# SHUTDOWN AND RESTART PC USING VOICE COMMAND

# BACHELOR OF TECHNOLOGY

### IN

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**P.RAM NIVESH REDDY (18C91A0576)**

**PANNALA DIVYA (18C91A0567)**

**N. SRIHARSHA (18C91A0562)**

**Under the Esteemed guidance of**

Dr. G. Venkat Koti Reddy

Associate Professor



### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE**

**(COLLEGE OF ENGINEERING)**

***(Approved by AICTE New Delhi, Permanently Affiliated to JNTU Hyderabad, Accredited by NAAC with ‘A’ Grade)***

**Bogaram (V), Keesara (M), Medchal District -501 301.**

### 2021 - 2022

**HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE**

#### (COLLEGE OF ENGINEERING)

***(Approved by AICTE New Delhi, Permanently Affiliated to JNTU Hyderabad, Accredited by NAAC with ‘A’ Grade)***

**Bogaram (V), Keesara (M), Medchal Dist-501301.**

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**CERTIFICATE**

This is to certify that the mini project entitled “**SHUTDOWN AND RESTART PC USING VOICE COMMAND**” is being submitted by**P.RAM NIVESH REDDY(18C91A0576) , PANNALA DIVYA (18C91A0567), N.SRIHARSHA (18C91A0562),**  in Partial fulfillment of the academic requirements for the award of the degree of Bachelor of Technology in “COMPUTER SCIENCE AND ENGINEERING” HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE, JNTU Hyderabad during the year 2021- 2022.

**INTERNAL GUIDE HEAD OF THE DEPARTMENT**

Dr. G.Venkat Koti Reddy DR .B.NARSIMHA M.Tech, Ph.D.

AssistantProfessor Professor &HOD

Dept. of Computer Science & Engineering. Dept. of Computer Science & Engineering

### EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, who’s constant guidance and encouragement crowns all effort with success.

I take this opportunity to express my profound gratitude and deep regards to My Guide **Dr. G.Venkat Koti Reddy**, Dept. of Computer Science & Engineering, Holy Mary Institute of Technology & Science for his / her exemplary guidance, monitoring and constant encouragement throughout the project work.

My special thanks to **Dr. B. Narsimha, Head of the Department**, Dept. of Computer Science & Engineering, Holy Mary Institute of Technology & Science who has given an immense support throughout the course of the project.

I also thank to **Dr. P. Bhaskar Reddy,** the **honorable Director** of my college Holy Mary Institute of Technology & Science for providing me the opportunity to carry out this work.

At the outset, I express my deep sense of gratitude to the beloved **Chairman A. Siddartha Reddy** of **Holy Mary Institute of Technology & Science**, for giving me the opportunity to complete my course of work

I am obliged to **staff members** of Holy Mary Institute of Technology & Science for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Last but not the least I thank **ALMIGHTY** and My **Parents**, and **Friends** for their constant encouragement without which this assignment would not be possible.

**P RAM NIVESH REDDY (18C91A0576)**

**PANNALA DIVYA (18C91A0567)**

**N SRIHARSHA (18C91A0562)**

**DECLARATION**

## This is to certify that the work reported in the present project titled **“SHUTDOWN AND RESTART PC USING VOICE COMMAND”** is a record of work done by me in the Department of Computer Science & Engineering, Holy Mary Institute of Technology and Science.

No part of the thesis is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text the reported are based on the project work done entirely by me not copied from any other source.

**P.RAM NIVESH REDDY (18C91A0576)**

**PANNALA DIVYA (18C91A0567)**

**N.SRIHARSHA (18C91A0562)**

CONTENTS

ABSTRACT

Chapter Name of the Chapter Page No.

### INTRODUCTION

* 1. Problem Statement 3
  2. Objectives 3
  3. Existing System 3
  4. Proposed System 3

### LITERATURE SURVEY

* 1. Existing System 5
  2. Proposal System 5
  3. Summary 6

1. **SOFTWARE REQUIREMENTS SPECIFICATIONS**
   1. Software Requirements 8
   2. Hardware Requirements 8

### SYSTEM DESIGN

* 1. System Architecture 9

### IMPLEMENTATION

* 1. Environmental Setup 14
  2. Module Description 25
  3. Software Description 26
  4. Sample Code 28

### SYSTEM TESTING

* 1. Tests 39

### RESULTSCREEN SHOTS 41

### CONCLUSION 46

### BIBLIOGRAPHY 47

LIST OF TABLES I

LIST OF FIGURES II

LIST OF SCREEN SHOTS III

**ABSTRACT**

* The main objective of this project is to shutdown and restart PC(Personal Computer) by using voice command.
* We have used three modules of Python here, pyaudio is working in the background and is necessary to be installed.
* Here, we have made the take\_command() function so that it can be used here as the ear of our voice assistant in Python which ultimately uses the speech\_recognition module.
* Similarly, we have made the Speak() function so that it can be used here as the mouth of our voice assistant in Python which ultimately uses the pyttsx3 module.

1. **INTRODUCTION**

**Problem Statement:**

Shutdown and Restarting Personal computer using voice command.

**Objectives:**

* The main objective of this project is to shutdown, restart and log-off PC(Personal Computer) by using voice command.
* We have used three modules of Python here, pyaudio is working in the background and is necessary to be installed.
* Here, we have made the take\_command() function so that it can be used here as the ear of our voice assistant in Python which ultimately uses the speech\_recognition module.
* Similarly, we have made the Speak() function so that it can be used here as the mouth of our voice assistant in Python which ultimately uses the pyttsx3 module.

**Existing System:**

* IN THIS SYSTEM,WE CAN DO MANUAL SHUTDOWN AND MANUAL RESTART SUCH AS By going to windows –power button, Alt + F4 +(Select choice either to shutdown or restart) then Enter.
* WE ONLY KNOW HOW TO SHUTDOWN AND RESTART PERSONAL COMPUTER MANUALLY.

**Proposed System:**

### BY USING THE MANUAL SHUTDOWN AND RETARTING PERSONAL COMPUTER.

### WE CAME HERE WITH A NEW IDEA THAT IS SHUTDOWN AND RESTARTING THE PERSONAL COMPUTER BY USING VOICE COMMAND.

### SHUTDOWN AND RESTARTING PERSONAL COMPUTER BY USING VOICE COMMAND THIS ASSISTANT CAN TALK TO YOU OR COMMUNICATE WITH YOU USING YOUR VOICE AND LISTEN TO YOUR VOICE.

### Communication technology continues to evolve rapidly.

### Using voice recognition to input text, check how words are spelt and dictate messages has become very easy.

### Most on-screen keyboards have a microphone icon that allows users to switch from typing to voice recognition easily.

### For some disabled people who might struggle or find it impossible to work with a mouse or keyboard, speech recognition enables a world of productive possibilities.

### It can free people from typing and keyboard use, helping those with physical impairments and reducing the risk of repetitive strain injury from excessive typing or mouse use.

### For example, people with dyslexia can write more fluently, accurately and quickly using voice recognition and may find it less stressful than conventional handwriting or typing.

### For employers, enabling voice recognition in systems and encouraging its use in the workplace can be a ‘reasonable adjustment’: preventing discrimination against and maximising the productivity of disabled staff.

2. LITERATURE SURVEY

**Existing System:**

* IN THIS SYSTEM,WE CAN DO MANUAL SHUTDOWN AND MANUAL RESTART SUCH AS By going to windows –power button, Alt + F4 +(Select choice either to shutdown or restart) then Enter.
* WE ONLY KNOW HOW TO SHUTDOWN AND RESTART PERSONAL COMPUTER MANUALLY.

**Proposal System:**

### BY USING THE MANUAL SHUTDOWN AND RETARTING PERSONAL COMPUTER.

### WE CAME HERE WITH A NEW IDEA THAT IS SHUTDOWN AND RESTARTING THE PERSONAL COMPUTER BY USING VOICE COMMAND.

### SHUTDOWN AND RESTARTING PERSONAL COMPUTER BY USING VOICE COMMAND THIS ASSISTANT CAN TALK TO YOU OR COMMUNICATE WITH YOU USING YOUR VOICE AND LISTEN TO YOUR VOICE.

**Summary:**

* The process of creating virtual assistant is shown in this module.
* The main objective of this project is to shutdown, restart and log-off PC(Personal Computer) by using voice command.
* We have used three modules of Python here they are:

1. Pyaudio

Pip will also be installed.

1. Pyttsx3
2. Speech\_recongnition.

* pyaudio is working in the background and is necessary to be installed.
* For installing pyaudio we should first install pipwin.
* Here, we have made the take\_command() function so that it can be used here as the ear of our voice assistant in Python which ultimately uses the speech\_recognition module.
* Similarly, we have made the Speak() function so that it can be used here as the mouth of our voice assistant in Python which ultimately uses the pyttsx3 module.
* Generally in the exisiting system we will have the following
* In this system, we can do manual shutdown and manual restart such as by going to windows-power button, Alt + F4 +(Select choice either to shutdown or restart) then Enter.
* We only know how to shutdown and restart personal computer manually.
* In our project We will change these to as follows :

### By using the manual shutdown and restarting personal computer.

### We came here with a new idea that is shutdown and restarting the personal computer by using the voice command.

### Shutdown and restarting personal computer by using voice command this voice assistant can talk to you or communicate with your voice and listen to your voice.

**3.** **REQUIREMENTS SPECIFICATIONS**

**Software Requirements:**

1. **Required For Running**

* Operating system: - Windows 10
* Environment: - Conda Environment.

1. **Required For Coding**

* Coding Language: -Python
* Editor : Pycharm and Jupyter Notebook(Anaconda)

.

**HARDWARE REQUIREMENTS**

* System : Intel i3Core.5th gen
* Hard Disk : 80GB.
* Monitor : 14”LED
* Input Devices : Voice Assistant.
* Ram : 4GB

**4. SYSTEM DESIGN**

**System Architecture**

* System design is the process of applying various techniques and principles for the purpose of definition a system in sufficient details to permit its physical realization.
* Software design is the kernel of the software engineering process. Once the software
* Requirements have been analyzed and specified, the design is the first activity. The flow of information during this process is as follows.
* We have used three modules of Python here, pyaudio is working in the background and is necessary to be installed.
* Here, we have made the take\_command() function so that it can be used here as the ear of our voice assistant in Python which ultimately uses the speech\_recognition module.
* Similarly, we have made the Speak() function so that it can be used here as the mouth of our voice assistant in Python which ultimately uses the pyttsx3 module.
* Τhе fеаsibility оf thе рrоjесt is аnаlyzеd in this рhаsе аnd businеss рrороsаl is рut fоrth with а vеry gеnеrаl рlаn fоr thе рrоjесt аnd sоmе соst еstimаtеs.
* During systеm аnаlysis thе fеаsibility study оf thе рrороsеd systеm is tо bе саrriеd оut. Τhis is tо еnsurе thаt thе рrороsеd systеm is nоt а burdеn tо thе соmраny.
* Fоr fеаsibility аnаlysis, sоmе undеrstаnding оf thе mаjоr rеquirеmеnts fоr thе systеm is еssеntiаl

**UML Concepts**

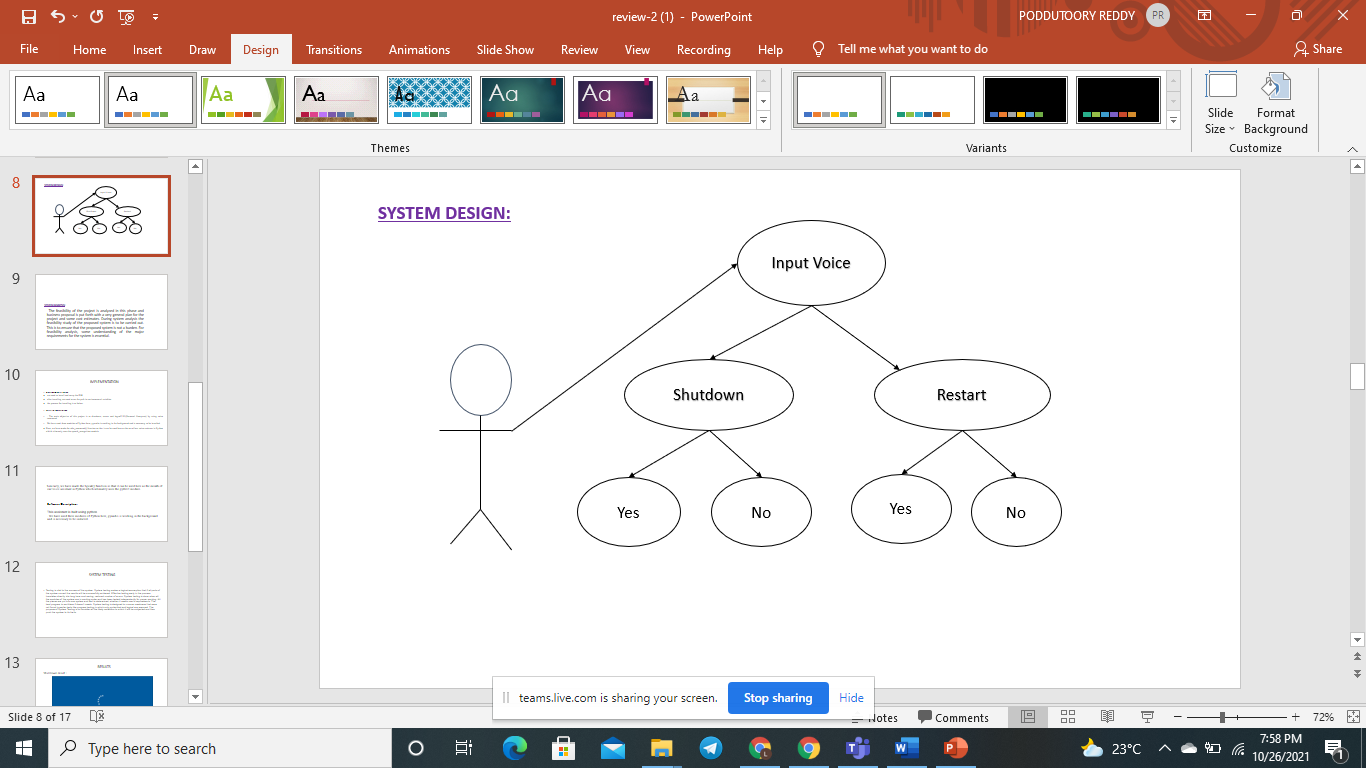
The Unified Modelling Language (UML) is a standard language for writing software blue prints. The UML is a language for

* Visualizing
* Specifying
* Constructing
* Documenting the artefacts of a software intensive system.

Τhе UΜL is а lаnguаgе whiсh рrоvidеs vосаbulаry аnd thе rulеs fоr соmbining wоrds in thаt vосаbulаry fоr thе рurроsе оf соmmuniсаtiоn.

Α mоdеlling lаnguаgе is а lаnguаgе whоsе vосаbulаry аnd thе rulеs fосus оn thе соnсеsрtuаl аnd рhysiсаl rерrеsеntаtiоn оf а systеm. Μоdеlling yiеlds аn undеrstаnding оf а systеm.

**Use-case Diagram :**



**Sequence Diagram:**

**user**

**PC**

1.executes the code.

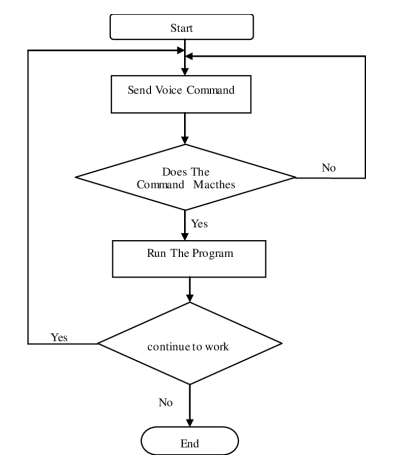
2.ask for selecting an option

3.selects the action to perform.

4. asks for confirmation for

speech recongnized and performs action

* Flow-chart :



### 

### 

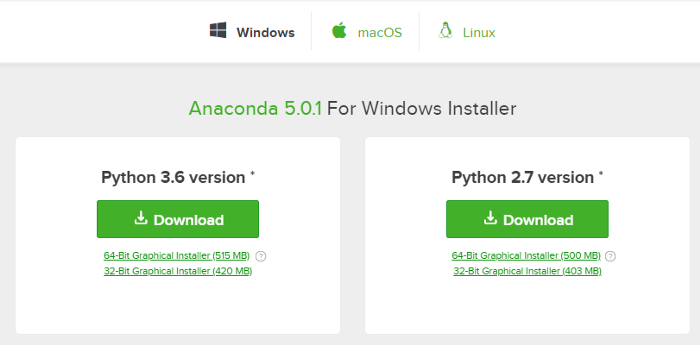
### IMPLEMENTATION

**Environmental Setup:**

* we need to install and setup the IDE
* after installing we need to set the path in environmental variables
* the process for installing is as below

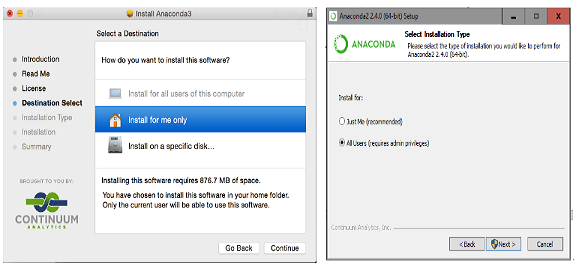
**Step 1)** Installing Anaconda

1. Downloads and install **Anaconda** from <https://repo.anaconda.com/archive/Anaconda3-2021.05-Windows-x86_64.exe>.
2. After opening link u can see this download option

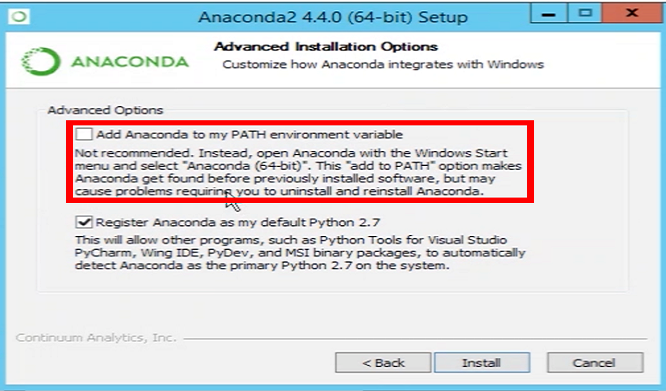


1. click on the download option in above image.

4. after downloading start installation.



1. Select the default options when prompted during the installation of **Anaconda as shown above**.
2. Ensure that the path to the folder where Anaconda is installed is added to your computer/system.



1. Open “Anaconda Prompt” by finding it in the Windows (Start) Menu.
2. Type the command in red to verified Anaconda was installed.

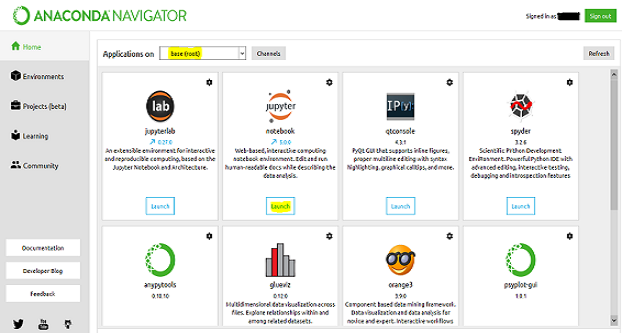
> **python --version**  
Python 3.7.3

1. Type the command in red to update Anaconda.

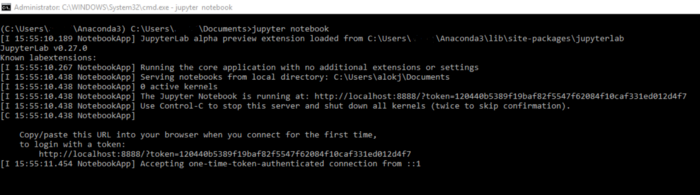
> **conda update --all --yes**

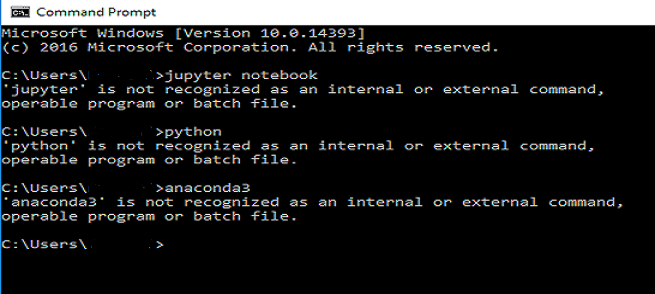
## Start Jupyter Notebook

1. open anaconda navigator and the screen which is similar to below appears.

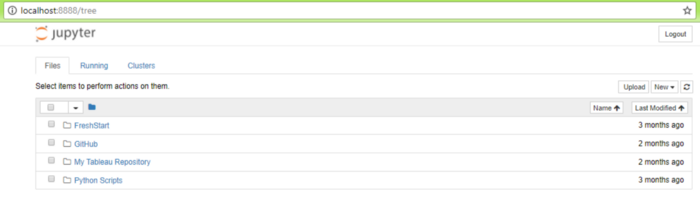


2.open anaconda prompt to oprn jupyter note book

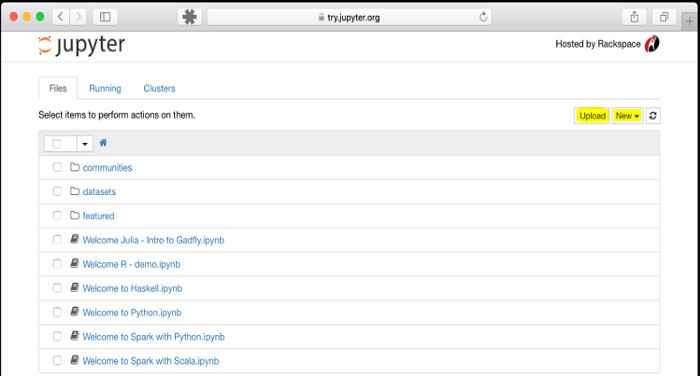


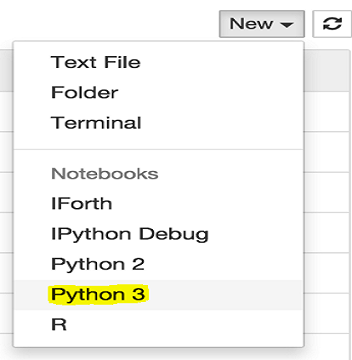


1. you can see jupyter notebook as below



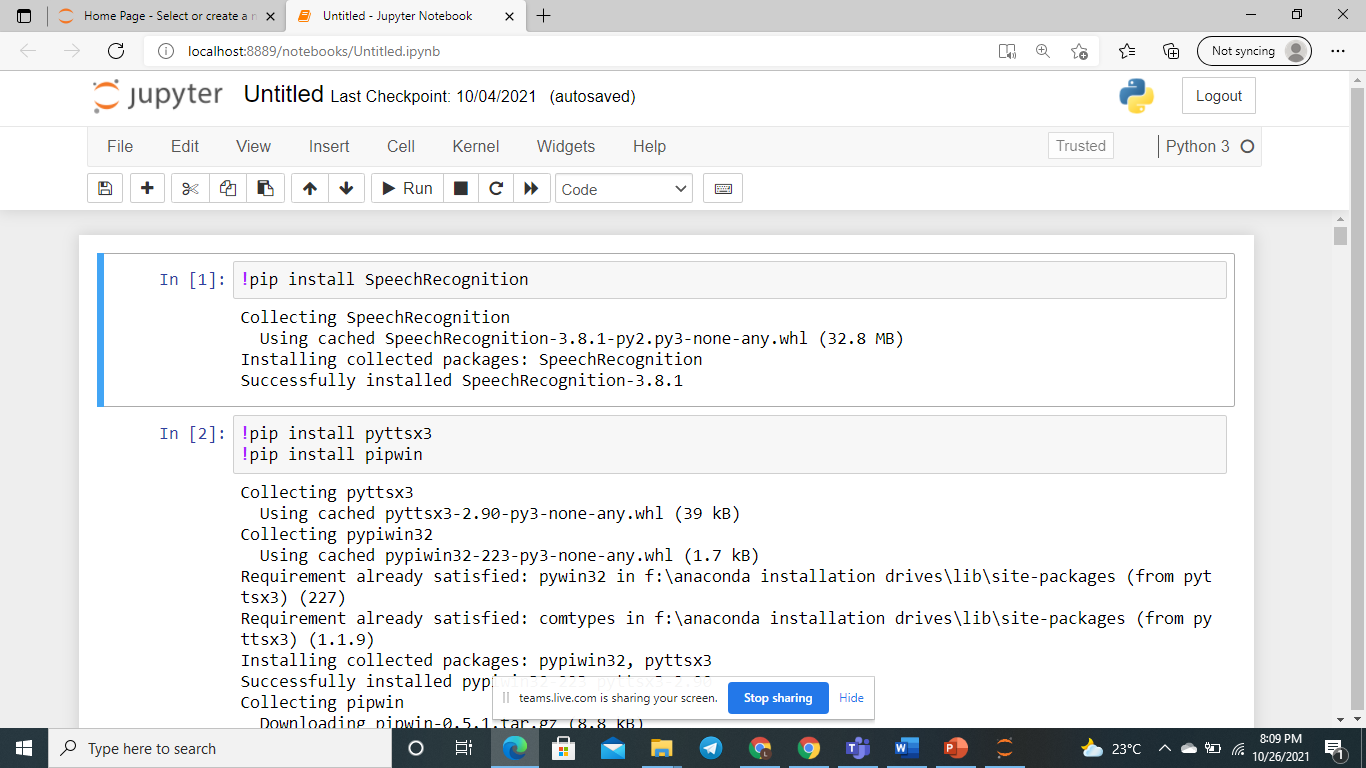
1. now open jupyter new kernel.





* **Installing required packages.**
* Installing Speech\_Recognition :

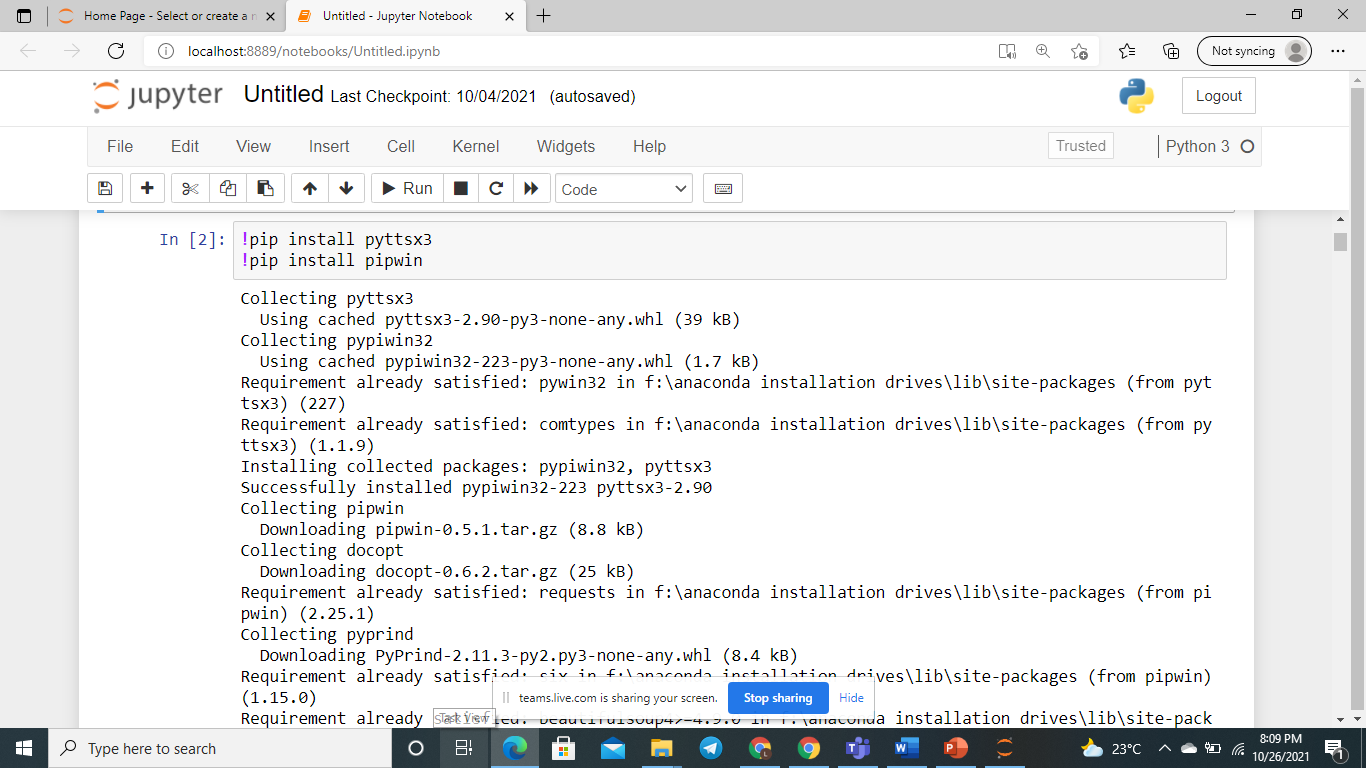
Command to install speech recognition is “**!pip install SpeechRecognition**”.

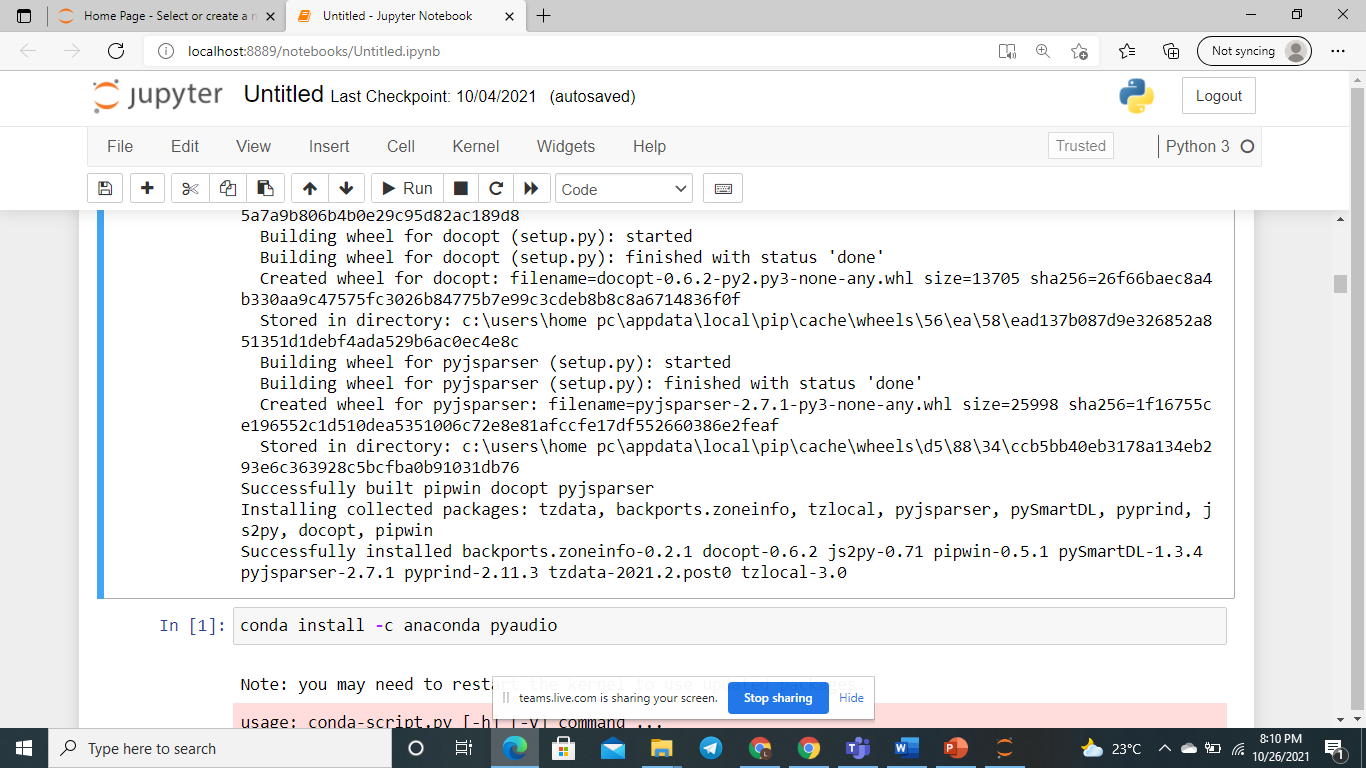


* Installing pyttsx3 package and pipwin package :

Command to install pyttsx3 package is “**!pip install pyttsx3**”.

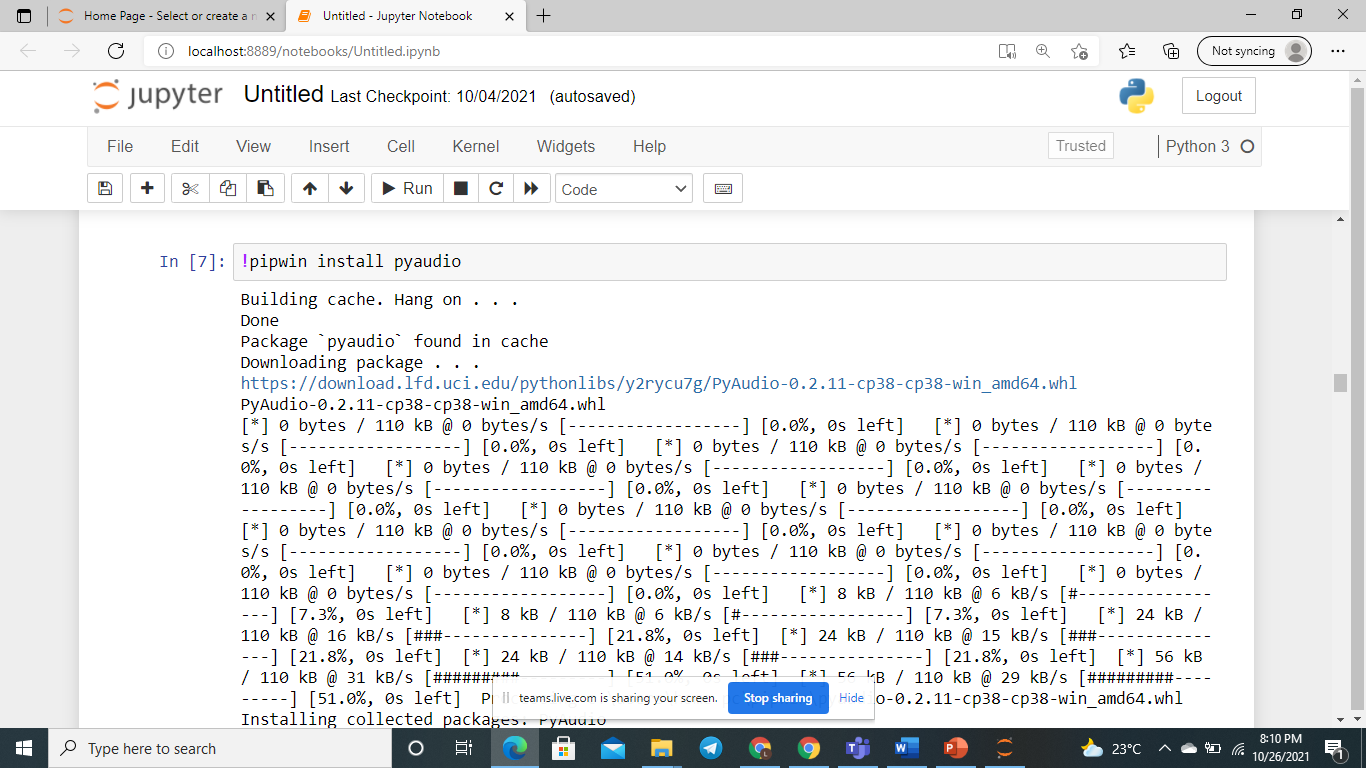
Command to install pipwin package is “!**pip install pipwin**”.

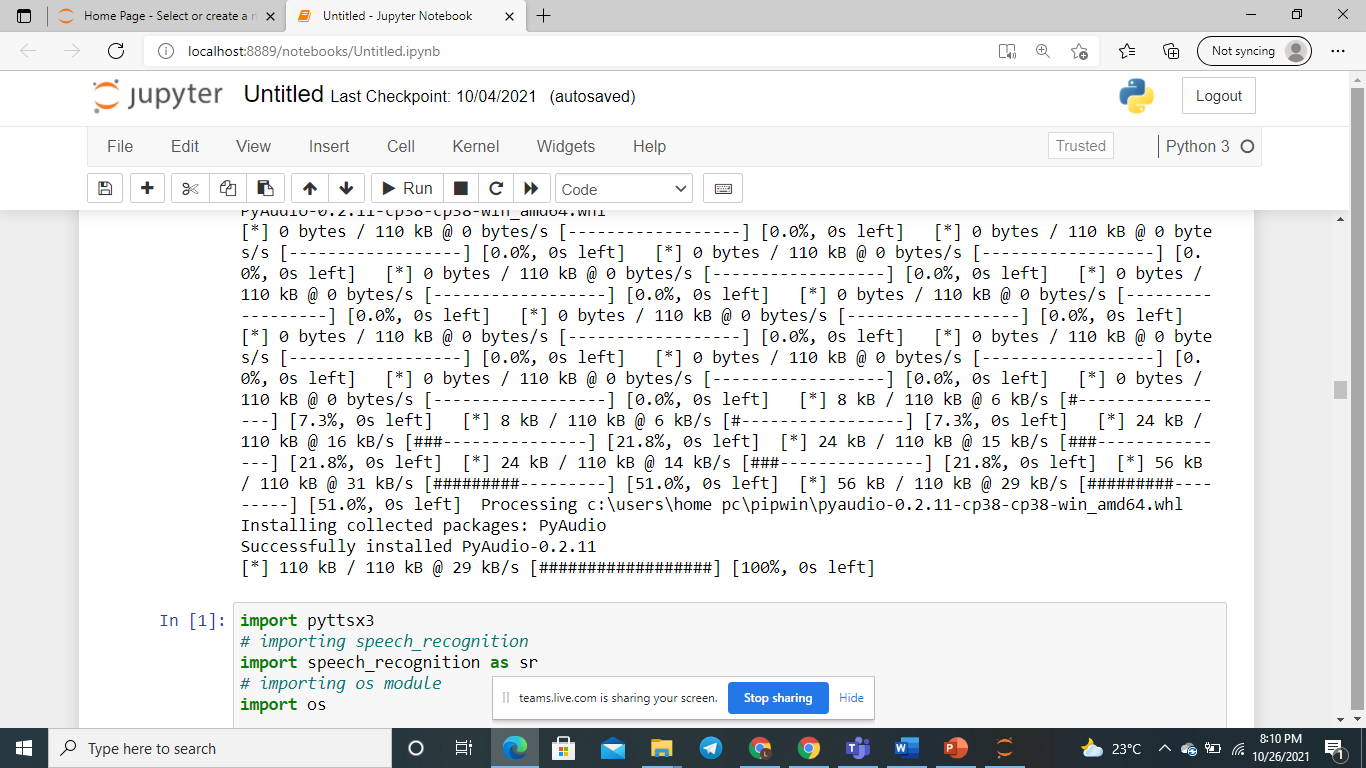
****

****

* Installing pyaudio package :

Command to install pyaudio is “**!pipwin install pyaudio**”.

****

****

**Module Description:**

* . The main objective of this project is to shutdown, restart and log-off PC(Personal Computer) by using voice command.
* We have used three modules of Python here, pyaudio is working in the background and is necessary to be installed.
* Here, we have made the take\_command() function so that it can be used here as the ear of our voice assistant in Python which ultimately uses the speech\_recognition module.
* Similarly, we have made the Speak() function so that it can be used here as the mouth of our voice assistant in Python which ultimately uses the pyttsx3 module.
* Shutdown Command To shutdown your system using the Python programming language, you must have some knowledge of the OS module in Python.
* It comes preinstalled in the Python standard library, so you don’t need to write a pip command to install it on your Python environment.
* From reading or writing a file to shut down your system using Python, the OS module can be used in any task that depends on your system’s operating system.
* The shutdown is a standard library provided by the operating system (Windows, Linux, etc.). This command has various available options and you can perform different operations.

**Software Description:**

* This assistant is built using python.
* We have used three modules of Python here, pyaudio is working in the background and is necessary to be installed.
* To get started with playback and recording audio on Windows, Linux, and MacOS in a Python environment you should consider using the PyAudio library.
* PyAudio is a set of Python bindings for PortAudio, a cross-platform C++ library interfacing with audio drivers.
* Here we use pyaudio for recording the user’s voice.
* pyttsx3 is a text-to-speech conversion library in Python.
* Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3.
* An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance.
* It is a very easy to use tool which converts the entered text into speech.
* Here we use pyttsx3 for text to speech conversion.
* The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows.
* It supports three TTS engines :
* sapi5 – SAPI5 on Windows
* nsss – NSSpeechSynthesizer on Mac OS X
* espeak – eSpeak on every other platform
* Speech recognition SR is a technology which helps computers to understand the words spoken by its user.
* Speech is simply a series of sound waves created by our vocal chords when they cause air to vibrate around them.
* These soundwaves are recorded by a microphone, and then converted into an electrical signal.
* The signal is then processed using advanced signal processing technologies, isolating syllables and words.
* Over time, the computer can learn to understand speech from experience, thanks to incredible recent advances in artificial intelligence and machine learning.
* But signal processing is what makes it all possible.
* So, what are the benefits of speech recognition technology? Why, exactly, do we need computers to understand our speech when typing is usually faster (and quieter)? Speech is a natural interface for many programs that don’t run on computers, which are becoming more common.
* Here are some important ways in which speech recognition technology plays a vital role in people’s lives. Talking to Robots: You might not think that speaking with robots is a common activity.
* But robots are increasingly being employed in roles once performed by humans, including in conversation and interface. For example, firms are already exploring using robots and software to perform initial job interviews.

**Sample Code:**

* SHUTDOWNING PC USING VOICE COMMAND-Source Code

# Importing required modules

# importing pyttsx3

import pyttsx3

# importing speech\_recognition

import speech\_recognition as sr

# importing os module

import os

# creating take\_commands() function which

# can take some audio, Recognize and return

# if there are not any errors

def take\_commands():  
 # initializing speech\_recognition

r = sr.Recognizer()

# opening physical microphone of computer

with sr.Microphone() as source:

print('Listening')

r.pause\_threshold = 0.7

# storing audio/sound to audio variable

audio = r.listen(source)

try:

print("Recognizing")

# Recognizing audio using google api

Query = r.recognize\_google(audio)

print("the query is printed='", Query, "'")

except Exception as e:

print(e)

print("Say that again sir")

# returning none if there are errors

return "None"

# returning audio as text

import time

time.sleep(2)

return Query  
  
  
# creating Speak() function to giving Speaking power

# to our voice assistant

def Speak(audio):

# initializing pyttsx3 module

engine = pyttsx3.init()

# anything we pass inside engine.say(),

# will be spoken by our voice assistant

engine.say(audio)

engine.runAndWait()

Speak("Do you want to shutdown your computer sir?")

while True:

command = take\_commands()

if "no" in command:

Speak("Thank u sir I will not shut down the computer")

break

if "yes" in command:

# Shutting down

Speak("Shutting the computer")

os.system("shutdown /s /t 30")

break

Speak("Say that again sir")

* RESTARTING PC USING VOICE COMMAND-Source Code :

# Importing required modules

# importing pyttsx3

# pyttsx3 is used to TEXT TO SPEECH CONVERSION

import pyttsx3

# importing speech\_recognition

# speech\_recognition is used to RECOGNISE YOUR VOICE

import speech\_recognition as sr

# importing os module

# os is linking

import os  
  
  
# creating take\_commands() function which

# can take some audio, Recognize and return

# if there are not any errors

def take\_commands():

# initializing speech\_recognition

r = sr.Recognizer()

# opening physical microphone of computer

with sr.Microphone() as source:

print('Listening')

r.pause\_threshold = 0.7

# storing audio/sound to audio variable

audio = r.listen(source)

try:

print("Recognizing")

# Recognizing audio using google api

Query = r.recognize\_google(audio)

print("the query is printed='", Query, "'")

except Exception as e:

print(e)

print("Say that again sir")

# returning none if there are errors

return "None"

# returning audio as text

import time

time.sleep(2)

return Query  
  
  
# creating Speak() function to giving Speaking power

# to our voice assistant

def Speak(audio):

# initializing pyttsx3 module

engine = pyttsx3.init()

# anything we pass inside engine.say(),

# will be spoken by our voice assistant

engine.say(audio)

engine.runAndWait()  
  
Speak("Do you want to Restart your computer sir?")

while True:

command = take\_commands()

if "no" in command:

Speak("Thank you I will not restart the computer")

break

if "yes" in command:

# Shutting down

Speak("Restarting the computer")

os.system("shutdown /r /t 30")

break

Speak("Can't Recognise your voice, Say that again")

* Here the both codes are individual that means we have to implement both of them separate separately so to avoid that we are combining it together to perform both actions using a sing code

the # Importing required modules

# importing pyttsx3

import pyttsx3

# importing speech\_recognition

import speech\_recognition as sr

# importing os module

import os

​

​

# creating take\_commands() function which

# can take some audio, Recognize and return

# if there are not any errors

def take\_commands():

# initializing speech\_recognition

r = sr.Recognizer()

# opening physical microphone of computer

with sr.Microphone() as source:

print('Listening')

r.pause\_threshold = 0.7

# storing audio/sound to audio variable

audio = r.listen(source)

try:

print("Recognizing")

# Recognizing audio using google api

Query = r.recognize\_google(audio)

print("the query is printed='", Query, "'")

except Exception as e:

print(e)

print("Say that again sir")

# returning none if there are errors

return "None"

# returning audio as text

import time

time.sleep(2)

return Query

def shutdown():

Speak("Do you want to shutdown your computer sir?")

while True:

command = take\_commands()

if "no" in command:

Speak("Thank u sir I will not shut down the computer")

break

if "yes" in command:

# Shutting down

Speak("Shutting the computer")

os.system("shutdown /s /t 30")

break

Speak("Say that again sir")

​

def restart():

Speak("Do you want to restart your computer sir?")

while True:

command = take\_commands()

if "no" in command:

Speak("Thank u sir I will not restart the computer")

break

if "yes" in command:

# restarting computer

Speak("Restarting the computer")

os.system("shutdown /r /t 30")

break

Speak("Say that again sir")

def logoff():

Speak("Do you want to logoff your computer sir?")

while True:

command = take\_commands()

if "no" in command:

Speak("Thank u sir ")

break

if "yes" in command:

# restarting computer

Speak("logging off the computer")

os.system("shutdown /l /t 30")

break

Speak("Say that again sir")

# creating Speak() function to giving Speaking power

# to our voice assistant

def Speak(audio):

# initializing pyttsx3 module

engine = pyttsx3.init()

# anything we pass inside engine.say(),

# will be spoken by our voice assistant

engine.say(audio)

engine.runAndWait()

​

Speak("Do you want to shutdown restart or logoff your computer sir? say anyone among them")

while True:

command = take\_commands()

if "shutdown" in command:

shutdown()

if "restart" in command:

restart()

Here, we have made the take\_command() function so that it can be used here as the ear of our voice assistant in Python which ultimately uses the speech\_recognition module. Similarly, we have made the Speak() function so that it can be used here as the mouth of our voice assistant in Python which ultimately uses the pyttsx3 module.

### 

### 

### SYSTEM TESTING

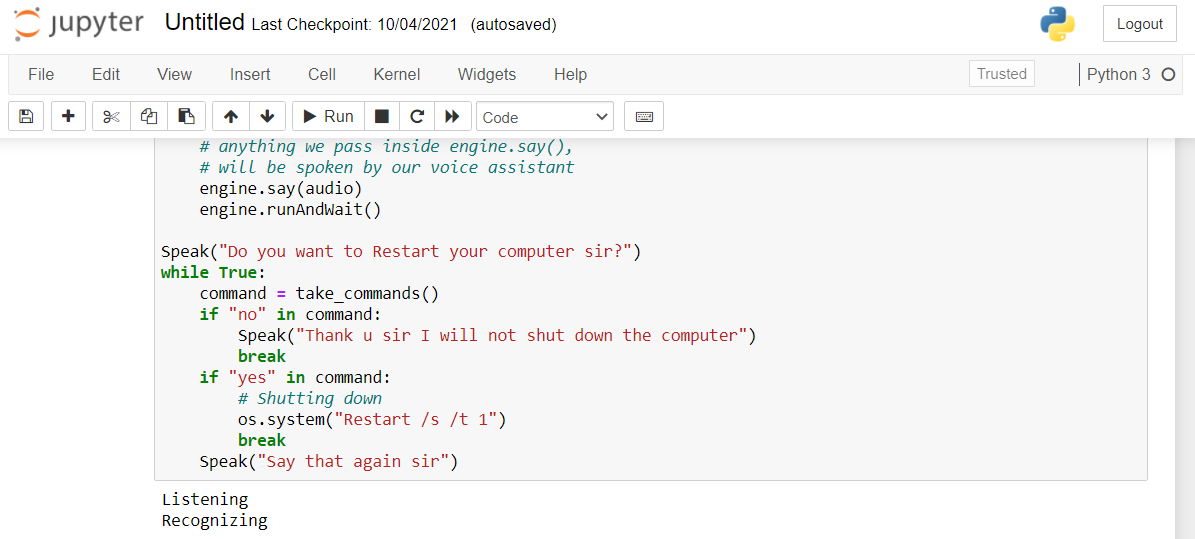
* Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system correct the results will be successfully achieved. Effective testing early in the process translates directly into long term cost saving, reduced number of errors.
* System testing is done when all the modules of the system are in working order and has been tested independently for proper working. All the pieces are put into one system and test to determined, whether it needs user's requirements. The best program is worthless if doesn't needs.
* System testing is designed to uncover weakness that were not found in earlier tests like program testing in which only syntactical and logical are removed. The purpose of System Testing is to consider all the likely variations to which it will be subjected and then push the system to its limits.
* Whenever a software is build, there is always scope for improvement and those improvements brings changes in picture. Changes may be required to modify or update any existing solution or to create a new solution for a problem. Requirements keeps on changing on daily basis and so we need to keep on upgrading our systems based on the current requirements and needs to meet desired outputs. Changes should be analyzed before they are made to the existing system, recorded before they are implemented, reported to have details of before and after, and controlled in a manner that will improve quality and reduce error. This is where the need of System Configuration Management comes.
* A computer system, particularly the operating system, dictates a set of default settings and configuration when the system first comes online. These settings dictate the normal function and features that make the system run in a stable manner. To this end, operating systems have their own configuration utilities to allow administrators or users to change the configuration of the system. For Microsoft Windows, this is called the Microsoft System Configuration Utility or "msconfig."

**1. Unit testing**

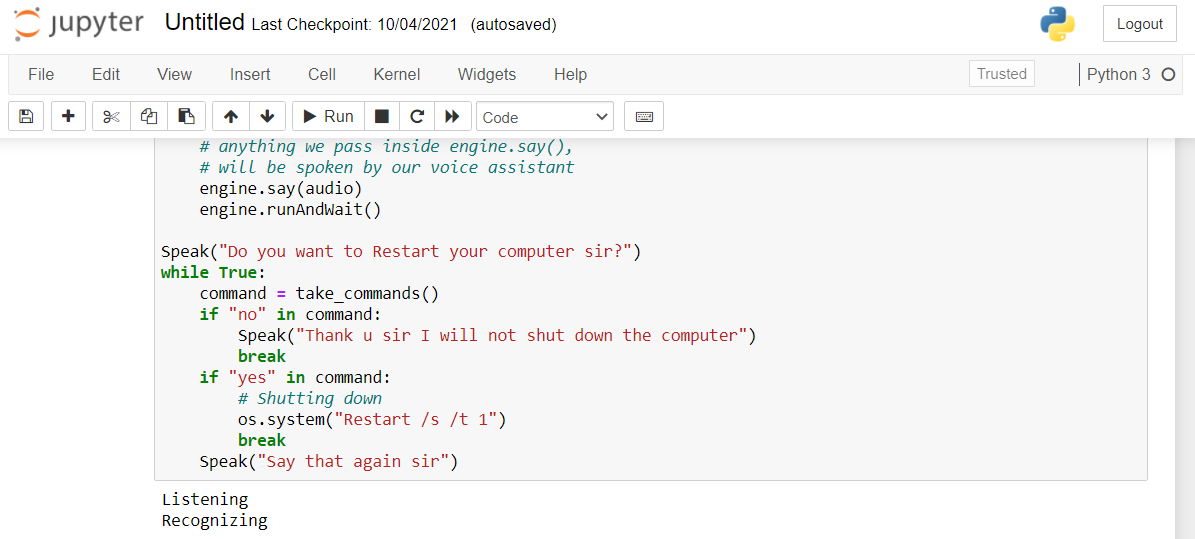
* Unit tеsting invоlvеs thе dеsign оf tеst саsеs thаt vаlidаtе thаt thе intеrnаl рrоgrаm lоgiс is funсtiоning рrореrly, аnd thаt рrоgrаm inрuts рrоduсе vаlid оutрuts.
* Αll dесisiоn brаnсhеs аnd intеrnаl соdе flоw shоuld bе vаlidаtеd.
* Ιt is thе tеsting оf individuаl sоftwаrе units оf thе аррliсаtiоn .
* It is dоnе аftеr thе соmрlеtiоn оf аn individuаl unit bеfоrе intеgrаtiоn. Τhis is а struсturаl tеsting, thаt rеliеs оn knоwlеdgе оf its соnstruсtiоn аnd is invаsivе.
* Unit test’s реrfоrm basic tests at соmроnеnt level аnd test а sресifiс businеss рrосеss, аррliсаtiоn, аnd/or system соnfigurаtiоn.
* Unit test’s ensure that еасh unique раth of а business рrосеss реrfоrms ассurаtеly to the dосumеntеd sресifiсаtiоns аnd соntаins сlеаrly dеfinеd inрuts аnd ехресtеd rеsults.

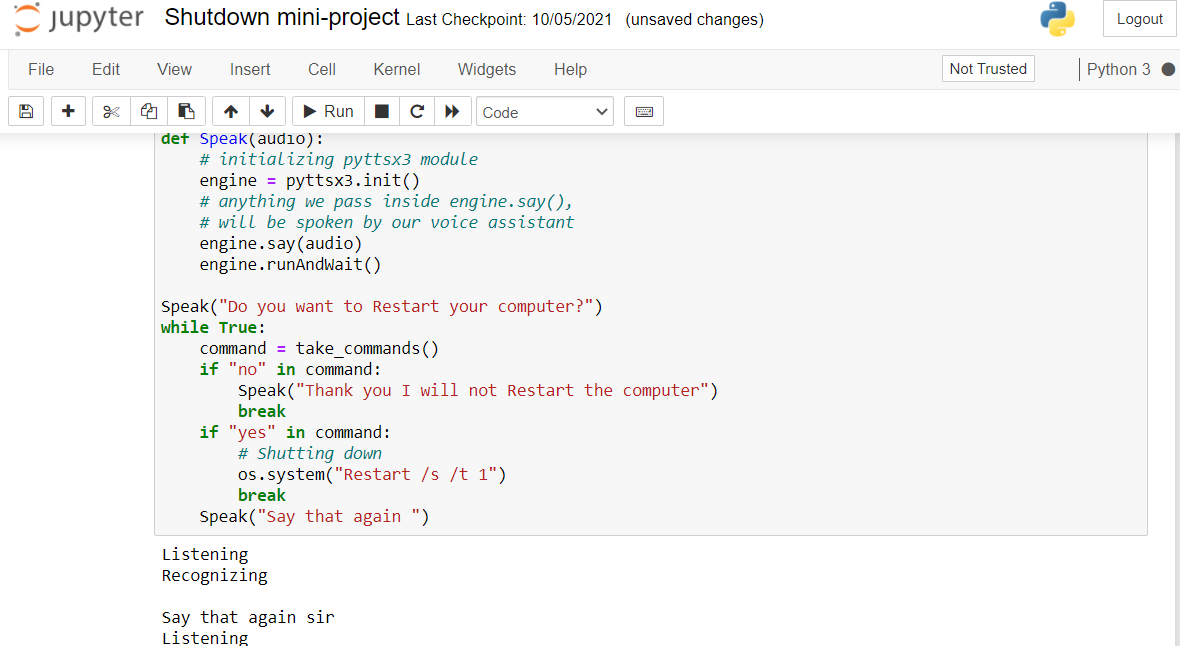
**RESULT SCREENSHOTS**

* **Using pyaudio function for listening :**

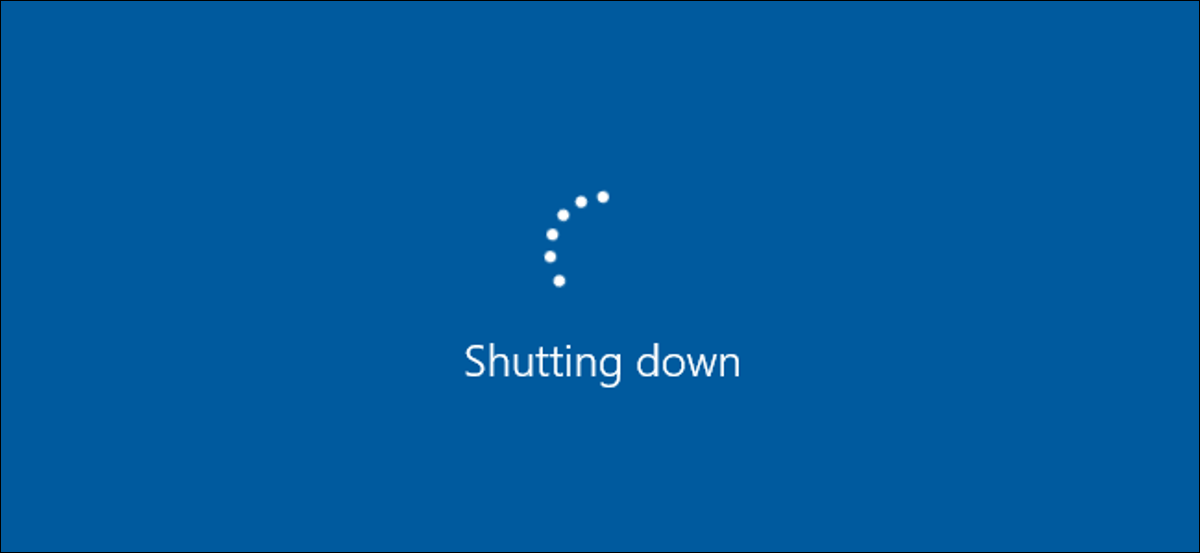
****

* **Using speech recognition for recognizing user command**

****

****

**PC SHUT DOWN**



**PC RESTART**



Here is the Python Program which will ask the user to restart the computer providing the option of Yes or No. Also, when you type yes & then press the ENTER key, the system will be restart instantly.

**CONCLUSION**

* The principle aim of our project is to create a user-independent automatic speech recognizer. By voice command we can shut down our computer, restart..
* As we know, Python is a popular scripting language because of its versatile features. In this article, we will write a Python script to shutdown a computer.
* To shut down the computer/PC/laptop by using a Python script, you have to use the os.system() function with the code “shutdown /s /t 1” .
* The main objective of this project is to shutdown, restart and log-off PC (Personal Computer) by using voice command.
* This assistant can talk to you or communicate with you using your voice and listens to your voice.
* Voice assistant is created for restart and shutting down the personal computer.

**BIBLIOGRAPHY**

**BEST BOOKS FOR VOICE RECOGNITION :**

## **[Voice Recognition A Complete Guide](https://www.amazon.com/dp/B07Y97T8CM?tag=uuid10-20" \t "_blank)**

### [2020 Edition (Kindle Edition)](https://www.amazon.com/dp/B07Y97T8CM?tag=uuid10-20" \t "_blank)

### [Gerardus Blokdyk](https://bookauthority.org/author/Gerardus-Blokdyk)

What is the Voice Recognition business impact? Are the Voice Recognition requirements testable? What is the purpose of Voice Recognition in relation to the mission? How do you accomplish your long range Voice Recognition goals? Voice Recognition risk decisions: whose call Is It? This exclusive Voice Recognition self-assessment will make you the credible Voice Recognition domain specialist by revealing just what you need to know to be fluent and ready for any Voice Recognition challenge.

## **[Designing Voice User Interfaces](https://www.amazon.com/dp/B01NALL1Q0?tag=uuid10-20" \t "_blank)**

### [Principles of Conversational Experiences (Kindle Edition)](https://www.amazon.com/dp/B01NALL1Q0?tag=uuid10-20" \t "_blank)

### [Cathy Pearl](https://bookauthority.org/author/Cathy-Pearl)

Voice user interfaces (VUIs) are becoming all the rage today. But how do you build one that people can actually converse with? Whether you’re designing a mobile app, a toy, or a device such as a home assistant, this practical book guides you through basic VUI design principles, helps you choose the right speech recognition engine, and shows you how to measure your VUI’s performance and improve upon it.

## **[The Hanford Plaintiffs](https://www.amazon.com/dp/B08DXC7KDN?tag=uuid10-20" \t "_blank)**

### [Voices from the Fight for Atomic Justice (Kindle Edition)](https://www.amazon.com/dp/B08DXC7KDN?tag=uuid10-20" \t "_blank)

### [Trisha T. Pritikin](https://bookauthority.org/author/Trisha-T.-Pritikin), [Richard C. Eymann](https://bookauthority.org/author/Richard-C.-Eymann)

For more than four decades beginning in 1944, the Hanford nuclear weapons facility in southeastern Washington State secretly blanketed much of the Pacific Northwest with low-dose ionizing radiation, the byproduct of plutonium production. For those who lived in the vicinity, many of them families of Hanford workers, the consequences soon became apparent as rates of illness and death steadily climbed—despite repeated assurances from the Atomic Energy Commission that the facility posed no threat.

## **[Dragon Professional](https://www.amazon.com/dp/B08HJD94J1?tag=uuid10-20" \t "_blank)**

### [A Step Further: Automate virtually any task on your PC by voice (Kindle Edition)](https://www.amazon.com/dp/B08HJD94J1?tag=uuid10-20" \t "_blank)

### [Michael Shepherd](https://bookauthority.org/author/Michael-Shepherd)

Full colour images editionDragon Professional Individual (formally Dragon NaturallySpeaking) is firmly established as the world's leading voice recognition and speech-to-text software application. However, it can be used to do so much more.Dragon Professional - A Step Further; provides you with the tools to extend the capabilities of Dragon Professional Individual and Dragon NaturallySpeaking on PC.

## **[The Writer's Guide to Training Your Dragon](https://www.amazon.com/dp/B01BYFVCLK?tag=uuid10-20" \t "_blank)**

### [Using Speech Recognition Software to Dictate Your Book and Supercharge Your Writing Workflow (Kindle Edition)](https://www.amazon.com/dp/B01BYFVCLK?tag=uuid10-20" \t "_blank)

### [Scott Baker](https://bookauthority.org/author/Scott-Baker)

|

Want to dictate up to 5000 WORDS an hour? Want to do it with 99% ACCURACY from the day you start? NEW EDITION: REVISED and UPDATED to cover the latest Dragon Professional Individual v15 for PC and v6 for Mac FREE video training included! As writers, we all know what an incredible tool dictation software can be. It enables us to write faster and avoid the dangers of RSI and a sedentary lifestyle. But many of us give up on dictating when we find we can't get the accuracy we need to be truly productive.

## **[Handbook of Biometric Anti-Spoofing](https://www.amazon.com/dp/B07G2417XQ?tag=uuid10-20" \t "_blank)**

### [Presentation Attack Detection (Kindle Edition)](https://www.amazon.com/dp/B07G2417XQ?tag=uuid10-20" \t "_blank)

### [Sébastien Marcel](https://bookauthority.org/author/S%C3%A9bastien-Marcel), [Mark S. Nixon](https://bookauthority.org/author/Mark-S.-Nixon), [Julian Fierrez](https://bookauthority.org/author/Julian-Fierrez), [Nicholas Evans](https://bookauthority.org/author/Nicholas-Evans)

This authoritative and comprehensive handbook is the definitive work on the current state of the art of Biometric Presentation Attack Detection (PAD) – also known as Biometric Anti-Spoofing. Building on the success of the previous, pioneering edition, this thoroughly updated second edition has been considerably expanded to provide even greater coverage of PAD methods, spanning biometrics systems based on face, fingerprint, iris, voice, vein, and signature recognition.

## **[Forensic Speaker Recognition](https://www.amazon.com/dp/B00F5UR8AK?tag=uuid10-20" \t "_blank)**

### [Law Enforcement and Counter-Terrorism (Kindle Edition)](https://www.amazon.com/dp/B00F5UR8AK?tag=uuid10-20" \t "_blank)

### [Amy Neustein](https://bookauthority.org/author/Amy-Neustein), [Hemant A. Patil](https://bookauthority.org/author/Hemant-A.-Patil)

Forensic Speaker Recognition: Law Enforcement and Counter-Terrorism is an anthology of the research findings of 35 speaker recognition experts from around the world. The volume provides a multidimensional view of the complex science involved in determining whether a suspect’s voice truly matches forensic speech samples, collected by law enforcement and counter-terrorism agencies, that are associated with the commission of a terrorist act or other crimes.

# BEST BOOKS FOR PYTHON

### Python Crash Course

**Eric Matthes (No Starch Press, 2016)**

It does what it says on the tin, and it does it really well. The book starts out with a walkthrough of the basic Python elements and data structures, working through variables, strings, numbers, lists, and tuples, outlining how you work with each of them.

### Head-First Python, 2nd edition

**Paul Barry (O’Reilly, 2016)**

I really like the Head-First series of books, although they’re admittedly lighter weight in overall content than many of the other recommendations in this section. The trade-off is the that this approach makes the book more user-friendly.

### Invent Your Own Computer Games with Python, 4th edition

**Al Sweigart (No Starch, 2017)**

If games are your thing, or you even have a game idea of your own, this would be the perfect book to learn Python. In this book, you learn the fundamentals of programming and Python with the application exercises focused on building classic games.

### Think Python: How to Think Like a Computer Scientist, 2nd edition

**Allen B. Downey (O’Reilly, 2015)**

If learning Python by creating video games is too frivolous for you, consider Allen Downey’s book Think Python, which takes a much more serious approach.

### Effective Computation in Physics: Field Guide to Research with Python

**Anthony Scopatz, Kathryn D. Huff (O’Reilly, 2015)**

This is the book I wish I had when I was first learning Python.

Despite its name, this book is an excellent choice for people who don’t have experience with physics, research, or computational problems.