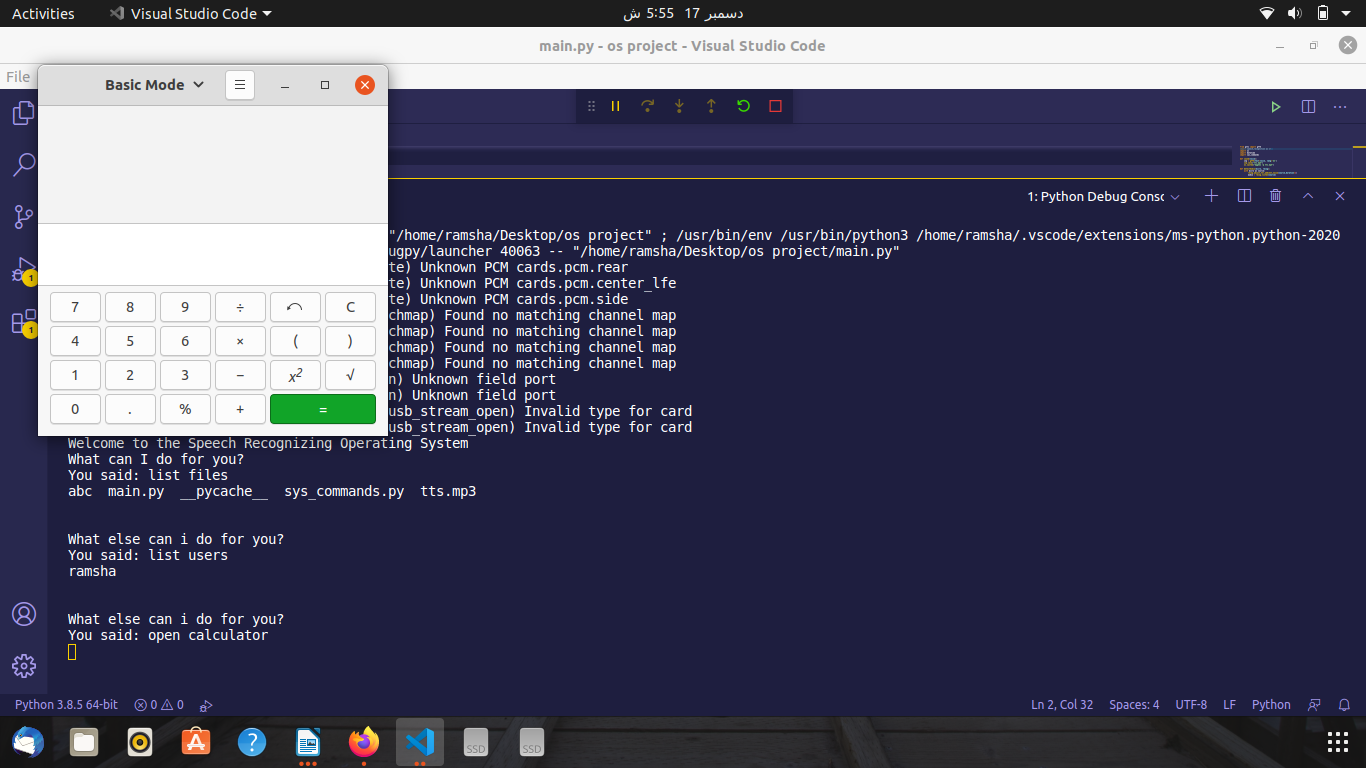
****

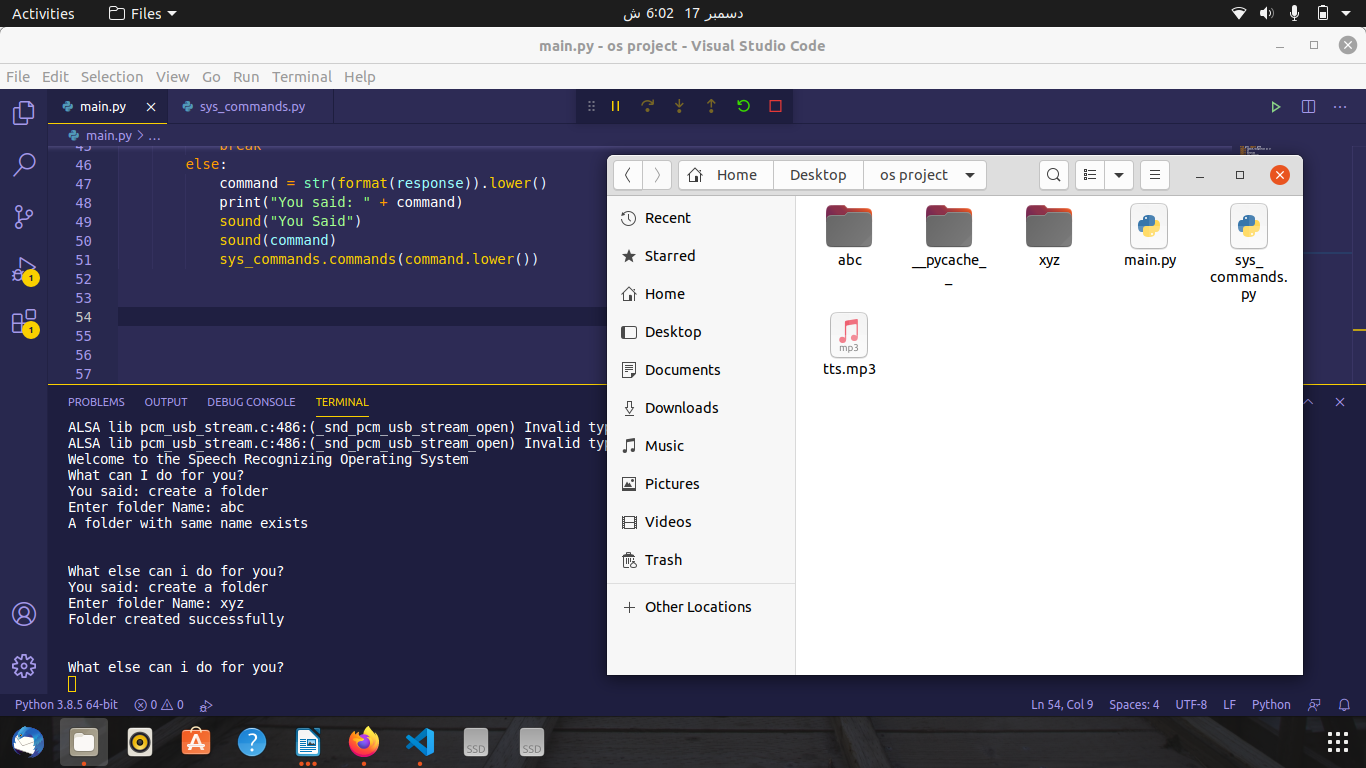
****

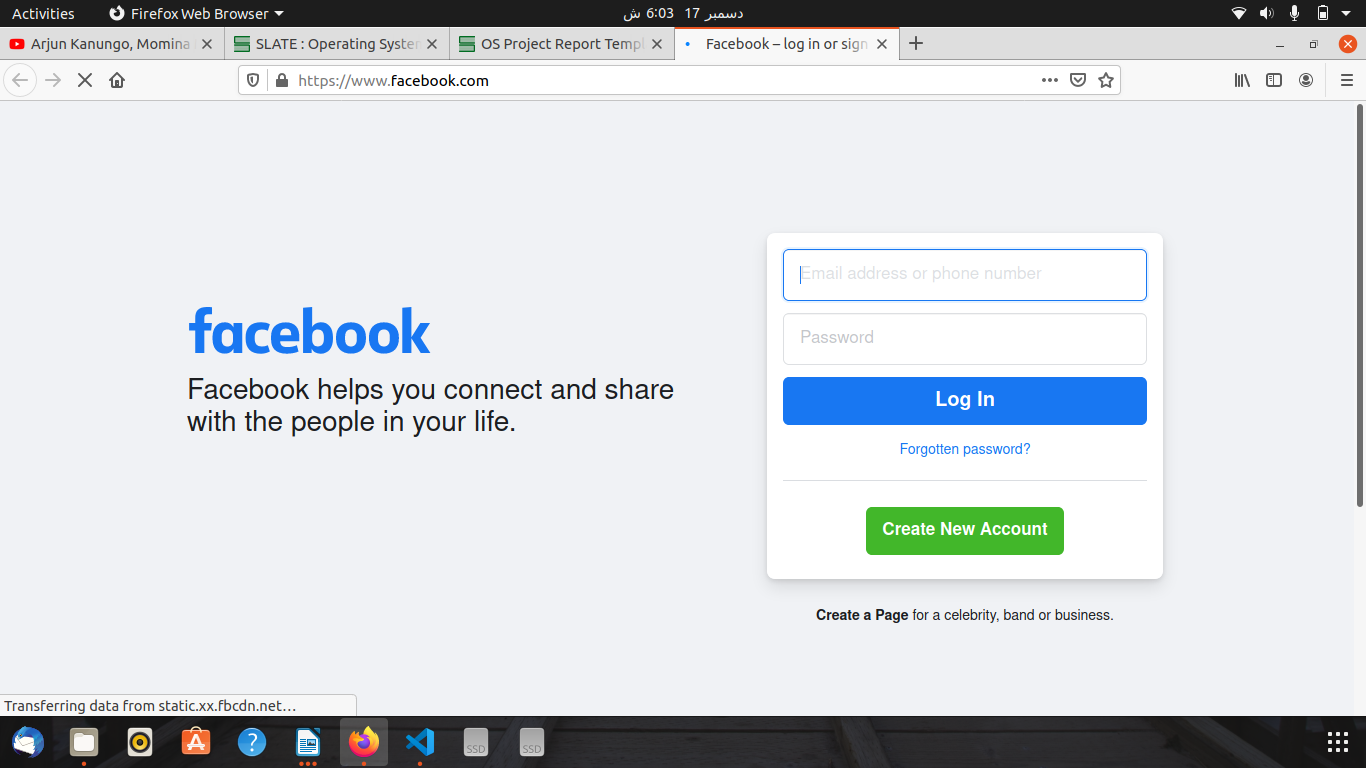
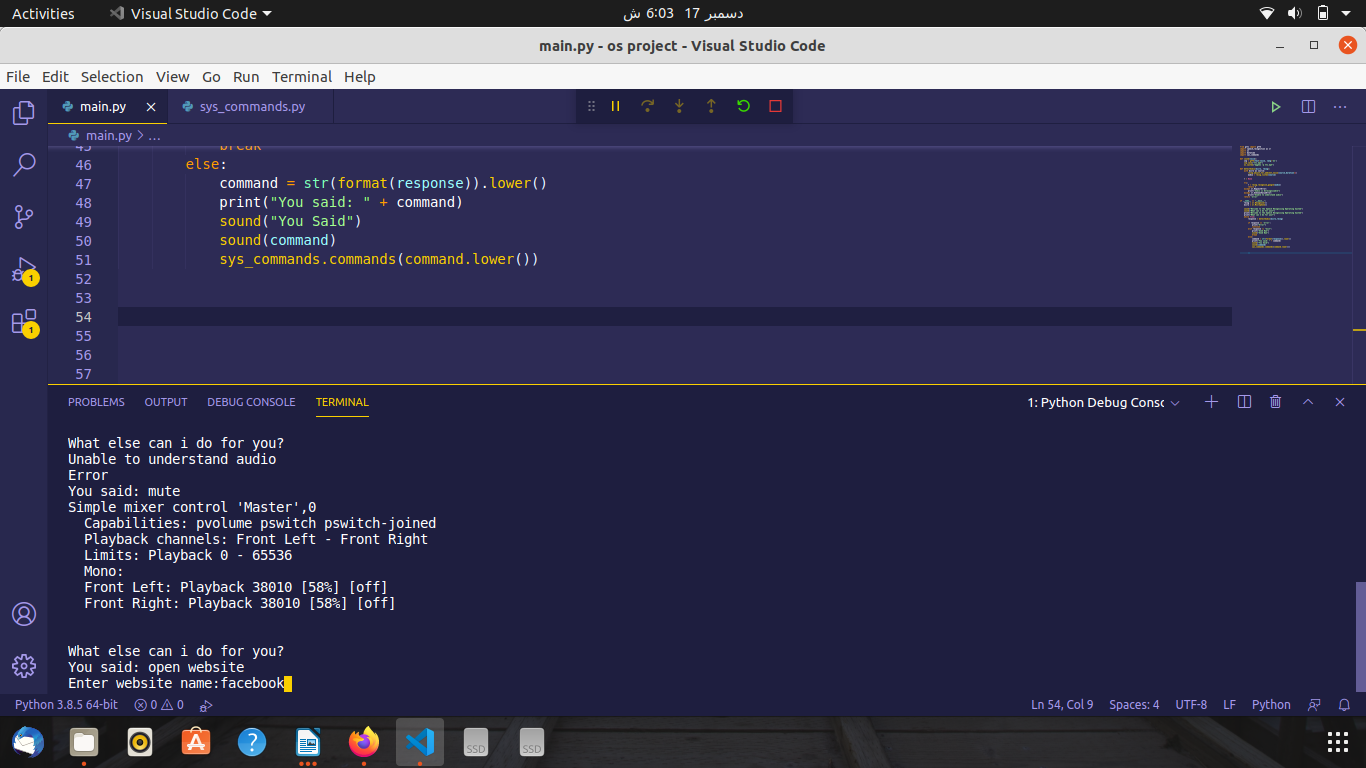
**There are 28 commands in total, and more can be added. The sample output of a few commands is as follows:**

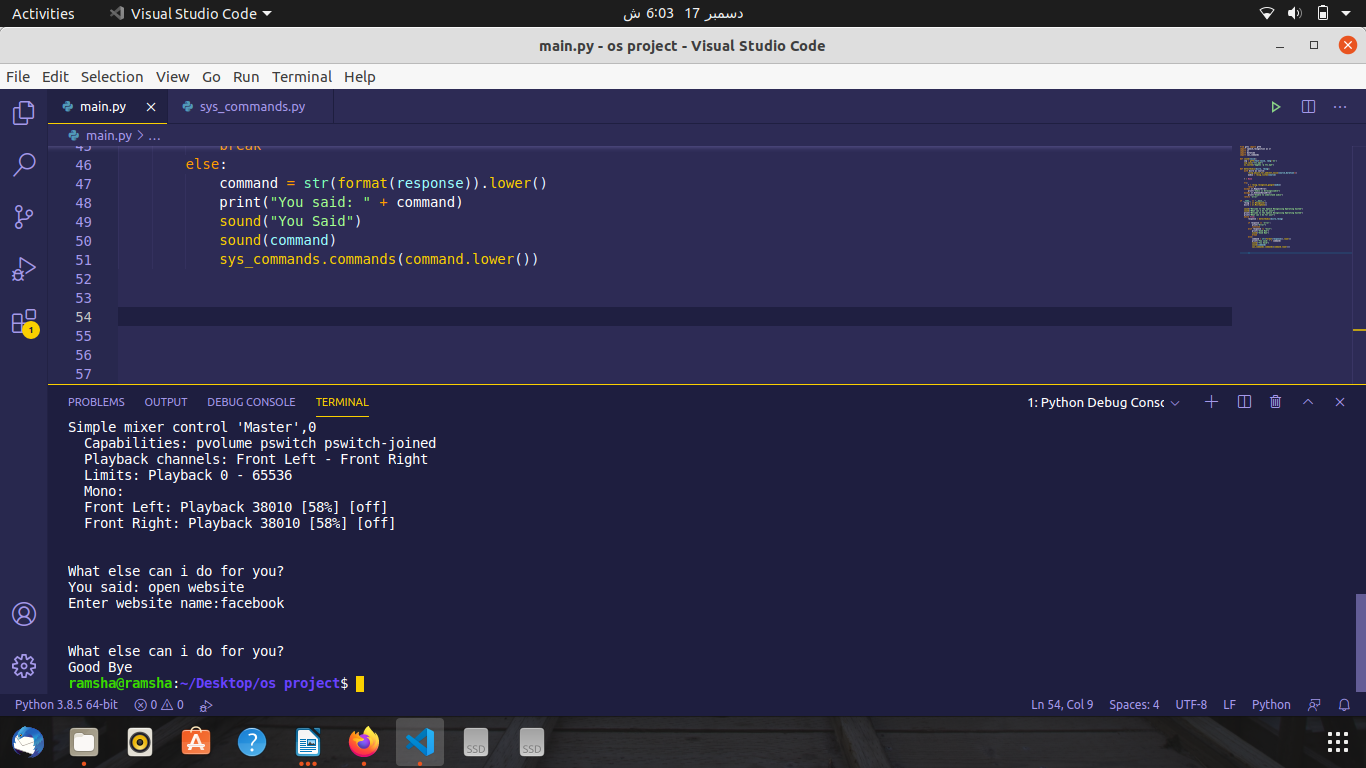
****

****

****

****

****

****

**REFERENCES:**

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

[https://atwing.net/home%20automation/shell-commands-assistant/](https://atwing.net/home automation/shell-commands-assistant/)​